

Roombots extended: challenges in the next generation of self-reconfigurable modular robots and their application in adaptive and assistive furniture

S. Hauser^{a,*}, M. Mutlu^{a,b,*}, P.-A. Léziart^a, H. Khodr^a, A. Bernardino^b, A.J. Ijspeert^a

^a*BioRob, School of Engineering, EPFL, Lausanne, Switzerland*

^b*Vislab, Institute of Systems and Robotics, IST, Lisbon, Portugal*

Abstract

This work presents a series of demonstrations of our self-reconfigurable modular robots (SRMR) “Roombots” in the context of adaptive and assistive furniture. In literature, simulations are often ahead of what currently can be demonstrated in hardware with such systems due to significant challenges in transferring them to the real world. Here, we describe how Roombots tackled these difficulties in real hardware and focus qualitatively on selected hardware experiments rather than on quantitative measurements (in hardware and simulation) to showcase the many possibilities of an SRMR. We envision Roombots to be used in our living space and define five key tasks that such a system must possess. Consequently, we demonstrate these tasks, including self-reconfiguration with 12 modules (36 Degrees of Freedom), autonomously moving furniture, object manipulation and gripping capabilities, human-module-interaction and the development of an easy-to-use user interface. We conclude with the remaining challenges and point out possible directions of research for the future of adaptive and assistive furniture with Roombots.

Keywords: Self-reconfiguring, Modular robots, Universal Gripper, Assistive furniture, Adaptive furniture

*Both authors contributed equally to this work

Email addresses: simon.hauser@alumni.epfl.ch (S. Hauser),

mehmet.mutlu@alumni.epfl.ch (M. Mutlu)

URL: www.biorob.epfl.ch (A.J. Ijspeert)

Preprint submitted to Journal of Robotics and Autonomous Systems.

April 17, 2020

The official accepted version is here:

<https://doi.org/10.1016/j.robot.2020.103467>

Highlights (in separate file)

- Depiction of how modular robots could be integrated into furniture in our living space
- Definition of five key-tasks of adaptive and assistive furniture
- 5 • Development of a dedicated GUI for an easy control of multiple Roombots (RB) modules
- Hardware demonstrations of various sub-tasks involving up to 12 modules
- Demonstrations of mobility, manipulation and human-module-interaction

1. Introduction

10 Modular robots (MRs) are the “Swiss army knives” of robotics. Instead of designing a specialized robot for a single task, the idea of modular robotics in general is to provide functionalities in robotic “modules”, and a robot can be constructed out of these modules. The advertised advantages of modular robots are threefold (based on [1]):

- 15 1. versatility: different tasks require different functionalities; in a modular robot, these functionalities can be put together by assembling the according modules.
2. robustness: if a module fails, only the functionality of this specific module is affected; the rest of the robot still can operate normally and the failing
20 module can easily be replaced.
3. low-cost: since each module is thought to only perform a simple function, it is in principle also simple in design, allowing cheap manufacturing; additionally, modules can more easily be mass-produced for wide availability, helping to reduce module costs.

25 There is an ongoing discussion if the current state of modular robots yet
delivers any of these advantages: it can be challenging to realize a function in a
compound of modules; more single robotic agents means more potential for fail-
ure; and most modular systems are still in their prototype stage, making them
expensive to produce. Only the future development of the modular robotics
30 field in general will show if the concept is able to overcome current limitations
with more research, new ideas and new technologies.

Among modular robots, self-reconfigurable modular robots (SRMRs) form
a special class. Not only are the modules now reconfigurable, meaning they
can be assembled into different morphologies, but the system has the ability to
35 perform the reconfiguration autonomously. This could potentially make such a
system extremely independent since it could transform its morphology and add
and remove functionalities all by itself, leading to an uncomfortable question:
what would be the limitations of such a system? This inspired many fictional
scenarios over the years, most recently in the Walt Disney movie “Big Hero 6”
40 (2014) where the tiny “microbots” are shown to have almost omnipotent powers
when working together in clusters of many thousands of modules.

As physical robots have to deal with real world effects however, namely grav-
ity and other physical effects, such scenarios are far from being realistic and will
remain fiction for the foreseeable future. Nevertheless, research on SRMR re-
45 tains a futuristic touch, and in practise SRMRs pose interesting interdisciplinary
challenges in hardware, control and their application for an end user.

The hardware presents an engineering challenge. The general idea of mod-
ular robotics is to combine “simple” modules into larger, more complex and
thus usually more capable structures that possess the required functionality of
50 a task at hand. A true reversible modular robotic system (meaning no perma-
nent state-change) has three key functionalities: reversible connection between
two modules, reversible disconnection between two modules, and actuation (ei-
ther within one module or in a structure as a whole). Note that reversible
connection does not automatically include disconnection as there can be differ-
55 ent mechanical systems for both connection and disconnection. For the design

of each of these systems, the overall dimensions of a module should be defined, which often are linked to the expected tasks. Lastly, one has to consider if all the sub-units are exact copies of each other (homogeneous MR) or if specialized units collaborate together (heterogeneous MR) [2], which influences the type of control and planning algorithms suitable for the system. In manually reconfigurable modular robot systems [2], the connection and disconnection procedures are outsourced to a human such that an MR mainly has to care only about the actuation system. This means that each module can take on almost any form with any mechanical properties, including soft modules. In contrast, SRMR [2] systems have to perform the connection and disconnection procedures autonomously. If modules are required to attach to specific locations to connect to another module, this introduces additional constraints on the rigidity. To enable (or at the very least simplify) self-reconfiguration, each module should be structurally rigid such that some form of alignment between modules is achievable - a usually necessary condition for self-connection - yet provide some form of movement to initiate self-reconfiguration, often requiring a delicate balance between the two. It is also possible to design modules that can attach anywhere on another module (as e.g. in [3]). Moreover, soft modules (e.g.[4, 5]) can have their own form of connection and disconnection methods. The property of self-reconfiguration is already challenging to achieve in 2D SRMR systems and adds a significant level of difficulty in 3D SRMR systems.

The control requires to deal with distributed systems. Multiple control frameworks can be studied on SRMRs. In a centralized way, either an external computation unit commands all modules or one of the modules can have the master role to command all other modules in the system. Alternatively, each module can act according to their own local perception in the decentralized control. Additionally, a way for a human to intuitively interact with one or more groups of modules may have to be developed such that the current state of the modular system can be understood and a future desired state can be communicated.

Lastly, one has to consider for which application the robotic platform is to

be developed. Even though SRMR can shape-shift into various morphologies, this does not necessarily mean that one module should be as small as possible to build any arbitrary morphology; one module should be as small as the smallest
90 unit required by the application. We must also define the added value of self-reconfiguration to justify the effort to overcome the challenges mentioned above: what is the utility of self-reconfiguration?

As one example that aims at addressing some of these challenges, over the last decade we developed “Roombots” (RB), a homogeneous 3D self-reconfigurable
95 modular robot system with the application - among others - of serving as adaptive and assistive furniture. In this framework, multiple groups of modules adapt to the needs of a user by creating different pieces of furniture or augmenting the capabilities of existing furniture. Additionally, such pieces of furniture can also act somewhat autonomously and provide assistance specifically to the elderly
100 by for example preventing falls as a result of a following chair, picking up and holding objects, and bringing items, all while retaining their functionality as furniture.

The first generation of Roombots proposed the general hardware design of our SRMR [6, 7, 8] together with algorithmic work on reconfiguration [9, 10, 11],
105 locomotion [12, 13, 14, 15], and user interfaces [16, 17]. As a proof-of-concept, only a limited number of modules was needed to demonstrate the three key functions connection, disconnection and movement. However, to further the vision of using Roombots as adaptive furniture, more modules were needed and a few adaptations in hardware and control had to be made based on the
110 experience with the developed prototypes.

To showcase some of the new capabilities of the upgraded Roombots system, we propose five tasks that we regard as core functionalities of adaptive and assistive furniture: self-reconfiguration, mobility, manipulation, human-module-interaction, and developing a user interface. Each of these tasks is demonstrated
115 by Roombots, often through various sub-tasks (see Fig. 1).

The first task described herein is scalable *self-reconfiguration* which stands at the core of an SRMR system and fits well into the vision of creating adaptive

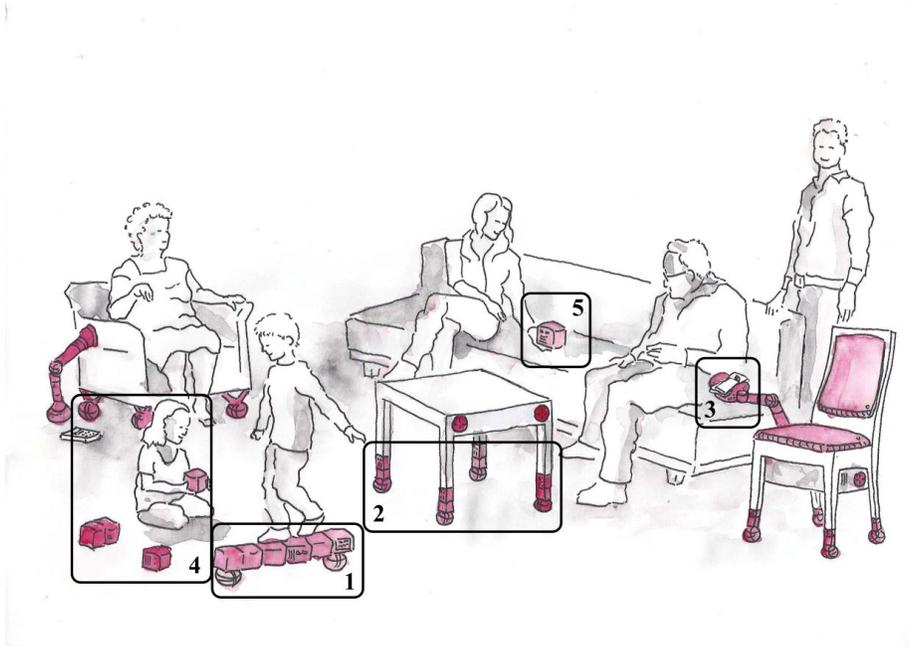


Figure 1: Roombots framework with the five tasks indicated. 1) self-reconfiguration of a group of Roombots modules into a shape to function as a toy. 2) mobile furniture where an existing piece of furniture has been enhanced with RB modules to enable it to move around in the living space. 3) object manipulation where furniture is able to assist in simple tasks such as picking up and holding a remote control. 4) interactive furniture, allowing users to work together with robotic-enhanced furniture, and 5) an easy-to-use user interface (UI) to monitor the current state of the robotic system.

furniture. For this demonstration, a loose group of RB modules self-reconfigures into a shape resembling that of a common piece of furniture. It should be noted that while this task essentially reduces to a series of connection, disconnection and movement actions which already have been demonstrated in e.g. [10], performing it with a larger number of modules is not trivial due to the scalability of the robotic system: continuous reliable reconfiguration with a group of modules rather than an isolated reconfiguration with two modules adds significant challenges in hardware and control to deal with real world physical phenomena that only occur for larger groups of modules. In the case of Roombots, the first hardware iteration experienced major elastic deformations and misalignment issues

such that the autonomous formation of larger structures was not possible. Due to the various hardware modifications presented in section 3.1 and closed-loop control described in section 3.8, such tasks lie now within or at least much closer
130 to the capabilities of the present hardware iteration.

Second, adaptive furniture should possess a certain degree of *mobility* such that pieces of furniture can dynamically move within a living space. Depending on the functionality of the structure, mobility can require certain adaptation
135 of the furniture to the environment as well as autonomy to react to changes in the environment. We performed a set of demonstrations to showcase the mobility aspect, ranging from following and evading pieces of furniture to slope compensation and even stair climbing.

To fulfill more of the assistive aspects of the Roombots vision, robotic-enhanced furniture must possess a way to *manipulate objects*. Simple actions
140 such as holding a book can already provide assistance, and it is clear that many of such actions require a method for manipulation, i.e. a gripper. Such a gripper can, for instance, be used for fetching a remote controller fallen on the ground. For the integration of a gripper into the RB framework - which should
145 be able to manipulate a wide range of everyday objects - we revisited the concept of granular jamming that was used to develop a “Universal Gripper” [18]. By miniaturizing the mechatronic components, we were able to equip selected modules with this gripper and demonstrate common tasks such as a piece of furniture picking up an object from the ground and an RB structure helping to
150 open a water bottle. Two gripper modules further are able to pass objects from one module to another, which will also be briefly discussed.

Users need an intuitive way of *interacting with modules*. Interaction could be through e.g. speech recognition where modules react to spoken commands, or through gesture control as e.g. in [17, 19, 20] where commands are created
155 through the tracking of gestures. Further, modules should provide feedback to the environment such that the current state of a single module or a group of modules can easily be understood. A simple way to achieve this is through light where different colors can be assigned to different states. We revisited our

approaches in the past and demonstrate again the interaction capabilities of
160 RB.

Lastly, when complete information about the state of the modules is required,
an appropriate *visualization tool* is needed. We describe the development of a
Graphical User Interface (GUI) that allows for easy control of RB modules. The
GUI greatly facilitated the development of many of the presented demonstra-
165 tions and will simplify the development of future challenges. Such challenges
(e.g. RB metamodule locomotion on-grid) are identified along with this work
and analyzed and discussed in section 5.

It is important to note that these five tasks are chosen by us and adapted
in ways such that they can be demonstrated by Roombots, i.e. the actual task
170 allocation is done by a human operator. It will be the work of future research
to automatically extract tasks from specific conditions and allocate it to a set
of modules; this is out of scope in the current work.

The main contribution of this work is showcasing hardware and control im-
provements of Roombots since [10]. Moreover, the demonstration of a variety
175 of tasks can serve as a benchmark for the possibilities of such an SRMR, which
we believe is so far absent in the community. The paper provides a framework
to put all published works and the improvements on Roombots in this work in
relation to adaptive and assistive furniture. Considerable hardware and control
engineering work was needed to achieve such demonstrations, which would go
180 unnoticed otherwise. Hence, these improvements are described in detail, and
the paper comprises of multiple small topics. Another contribution of this pa-
per is explicitly explaining the limitations of Roombots by giving examples and
demonstrations as we believe that pointing out shortcomings is as important
as explaining the potential benefits of SRMRs. Note that, unlike most other
185 articles in SRMR by others and by us, this article is qualitative rather than
quantitative. In particular, it does not provide many quantitative results, but
only qualitative descriptions of proof-of-concept experiments as well as assess-
ments of pros and cons of our Roombots. We hope that the article will still
be useful to share our vision, our experience, and possible insights about future

190 improvements in the fields of SRMR applied to adaptive and assistive furniture.

The manuscript is structured as follows. It first details the hardware and software modifications of the newest generation of Roombots and then reports significant demonstrations in hardware of selected scenarios. It then concludes with the limitations of the current platform and with a general discussion of
195 challenges of self-reconfigurable modular robot systems.

2. Related work

Over the years, a large variety of 2D and 3D proof-of-concept SRMR systems has been developed. A comprehensive list of both mechanical designs and algorithms of SRMR can be found in [2] and [21]. If research includes experiments
200 with prototype hardware modules, the basic functions of connection and disconnection are in many cases shown with a minimal number of modules, often in an isolated experiment involving two to five modules. Larger reconfigurations with more modules are usually shown in simulation, whereas structures in hardware experiments with more modules (e.g. some form of collaborative behavior)
205 are mostly manually assembled, e.g. PolyBot [22], SMORES [23], UBot [24], SUPERBOT [25], CoSMO [26], 3D M-Blocks [27], Soldercubes [28] and AMAS [29].

With Roombots, the focus is not only task execution with prepared structures, but also the actual formation process in 3D. While some of the systems in
210 [2] and [21] certainly will mature further and with additional development will be able to demonstrate larger reconfigurations in hardware, most (if not all) SRMR systems have difficulties when the number of modules is increased, especially for 3D systems. Because of significant hardware challenges, only a limited number of such SRMR systems are able to perform 3D self-reconfiguration with more
215 than five modules where all modules are involved in the self-reconfiguration, allowing for a more comprehensive understanding of the potential of reconfigurable systems.

As Roombots are thought to work in groups that can involve up to tens of

modules (task 1), we more closely inspect the largest structures formed through
220 self-reconfiguration with 3D SRMRs, and could only find three examples where
the formation or reconfiguration is explicitly stated in literature: (i) M-TRAN
III [30] formed a mesh structure with 24 modules (48 DoF) that locomotes
by disconnecting modules from one side of the mesh and reattaching them on
the other side, (ii) ATRON presents a self-reconfiguration sequence with three,
225 three-unit meta-modules in [31], resulting in 9 modules (9 DoF), and (iii) EP-
SMORES [32] demonstrate a self-reconfiguration sequence with 11 modules (44
DoF), although only 4 modules exhibit attaching and detaching during the
process (more reconfiguration work of EP-SMORES is presented in [33] and
[34]). We aim to show that Roombots is one of the few systems where 3D
230 reconfiguration with 10 modules (30 DoF) and more is possible.

For controlling distributed systems, there are specifically designed program-
ming languages such as Protelis [35], Meld [36], Dolphin [37] and Buzz [38] for
handling certain difficulties in aggregate programming. Although we are not us-
ing any of those for planning purposes, they could potentially benefit SRMRs.
235 We are mostly using conventional C, C++ and C# because these languages are
widely embraced with the SRMR field and well suited for planning algorithms.

Challenges still remain even after a structure is formed. The shape-shifting
capabilities of SRMRs raise the question of how humans can interact with such
complex robot systems [39]. For humans, vision is one of the most powerful
240 senses that enables accurate remote sensing and allows for interaction. Partic-
ularly the sense of depth plays a crucial role in almost all of our fundamental
sub-tasks such as grasping and locomotion. Similar to humans, robotics systems
can benefit from vision too. One of the easiest ways of getting the depth map
information is by using RGB-D cameras. There exists numerous ways to utilize
245 RGB-D cameras for human tracking [40, 41, 42], detecting gestures [43] as well
as other human motion [44] or learning object affordances [45]. Modular robots
can benefit from vision as well, although the integration of a vision system into
a modular robot system is not straightforward. There are several possibilities
(e.g. integrated in a module, a specialized module, an external system), each

250 with their own advantages and disadvantages and engineering difficulties. A recent use of RGB-D sensor in the scope of SRMR has been reported in [33] where a system made out of SMORES carries the RGB-D sensor around and uses it for closed-loop self-reconfiguration as well as navigation. The simplest way for tracking however is to keep the RGB-D camera external. Such an external camera could be placed on top of a structure built out of modular robots as e.g. in 255 [46]. However, because the camera in this work is used to track multiple objects in a simulated living space, we use a stationary mounted external camera with an overhead view.

Concerning the vision of providing assistance in everyday tasks, robots started 260 entering houses to do chores much later than their industrial counterparts used in process automation. One of the pioneering assistive robots which became widely available is Roomba [47]. Although consumer robots are not very widespread, there is a deep literature on the research of assistive robots reporting various tasks such as cloth folding [48, 49], social interaction with humans 265 [50, 51, 52], doing a combination of different chores [53, 54, 55] and helping disabled people with an autonomous wheelchair [56] or bipedal robot [57]. All of those robots embody either a partial generic human form, or are specifically designed for a given task. Here, we explore the possibility whether such tasks can be attempted by distributed systems such as SRMRs, specifically in the 270 scope of creating an assistive environment.

Whereas conventional furniture consists of passive elements that stay where they are placed, advancements in technology and robotics call for smarter and interactive houses. Examples include a robotic ottoman system guessing your intention and responding to your needs [58] or a robotic wardrobe to modulate 275 living space to the needs of the user [59]. Thanks to the expansion of the internet of things, there is an increasing number of architectural efforts to sensorize furniture using cameras and other complementary sensors [60] and in designing actuation systems for furniture [61, 62].

The vision of Roombots is to introduce dynamically reconfigurable furniture 280 into our future living space, redefining room arrangements and available

space. In this work, we show proof-of-concepts of functionalities enabled by such reconfigurable furniture, namely dynamic adaptivity and assistance. We are aware that our current work does not contain much quantitative data; we rather report qualitative data by showcasing the many abilities of a reconfigurable system. We believe that such progress is also important to report as it can serve as a benchmark for future iterations and generations of reconfigurable systems.

3. Material and methods

The following subsections describe the different modifications of the RB hardware and software in detail: the adaptations on the module itself, the added sensors and the new control possibilities, the new gripper, added electronics, the Graphical User Interface (GUI), the external RGB-D camera and the passive parts (furniture).

3.1. *Mechanics, motors and connection*

The overall design and dimensions of one Roombots module are kept as the original design described in [10], thus only the main features are described here. One RB module consists of 4 interconnected half-spheres (hemispheres) that can continuously rotate around each other by 3 motor units. Each module has 10 connection plates of which (due to space constraints) two possess actively retractable hooks that can latch onto any other connection plate. This Active Connection Mechanism (ACM) is the mechanism that allows one module to attach to another module or to any surface equipped with our connection plate, forming dynamic multi-module structures such as adaptive furniture. One module contains all necessary electronics for operation, a bluetooth module for communication and batteries, making it fully autonomous. Fig. 2 gives an overview of the design of a single module.

As already pointed out in [10], a stiffer construction and stronger actuators were needed to overcome some of the limitations of the initial design. From

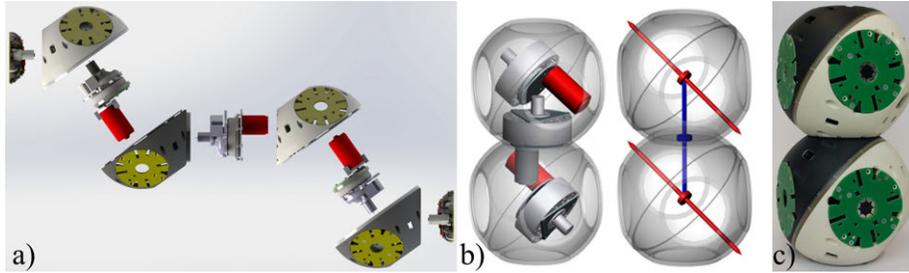


Figure 2: Design of a single module with a) an exploded view of the main components Active Connection Mechanism ACM (2 per module), hemispheres (4 per module) and motor units (3 per module), b) arrangement of the motor units inside the assembled module and orientation of the rotation axes (red and blue axes), and c) fully assembled Roombots module of the newest iteration with dimensions 11x11x22 cm. Parts of this image have been adapted from [6].

previous iterations of the hardware, it became clear that the design needed major
 310 improvements in three areas: output torque, control precision, and docking
 reliability. As a first step, the material of the three main gearboxes that are
 driving each module was changed from plastic to brass together with replacing
 the respective motors to more powerful ones. On the one hand, this significantly
 increased the torque of the Degrees of Freedom (DoFs) of one module (19%
 315 increase for the middle DoF with a Maxon RE-max 24 DC motor to 4.3 Nm,
 and 70% increase for the outer DoF with a Maxon RE 25 DC motor to 8.4
 Nm, [63]). On the other hand, due to the much more precise metal gearboxes,
 the backlash of each gearbox was reduced by a factor of 10 (from 2° to 0.2°)
 which was necessary to improve the control precision and thus the connection
 320 reliability of a module. Many difficulties in the ACM caused by the use of
 3D-printed parts and plastic gearboxes also required a change of material. This
 mechanical latch needs to withstand the full weight of a few connected modules.
 Thus, for the current iteration we decided to also implement metallic gearboxes
 and manufacture the structural parts of the ACM out of aluminum which also
 325 helped to increase the connection reliability of a module.

In an effort to compensate for small misalignments during reconfiguration,

we reconsidered the hybrid ACM mentioned in [10] in which permanent magnets are used to assist the connection action. It consisted of one permanent magnet in the middle of the ACM and connection plates that would need a specific polarity to create attraction, thus unfavorably changing the hermaphrodite ACM into a gendered connection mechanism. We achieved a hermaphrodite connection with permanent magnets by creating a *ring* of magnets whose polarities are oriented in a specific manner (see Fig. 3) such that an attraction is created every 90° which corresponds to how two modules can connect to each other. A standard ring (for a non-ACM connection plate) is formed by 8 disk-shaped magnets with alternating poles facing the surface; due to space constraints caused by the gears of the ACM, a ring in an ACM connection plate only contains 6 magnets. Each ACM and every possible connection plate both of a module and of a passive grid structure has been equipped with this ring of permanent magnets. Taken together, this helps to form an aligned contact between two connection plates that are supposed to be touching. This magnetic connection is strong enough to form a small region of attraction when two plates are close, however it is not capable of transferring any load and the connection is easily broken by any movement of a module. The researches in [30] and [23] use a similar magnetic ring idea as a main connection method. Using permanent magnets as main connection method results in a harder disconnection process. However, the magnet ring is used as only a complementary method to the mechanical ACM in Roombots.

In the end, we built 11 new fully functional modules with the presented hardware modifications (with a total of 13 modules with 2 modules of the previous iteration).

3.2. Sensors

The most important sensorization was done on the rotary joints by adding absolute encoders (12-bit capacitive absolute encoder AMT203). Initially, Roombots had relative encoders which meant that modules were supposed to be powered up only after all joints were manually aligned to a certain angle such as 0° .

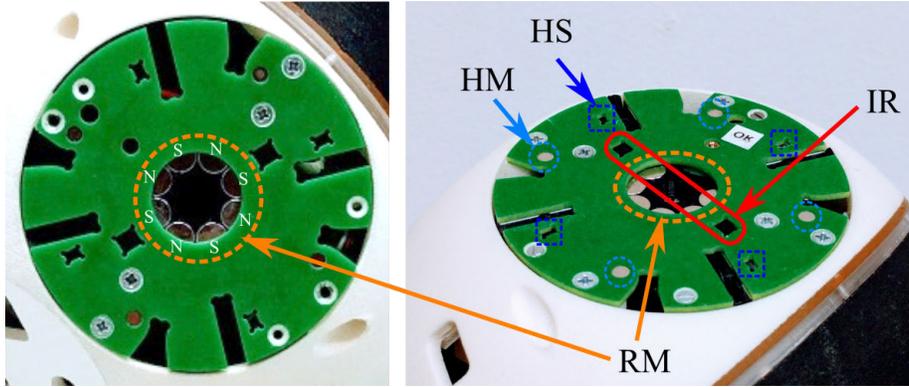


Figure 3: Modifications of the connection plates. In the middle of each plate, a ring of permanent magnets with alternating poles (N and S) facing the surface (RM; 8 magnets for a standard non-ACM connection plate, 6 magnets for an ACM connection plate) assists in forming contact between two plates. Two infrared sensors (IR) give information about the proximity of neighboring plates or the environment whereas four pairs of a hall-effect sensor (HS) and corresponding magnet (HM) indicate if two touching plates are oriented appropriately such that a connection with the ACM can be formed.

New absolute encoders ensure that the control of a module no longer depends on the initial position when switched on, thus the initialization of experiments became easier. Although both relative and absolute encoders could have been
 360 used in conjunction to increase the control precision due to the higher resolution of the relative encoders, they have been removed to provide space for stronger DC motors.

There has been a high demand for proximity sensors on Roombots for various applications, particularly for reliable self-reconfiguration. Each ACM plate is
 365 equipped with two infrared (QRE1113GR) and four single-axis linear hall-effect sensors (DRV5053RA) (shown as IR and HS respectively in Fig. 3). Infrared sensors detect existence and distance of a nearby object within the range of 1 mm to 15 mm. The reading of the infrared sensor relies on the reflectivity and orientation of a facing surface. In contrast, hall effect sensors report the
 370 magnetic flux, Φ_B , crossing the sensor area which correlates to the proximity of a magnet. The actual Φ_B reading depends on distance, strength, orientation

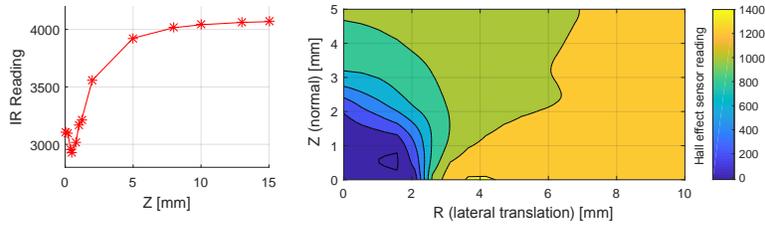


Figure 4: Sensor responses of infrared sensor (IR) when docking surface is getting further away (left) and hall effect sensor (HS) response with increasing normal distance Z to the magnet, and lateral translation R (right).

and misalignment of a facing magnet. Every possible docking surface (matching surface of ACMs) is equipped with four cylindrical magnets with a diameter of 3 mm (illustrated as HM in Fig. 3). All four magnets are situated right in front of the hall effect sensors when an ACM is connected. Thus, the viability of docking can be assessed. Sensor readings of infrared and hall-effect sensors are presented in Fig. 4. Additionally, each ACM is equipped with a three-axis accelerometer (ADXL337) to detect the gravity vector. The gravity vector helps to detect external disturbances or internally accumulated errors such as gear backlashes, manufacturing/assembly misalignments or local control imperfections. The use of similar sensors to increase the connection reliability has also been reported in other modular robots thanks to their cost effectiveness and simple integration. For instance Sambot in [64] uses infrared sensors, accelerometer and gyroscope integrated in modules whereas [65] presents a non-actuated sensor module just for carrying linear hall effect sensors and a camera. Because such sensors already have been successful integrated into these similar robot systems, we decided to adapt this solution to our case to save development time of a custom sensor solution.

3.3. Spotlight LED

A powerful RGBW-spotlight LED board has been developed to illuminate a workspace whenever the ambient lighting is insufficient. This spotlight can be integrated within any Roombots module. Particularly when Roombots need to

incorporate a camera, the environment needs to be well-lit. Whenever there is a lack of light, a module can carry the spotlight to the desired location and enable
395 the computer vision approaches. Furthermore, such a colourful and strong light can be used to induce emotions by playing with color and intensity functions. For example creating a sunset effect in a bedroom without any windows or fireplace engenders good feeling in a living room without a fireplace.

3.4. LED rings

400 For SRMRs, colored light is one of the easiest method to give feedback to users or even as a channel for communication between modules. In some SRMRs, e.g. SolderCubes [28], there is a specific light module dedicated to these functions. Colored light can also be part of the module design as e.g. in [66], and Roombots follows this design. We implemented LED rings around
405 each of the outer DoFs, consisting of 6 RGBW LEDs each. A prototype of these rings is presented in [17] for a dedicated set of modules; an upgraded version has now become the standard for all modules. With it, different states of the module can be assigned to light colors, light intensities and various blinking patterns, which can be used for feedback to the operator.

410 3.5. Universal Gripper

In the vision of Roombots as depicted in Fig. 1, the functionality of object manipulation had to be implemented in the Roombots platform. There were essentially two options available: a manipulator either could be designed as a standalone system that can be attached to a module with the ACM, or it can
415 be implemented directly into a module. For the first option, the method of manipulation is more open since the space needed for it is not restricted as a standalone gripper unit could have arbitrary dimensions. As a disadvantage, a new power management and communication line would have to be designed for such a system. For the latter option, only methods that do not require much
420 space can be considered as everything has to fit into one hemisphere, however power and communication can be shared from the module itself. This advantage

led to the decision to integrate a manipulator into a module, and thus an appropriate gripper technology had to be found with two requirements: (i) to work in limited space and (ii) to be able to manipulate a variety of everyday objects (glasses, remote control, USB stick, cutlery, etc.). After much consideration, the interesting concept of granular jamming was used to develop a “Universal Gripper” (UG) similar to the one in [18]. This gripper consists of a closed, flexible membrane (often a party latex balloon) filled with small granules (e.g. sand, ground coffee, etc.). The membrane is normally soft and adapts to the shape of an object when pushed onto it. The actual gripping is achieved by transitioning the fluid state of the granules into the solid (jammed) state by creating a vacuum inside the membrane. This effectively locks the shape of the membrane which now is able to exert a gripping force on the object. Due to this interplay of a soft, shape-adapting state and a solid, gripping state, this gripper is capable of manipulating a wide range of objects as demonstrated in various previous works (e.g. [18, 67, 68]).

Integrating a UG into a Roombots hemisphere still posed a number of challenges, mostly around downsizing the mechatronic components, especially the required and usually large vacuum pump. A careful selection of a miniature vacuum pump (Schwarzer SP 100 EC) and small-scale solenoid valves (SMC S070C-SAG-32) resulted, to the best of our knowledge, in the first-of-its-kind mobile jamming gripper integration within an MR. Two modules have been equipped with this type of gripper which is filled with ground coffee as in the original version in [18]. In addition of being able to manipulate various objects, the control of the gripper is extremely simple and robust where the vacuum pump - with the solenoids - either inflates or deflates the membrane to set pressures, measured by a single pressure sensor (Honeywell 015PAAA5). Details of the implementation are depicted in Fig. 5 left, and the final Roombots gripper module is shown in Fig. 5 on the right. Although the Universal Gripper is very capable, it has a few limitations such as the object size and weight which depends on the size of the gripper, pressure in the membrane chamber, and type of granules and membrane. Furthermore, it needs to press against the ob-

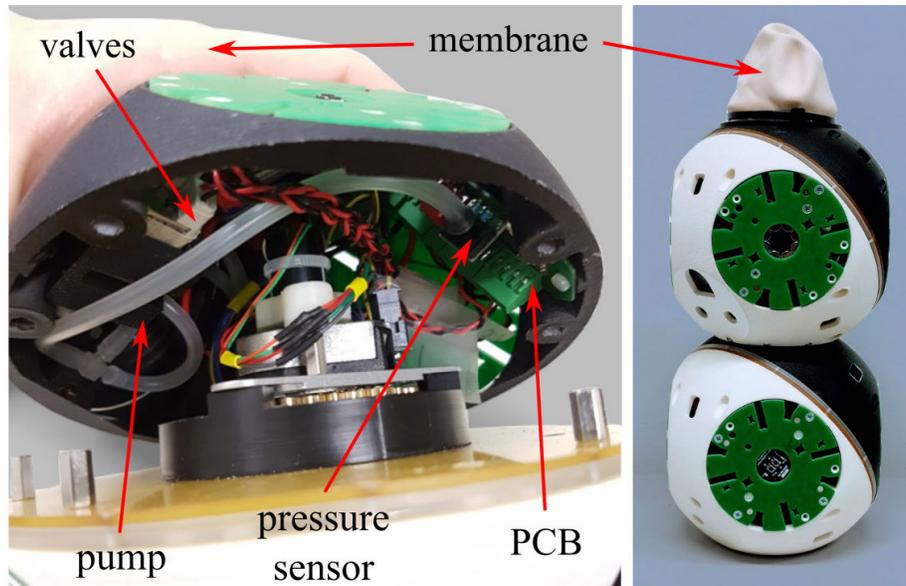


Figure 5: Integration of a Universal Gripper into a Roombots hemisphere. Left: a miniature vacuum pump and small-scale solenoid valves allowed the components to be fully contained and a specialized gripper PCB alike the existing electronics measures and controls the pressure inside the flexible membrane. Right: a gripper module with the Universal Gripper sticking out on top, replacing one of the ACMs.

ject, which can be difficult with a limited DoF system. Hence, it is not always suitable to handle standing objects that have sensitive balance, or soft objects.

455 3.6. Electronics

A considerable part of the electronics was redesigned to accommodate for hardware changes (see Fig. A.20 for details). The power board, motor control boards and ACM control boards went through major updates. The technology of communication with the modules (Bluetooth) as well as the communication
 460 bus inside a module (RS-485) were kept the same. Three new board types were designed to control the additional hardware. The hemisphere with an integrated Universal Gripper contains a gripper board that is responsible for controlling the pressure inside the membrane by using a pressure sensor, vacuum pump and valves. Further, a specialized spotlight board controls the powerful RGBW-LED

465 integrated in one hemisphere which is used for illumination purposes in different colors and brightness. Finally, an LED-ring board gives feedback to a nearby user (a prototype of this board has been developed in [17]).

Further, readout boards for other new sensors have been integrated in Room-
bots. The values of the proximity sensors and accelerometer are not directly
470 used by modules, however they can be reported to an external PC which can execute higher level controllers (as e.g. in [69] for a local connection search algorithm which is briefly summarized in section 3.8).

3.7. Motor control

Changing the hardware led to changes in the firmware of various boards in
475 Roombots. The nature of the new absolute encoders is that they give a much lower number of ticks per rotation than relative encoders which particularly affected the derivative control. Hence, the frequency of the control loop was reduced and PID parameters were adapted. Absolute encoders read over SPI and the communication channel is going through slip rings. Although the commu-
480 nication with absolute encoders is mostly smooth, rare encoder reading errors can occur. Various reading error detection and safety procedures have been implemented. For instance, all the DC motors are disabled if the position error gets unexpectedly large which may occur due to external disturbances or persistent erroneous encoder readings. Other procedures include e.g. implementing
485 timeouts if a readout gets stuck and keeping the last read sensor value in case of communication loss. Such simple procedures are case-specific and should be relatively easy to implement in other robot systems when needed.

Due to safety considerations, the motors operate at roughly 1/3rd of their full speed. This gives us time to manually intervene in the reconfiguration process in
490 case of issues, by e.g. sending a “motor shutdown” command that immediately disables all motors of all connected modules. While this considerably slows down the task execution time in the demonstrations, we valued the safe and successful task execution higher than execution speed. This point will be discussed in section 5.

495 *3.8. Connection control*

The proximity sensors and accelerometer data can be used to check the validity of a connection. A connection can only be formed if two plates are close enough and correctly oriented which can be checked by the IR-sensors and hall effect sensors. Additionally, in a lattice-type configuration, the gravity vector
500 must be collinear to either the global x-, y- or z-coordinate, measured by the accelerometer; other directions indicate non-idealities such as body elasticity, local control error, or gear and ACM backlashes.

These properties are used in [69] to develop a local search algorithm for reliable connections that has two stages. In our case, the condition for a successful
505 connection is the proper alignment of the four hall effect sensors. Since their values are maximal when directly opposite a magnet, it is enough to threshold the sum of the four sensors to validate a possible connection. If the sum is above the threshold when a connection sequence is initiated, the command list continues normally. If the sum is below the threshold, the first stage of the local
510 search engages, where the accelerometer data is converted into a *pitch* angle. The pitch then is used to generate motor commands for a new desired angle by an inverse kinematics model of the module (Fig. 6 a)) to reorient the plates orthogonally. If this is still not satisfying the threshold condition due to elasticity in the module and docking plates, the second stage with random movements
515 engages. This is because the occurrence of ACM misalignments strongly depends on the movement history of each module such that modeling this effect is infeasible. Instead, one of the three motors is randomly chosen and moves by either +1 or -1 degree, followed by a new evaluation of the hall effect sensor values. This process is repeated until the threshold condition is satisfied.

520 While locomoting on a plane, a single module can only reach two adjacent targets. E.g starting from the position in Fig. 6 b.1), a module can only reach the position in b.2) (“UP”) and b.3) (“LEFT”). Similarly, starting from the position in Fig. 6 c.1), a module can reach the position in c.2) (“DOWN”) and c.3) (“RIGHT”). Gravity affects these sequences differently, depending on
525 the locomotion plane. On ground, gravity doesn’t deviate the module from the

target position and actually assists the connection. On a wall, the UP, LEFT and RIGHT sequences are negatively influenced by gravity where a connection control is needed. On a ceiling, all sequences are similarly influenced by gravity and need the assistance of a connection control.

530 The result of this local search algorithm was a drastic improvement of single-module on-grid locomotion. While one module before could only locomote on a tabletop with 100% success rate, it had a 0% success rate for certain movements on a wall or ceiling. The new connection control can handle all cases. We tested three of the most challenging scenarios, each initialized with the motors at zero
535 position and one ACM of the module attached to the grid: (i) the module is hanging from the ceiling and connects the second ACM to the ceiling (“CEIL”), (ii) the modules is attached to a wall and connects the second ACM to the connector directly above (“UP”), and (iii) the module is attached to a wall and connects the second ACM to the connector on the left (“LEFT”). Each scenario
540 was repeated 10 times. The new docking position search method works with a 100% reliability for all movements, however it can be somewhat slow. The average convergence times for a connection by random movement are reported in the boxplot in Fig. 6 d) and range from 20 s (“LEFT”) to 39 s (“CEIL”) to 90 s (“UP”). Nevertheless, the improved reliability in the connection forming
545 process is a crucial property for any self-reconfiguration with the hardware.

3.9. Graphical User Interface (GUI)

For designing different complex tasks as outlined in the introduction, an easy-to-use User Interface (UI) was needed. Different user interfaces with Roombots were already explored in the past, e.g. using Playdough to form structures [70]
550 and an approach based on Virtual Reality (VR) [20], however these previous works could not be adapted to larger scale dynamic formations. The Playdough interface does not work in real-time, and the VR only allows a limited kinematic range for reasons of simplicity. Since Roombots are able to form complex geometrical structures, it seemed natural to develop a Graphical User Interface
555 (GUI) to be able to visualize such formations. This GUI has been developed with

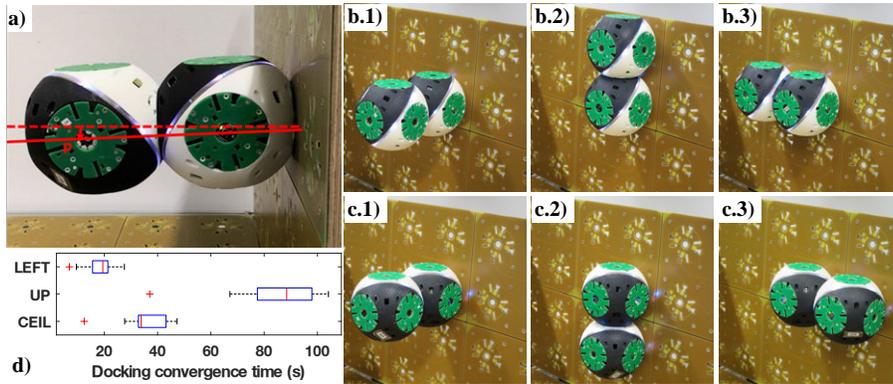


Figure 6: Single module connection with pitch angle compensation and random search. a) Pitch angle compensation. The end effector pose of a module can exhibit deviations from the desired pose due to external disturbances such as gravity. The accelerometer detects such deviations (red solid line) from the global orthogonal frame (red dashed line) and uses an inverse kinematics model of the module to correct imperfect attachments and module deflections. If this compensation does not result in a valid connection, a second stage with small random module movements engages, aiming at aligning the hall-effect sensors with the corresponding connection plate. b) and c) show the possible connection sequences of a single module when attached to a plane. d) Average convergence times for the connection by random search for three different cases.

Unity¹, a 3D animation software that often is used for creating video games. Even though Unity has the capability to function as a physics simulator, we only use the visualization aspect of the software. Naturally, the visualization does not always represent the real configuration of modules perfectly due to the physics present in the real world, causing e.g. elastic deformation of modules and attachment plates under load. This effect as well as others arguably could be modeled and hence included in the visualization. However, through our experience with the real modules, we found these effects and other sources of noise to be too stochastic for modeling and decided to implement control routines in the modules to account for discrepancies. These routines will be discussed below in section 4.1; the GUI itself was kept simple and assumes perfect conditions. At

¹<https://unity3d.com>

the current state of the project, our system focuses on forming viable connections of single modules, for which the development of these routines was feasible. Connection sequences of multi-module formations likely will nevertheless need
570 additional modeling effort, which remains a future challenge.

Fig. 7 and 8 show screenshots of an example usage of the GUI. There are three main parts: control panel, visualization panel and message panel. The control panel has three sub-categories: manual commands, operation mode and list of modules. In “manual commands”, single commands (motor, ACM, LED,
575 gripper, spotlight) can be created and sent to specific modules. These commands can also be added to a list which is created on the fly. In “operation mode”, this list of commands is stored, and it can be executed in a single sequence to perform a specific task. Command lists also can be saved and loaded here. This category also allows the execution of predefined movements of a Central Pattern
580 Generator (CPG)[71] as well as executing external plug-ins, e.g. control of a motor with a computer keyboard. The yellow border around the panel indicates that the control has been switched from the virtual modules to the actual real world modules. Connected modules provide real-time feedback of all the motor and sensor states which then are displayed in the GUI. In “list of modules”, all
585 virtual modules are listed and modules can be created or deleted. The current state and positions of all modules can be exported as a “scene”, and scenes can also be imported again, allowing a quick setup for new tasks. Here, each virtual module can also be connected to a real module by bluetooth, which then allows an easy switch between performing a list of commands with the virtual modules
590 only or also with the real connected modules. The checkbox next to each module enables the display of the virtual or - if connected to a real module - the real module ID next to the corresponding module in the visualization panel. In the visualization panel, an arrangement of virtual Roombots modules is visible. This panel also contains an (empty) message panel to display warnings and
595 other messages (not shown here).



Figure 7: Screenshots of parts of the GUI. The control panel can be switched between “manual commands” where specific commands can be sent to single modules, “operation mode” where a sequence of commands can be played and additional modes are available (transfer of parameters of a CPG network, and external plug-ins).

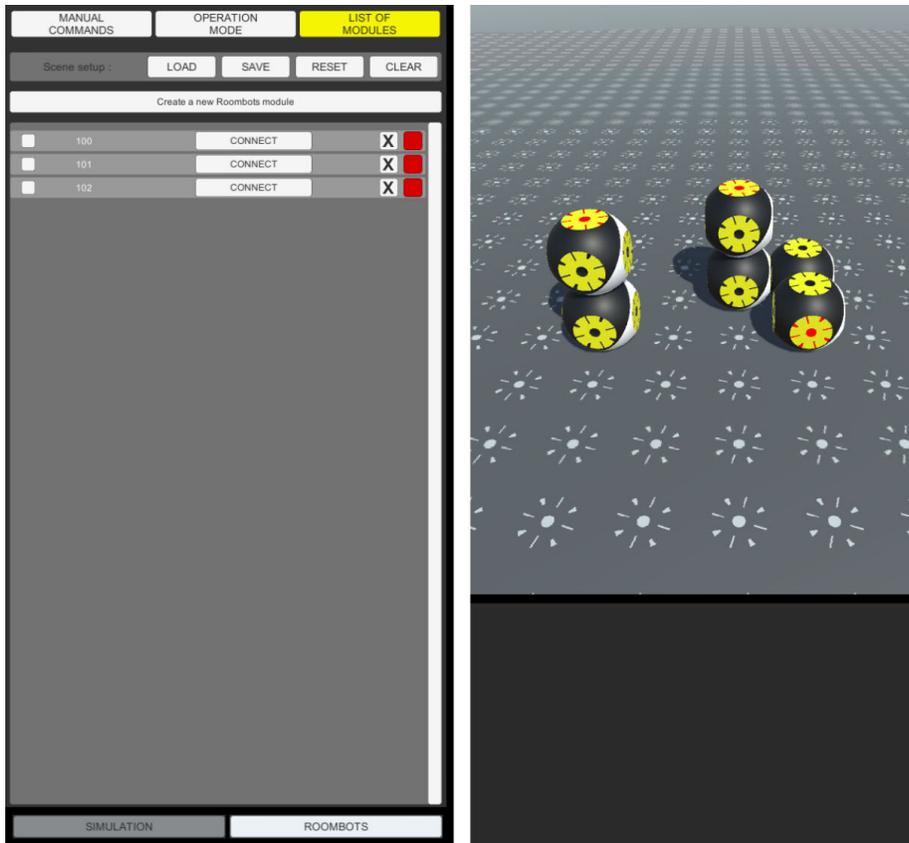


Figure 8: Screenshots of parts of the GUI. Additionally, a “list of modules” panel is available where all modules are listed and can be connected by bluetooth to real Roombots modules. The last panel is the visualization and shows a Roombots arrangement, and a message panel below can display various information during runtime (not shown here).

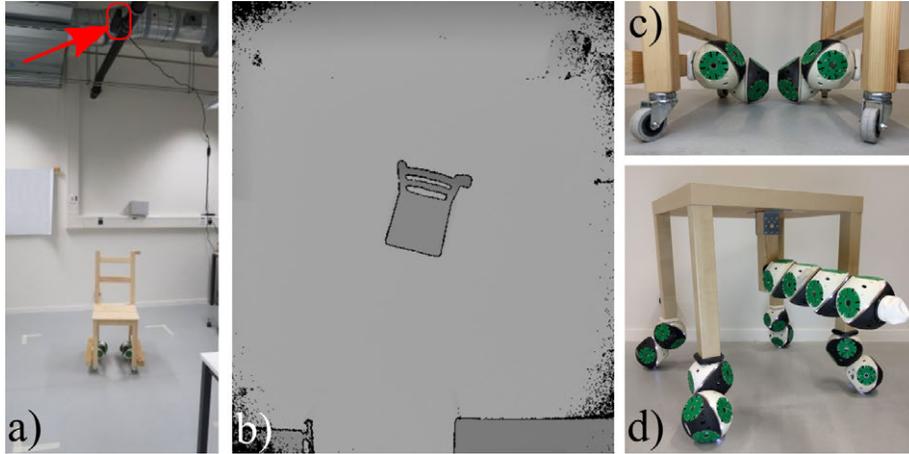


Figure 9: RGB-D camera and furniture used in the sub-tasks. a) An overhead Microsoft Kinect depth sensor (red arrow) can track objects within the area marked with masking tape. A chair is shown in this area. b) View of the camera; depth is mapped onto pixel gray-scale with black edges, making the chair easily identifiable. c) Close-up view of the caster wheels and the differential drive-like wheel modules of the chair. d) The table with four wheel modules and a meta-module with a gripper attached underneath.

3.10. RGB-D vision system

Some of the proposed tasks require an external system that tracks a user and furniture present in a setup. We have already used the Microsoft Kinect sensor in previous works for tracking purposes and reimplemented the same system also for the current work. Here, a Kinect v2 sensor is mounted on the ceiling and observes a designated ground area. It functions as a depth sensor and can distinguish objects with different heights in the scene. It is assumed that the highest object in the scene is the head of a user, and multiple pieces of furniture are marked in specific ways to make them unique and define their orientation. Fig. 9 a) and b) give an overview of the tracking system. A recent use of RGB-D sensor in the scope of SRMR has been reported in [33] where a system made out of Smores carry the RGB-D sensor around and use it for closed loop self reconfiguration as well as navigation.

3.11. *Robotic-enhanced furniture*

610 As two example pieces of furniture to demonstrate the proposed tasks, we equipped a small table and a chair with Roombots modules to extend their functionalities. The chair is put on passive caster wheels such that it can roll in any direction, and two modules use their outer hemispheres as wheels to drive the chair similar to a differential drive. The motor positions of the two
615 modules are chosen manually, and rubber bands wrapped around the respective hemisphere shells are used to produce an adequate friction between the shells and the ground. Since the main load is borne by the passive wheels, a user can sit on the chair, however the chair is only moved by the modules under no additional load. For the table, each leg is extended by one module, and by again using
620 the outer hemispheres for rolling, it gives the table the ability to move forward and backward and to rotate. Additionally, a gripper structure consisting of one gripper module in series with a standard module can be mounted underneath the table. This allows the table to pick up objects from the ground and place them onto itself as will be shown later. Fig. 9 c) and d) show both pieces of
625 furniture.

4. **Experimental results**

To demonstrate the augmented capabilities of the modified RB system, we designed five main tasks which are demonstrated in various sub-tasks. The results of these experiments are presented in this section and the link to the
630 supplementary video including all described experiments can be found in Tab. A.1.

4.1. *Easy-to-use user interface: the GUI*

To easier understand how some of the following demonstrations were developed, we present the GUI first. As an example to present the work flow of the
635 GUI from task definition to hardware execution, we here describe the development of a simple demonstration that we call “circle walk”, i.e. a sequence

of movement commands that make a module locomote on-grid in a circular manner.

At the start, the GUI visualization area is empty, and a new virtual RB
640 module is added. A module usually is initialized with both ACMs opened and
all motors in their zero state, and by default is placed in the GUI vertically onto
the grid with one ACM facing the grid. First, an initialization script is loaded
into the GUI that attaches the module to the grid and brings the motors in the
correct position to start the circular loop. Then, the main list is loaded and
645 the loop-box ticked which makes the full sequence loop as long as needed. This
main list of commands consists of a predefined series of opening and closing
ACMs and performing 90° rotations of motors in between where looping these
actions makes the module move in a circular fashion on a single 2x2 grid plate.

Once the demonstration has been developed in the GUI, it is only a matter
650 of connecting a real module to the virtual module in the GUI. Once connected,
the mode is switched from “Simulation” to “Roombots” to send commands to
the bluetooth connection, and the demonstration is performed with the hard-
ware (Fig. 10). To increase the rate of success for executing commands with
the real modules, additional control routines are implemented. The GUI con-
655 tinuously crosschecks the setpoints of motors and ACMs with the actual values
fed back from the module and resends commands for values that do not lie
within a small margin. In particular, the most crucial part of a reconfiguration
is forming the connection to a new attachment point. Handling misalignments
during this process required more elaborate strategies. In our case, the sensors
660 implemented on the ACM plates give the necessary information concerning the
correct alignment. If a misalignment is detected, a search algorithm performs
small, random actions around the setpoints that usually results in a successful
connection. Details of this algorithm can be found in [69].

The GUI possesses more features, e.g. forcing the starting point for the
665 forward kinematics of connected RB modules in simulation, adding pauses to a
sequence of commands, running CPG networks and external plugins and more.
The described example only present the basic functions of the GUI which are

enough to make the development of simple demonstrations relatively quick and easy, both with virtual modules in simulation and with real modules in hardware. Moreover, even though each module can separately be connected to the GUI and thus a “mixed-reality” setting with real and virtual modules in parallel could be created, we did not consider this case at this stage. The switch from “Simulation” to “Roombots” converts all commands in the command list to hardware-accepted versions, upon which virtual modules cannot be controlled anymore. A differentiation to allow the mixed-reality approach and the new possibilities enabled by it is work of future research.

Research on most of the (SR)MR systems begins with simulation to study the feasibility of the system because simulation is easier and faster to set up compared to robot hardware and can give valuable information. Moreover, very complex scenarios can be explored in simulation. Most of the modular robotic systems are accompanied with simulation/GUI tools. For instance, [72] explains a simulation environment for a modular system whereas [73] includes self-reconfiguration steps too. One of the most comprehensive and recent SRMR GUI (VSPARC) is designed for SMORES in [74]. It uses the Unity engine similar to our approach. Most of the features presented for VSPARC also exists in the Roombots GUI, but adapted for the Roombots hardware.

4.2. Scalable self-reconfiguration

As an example of a larger self-reconfiguration task (i.e. with more modules and more steps) that goes together with the application of adaptive furniture, we show the formation of a small chair with 12 modules. The details of the planning of this demonstration can be found in [69]. The work includes the development of a local search algorithm and closed-loop control during the connection process that enabled more reliable reconfiguration, as well as the derivation of an A* search algorithm that finds the motor command sequence to form a given structure from arbitrary initial conditions.

Here, only the final formation sequence is shown, starting from 12 modules standing on a flat grid. In order to decrease the number of nodes on the search

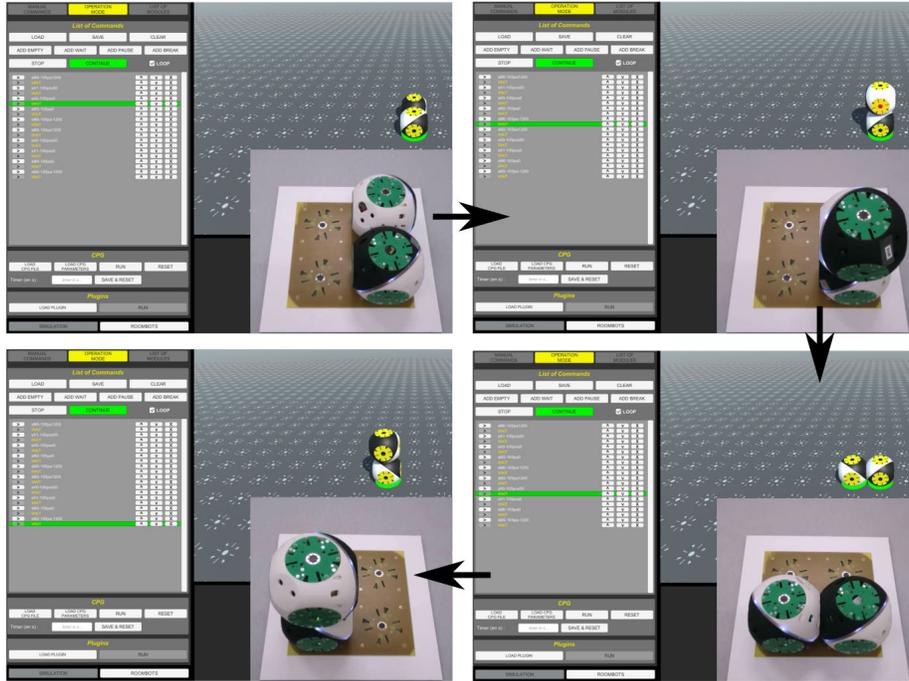


Figure 10: Circle walk demonstration. The GUI is shown in “operation mode” where a preprogrammed list of commands is executed in sequence. The first half of the circle walk is shown; looping the script makes one module go around on a 2x2 grid. The corresponding hardware configuration is shown as insets.

tree for such a large number of possible motor actions at each state, pruning methods are included in the search algorithm to avoid exploring unpromising branches. The formation of the full chair is divided in to 6 sub-configurations (manually predefined by an operator) that must be reached in a specified order and build up the chair in stages. This is needed for the search algorithm to find a solution in under 30 minutes for a reconfiguration sequence that involves more than 4 modules. Fig. 11 shows snapshots of the formation process.

Although M-TRAN, ATRON, EP-SMORES and various other SRMR are potentially capable of showing a similar application shown in Fig. 11, to the best of our knowledge this is the first time a 3D self-reconfiguration is shown using more than 30 DOF (36) with all modules initially separated (however

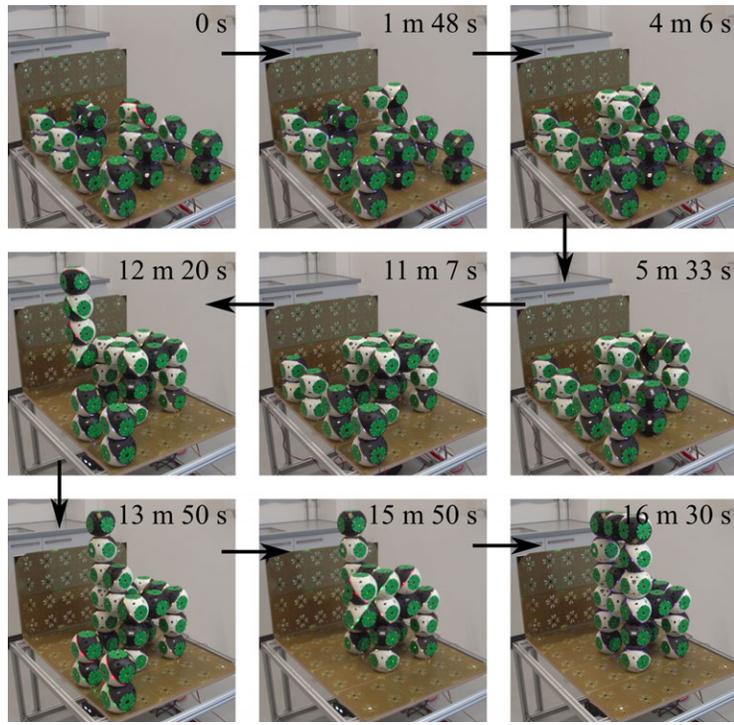


Figure 11: Snapshots of the chair formation sequence. The chair is built in 6 stages (4 for the legs and 2 for the backrest). The large time loss in the middle is caused by a battery change and a manual intervention to connect the leg segments, however as shown in the supplementary video, the self-reconfiguration is largely autonomous.

positioned and attached to a grid on which the structure is formed).

710 4.3. Mobile furniture

The task of mobile furniture is demonstrated in four sub-tasks: following chair, evading chair, adaptation to environment and overcoming obstacles.

4.3.1. Following chair

We explore the possibility of an assistive chair that follows a user to always be
 715 available in case of the user needing to rest. The system is aimed at preventing falls and providing assistance for manual recovery after a fall. It consists of an RGB-D camera tracking the position and orientation of the user and of the chair

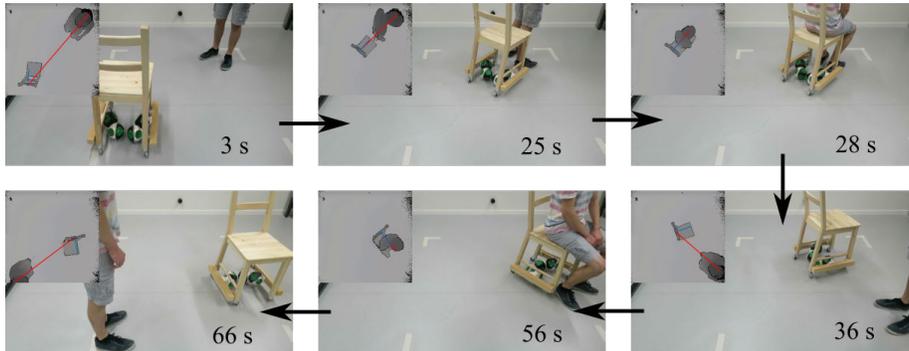


Figure 12: Following chair. An overhead Kinect (top left panels in each snapshot) tracks the position and orientation of an assistive chair (short blue line) and the path to the patient (long red line). The user can sit whenever needed as the chair follows the user inside the predefined area.

in a predefined area. Two Roombots modules are attached to the chair allowing it to move anywhere and in any orientation in the area, similar to a two-wheeled differential drive robot. A controller (PC) computes the difference between the position and orientation of the user and the chair and sends movement commands to the modules such that the chair is always next to the user (Fig. 12).

4.3.2. Evading chair

We also demonstrate the inverse scenario where Roombots furniture has to make space to let a user pass. This scenario could take place in a packed apartment where the available space is limited and shared between user and furniture, or a user is physically unable (e.g. in a wheelchair) to move furniture out of the way. We again used the depth map to track a user and the same chair as in the previous demonstration. The chair initially is located in the middle of a defined area and the user would like to cross this area. A controller (PC) extrapolates the desired path of the user and moves the chair orthogonally to this path out of the way to make space for the user to pass. After creating enough space and the user passing, the chair moves back to its original position (Fig. 13).

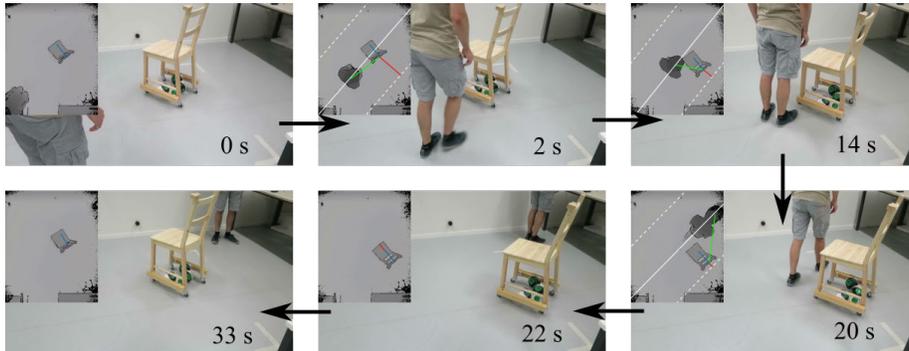


Figure 13: Assistive furniture makes space. An overhead RGB-D camera (top left panels in each snapshot) tracks an assistive chair (short blue line) and the users location with respect to the chair (green line). It extrapolates the path of the user (white lines) and moves the chair orthogonally to it (short red line) before returning to its original position after letting the user pass.

It is important to note that both “following and evading chair” demonstrations are not complete and comprehensive solutions to solve the problem. The full solution would require significantly more research which falls out of the scope of this paper and the SRMR field. Such a solution would involve detailed intention detection considering a much wider spectrum of actions and would incorporate safety and emergency protocols. Nevertheless, they serve as a proof-of-concept that SRMR can be used in such applications.

4.3.3. Adaptation to environment

Roombots-enhanced furniture can have additional functionalities besides providing mobility to existing furniture. In this demonstration, we equip a small table with four modules, fixed to the end of each leg of the table. We use the outer hemispheres (one DoF) as wheels to enable the table to move forward and backward and to rotate. In this specific example, the two other DoFs of a module can move in a null-space such that the total height of a leg can be varied. The table transports a set of objects and this height adjusting ability can be used to keep the tabletop horizontal on uneven terrain to prevent the objects from falling off the tabletop. We let the table drive from a straight area

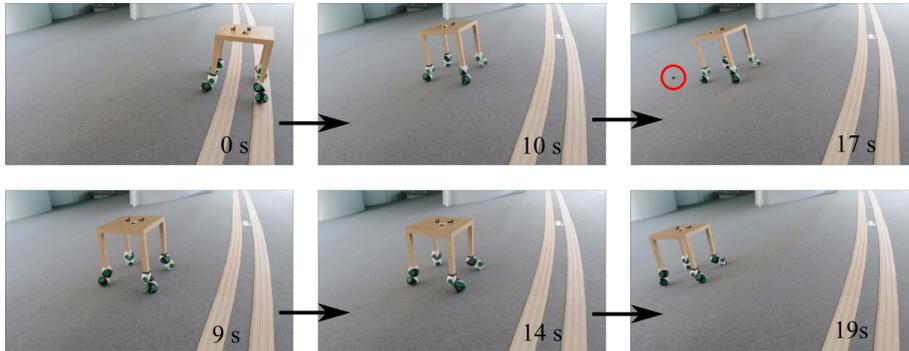


Figure 14: Active slope compensation. Top row: a table driving over a sloped area (left side of double stripes) tilts its tabletop, causing objects on top of it to fall (red circle). Bottom row: the attached RB modules can partly compensate for the slope by shortening the hind legs of the table, keeping the tabletop more horizontally such that the objects stay on top.

into a sloped area. In the first case without adaptation, the set of objects falls off the table after a short distance on the slope area. When the manual slope
 755 compensation is performed, the tabletop can be kept closer to the horizontal plane which prevents the objects from falling when the table is on the sloped area. Fig. 14 depicts snapshots of both cases.

4.3.4. Overcoming obstacles

For the table in the previous demonstration, an additional behavior is avail-
 760 able by “rotating” the RB modules around their attachment point at the end of each leg. This causes a module to move in a circular manner around the tip of the leg, performing a movement similar to a step. We use this movement to demonstrate climbing a small ledge that the table cannot drive over. The table drives towards the ledge until the front legs are aligned and touch the
 765 ledge. The front RB modules then rotate, moving on top of the ledge by doing so. The same process then repeats for the hind legs, after which the table has successfully overcome the ledge (Fig. 15). Here, the table is piloted by a human and the sequences are manually initiated.

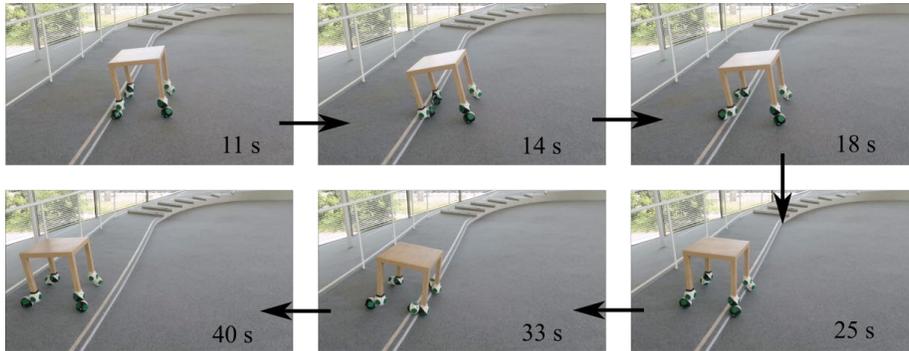


Figure 15: Overcoming a ledge. A table drives until its front legs touch the ledge. The attached RB modules then perform a rotation around their attachment point, causing them to move on top of the ledge in the process. The same movement then is repeated for the hind legs after which the table has overcome the ledge.

4.4. Manipulating furniture

770 One of the most needed and repeated task in daily life (as well as industrial settings) is object manipulation. Hence, it is an implicit requirement for a SRMR to be able to manipulate the surrounding objects, particularly when humans and SRMR need to cohabit.

4.4.1. Object pick-up

775 A small table is equipped with one module on each leg for wheeled locomotion. Additionally, a meta-module structure with an implemented Universal Gripper is attached to the underside of the table. The task is to start at location X, pick up a pen at location Y and bring it to location Z. The table is again manually controlled and rolls into position to pick up the pen. The gripping sequence is preprogrammed and executed once the table is in the correct position,
 780 and the table is piloted by an operator. It picks up the pen from the ground and puts it on top of the table which then rolls to the final location. Fig. 16 shows snapshots of this manipulation task.

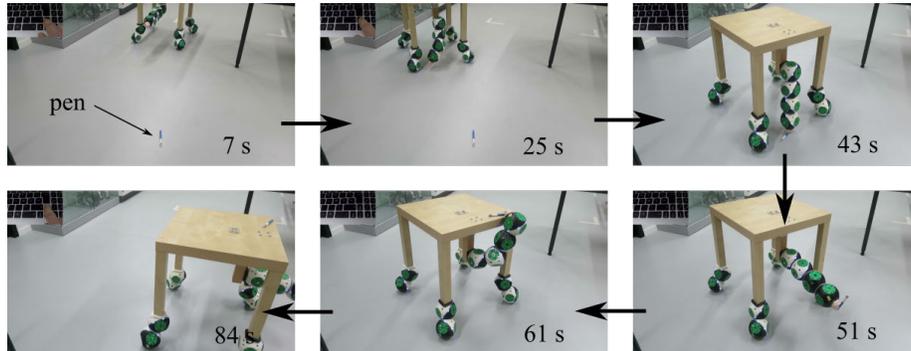


Figure 16: Table picking up an object. An assistive table with manipulating capabilities moves towards a pen dropped on the floor. It picks up the pen with a Universal Gripper, places it on top of the table and brings it to the user.

4.4.2. Passing objects

785 One of the advantages of the Universal Gripper is that it can grip objects basically regardless of their shape and orientation (within a size limit given by the size of the gripper). This is promising for passing objects from one gripper to another since the passing sequence does not need to take the object's shape or orientation into account. In the framework of multiple RB furniture pieces collaborating with each other, such a scenario of passing an object between

790 pieces of furniture can easily be imagined. We adopt this task in a simpler setup where an object is passed from one RB gripper module to another. The first module picks up an object (pen) from a tabletop and rotates its gripper to face the gripper of the second module. In this position, it is important to notice

795 that simply making the two grippers touch orthogonally causes issues for the second gripper to actually grip the object as it would require the surface on the object that is currently occupied by the first gripper. There are several ways to deal with this issue; here we present the preliminary results of inducing a "shift" such that the second gripper touches a free part of the object (this also requires

800 that the object is somewhat elongated to possess such a free part). The second gripper then grips the object upon which the first gripper releases, transferring the object to the second gripper. Fig. 17 shows snapshots of transferring a pen

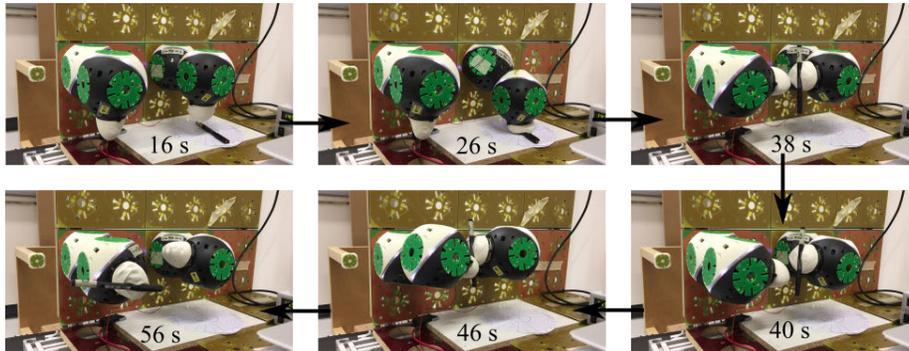


Figure 17: Two RB gripper modules passing a pen in mid-air. The first module picks up the pen from the table and orients it to face the gripper of the second module. A shifting motion then takes place such that the second gripper has a free part of the pen to grab onto. The first module then presses the pen into the gripper of the second module and releases it after the pass is completed.

from one to another module.

Even though we only show one example of such mid-air sensor-less passing
 805 of objects between two modules, we see much potential in using this unique way
 of object transfer in a collaborative environment as depicted in Fig. 1.

4.4.3. Opening water bottle

The third demonstration in this task concerns a more assistive subtask of
 manipulation. Here, we briefly validate if a gripper structure can help with
 810 opening a PET water bottle. A user has to hold the bottle and push it into the
 gripper of a waiting RB structure. The gripper then actively grips the cap of
 the bottle and performs a rotation to open the cap. Once opened, the user can
 remove the bottle and the modules place the cap on the table (Fig. 18).

4.4.4. Object manipulation capabilities of SRMRs

815 Almost all SRMRs have means of connecting and detaching to each other.
 Some use magnetic attraction whereas a big portion of them use mechanisms.
 For example, the Active Connection Mechanism (ACM) is the mechanical latch
 Roombots use for the purpose of connection. In theory, this connection mecha-

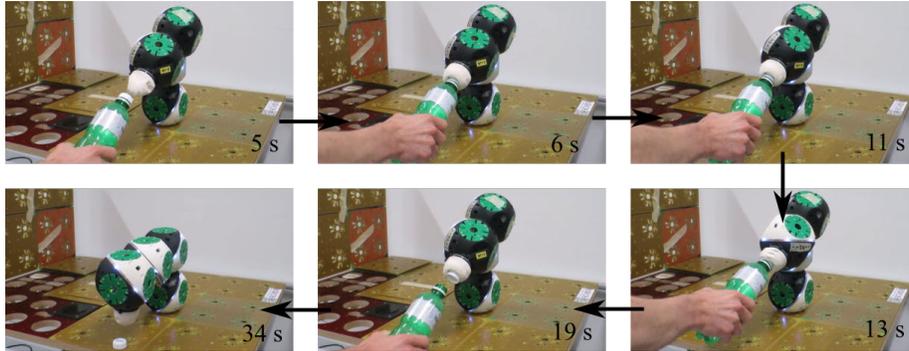


Figure 18: A gripper metamodule assisting in opening a PET water bottle. The user pushes the bottle into the gripper which grips the cap and then performs a rotating movement to open the bottle.

nism - and similar ones in other systems - could also be used for object manipulation. However, most of the times these self-reconfiguration methods are not suitable for manipulating arbitrary objects. It is possible to design a gripper module as an extension for almost all of the SRMRs similar to [31]. Also, some of the SRMRs have already integrated gripper modules such as [66] and [75]. These existing grippers are similar to conventional manipulator end effectors. An integrated universal manipulator is demonstrated for the first time with this work.

4.5. Interactive furniture

4.5.1. Kinect tracking control

We can couple movement control of RBs with the tracking abilities of the Kinect. Using depth information and the body segregation method of the Kinect, we can define e.g. intuitive hand gestures for certain commands. Such a user interface already has been explored in [17], however not in real-time. There, a user would first point to an RB module and then to a desired goal position, and a planning algorithm then computes and executes a movement sequence. Here, we use real-time inverse kinematics to convert the position of a users hand, tracked by the Kinect, into movements commands to make a

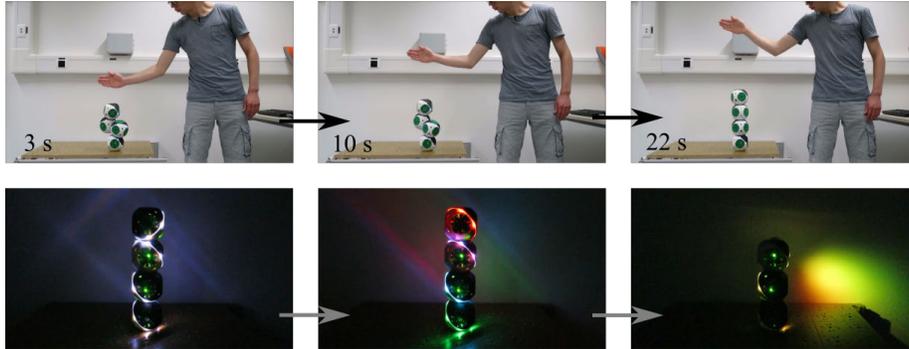


Figure 19: Interaction with tracking and lighting. Top row: a users hand is tracked with the Kinect sensor and its position converted into inverse kinematic commands that make the meta-module follow the hand. Bottom row: lighting capabilities of Roombots with the integrated LED-rings and the powerful RGBW-spotlight.

meta-module follow the hand. The top row of Fig. 19 gives an example of this interaction.

4.5.2. LED capabilities

840 Lastly, we showcase LED lighting capabilities as an interaction method. In the bottom row of Fig. 19, some lighting examples are shown: each module can emit colored light with the two RGBW-LED rings around their outer DoFs, and one module possesses a powerful RGBW-LED spotlight (bottom right in Fig. 19). The spotlight can be used to illuminate a specific region (as already discussed in [19]). For the LED rings, we currently mostly use them to display the overall state of the module and specific events. In most of the demonstrations, the LED rings emit white light, indicating that all components in the module should work as intended. In the self-reconfiguration sequence, more colors are used to communicate the stages of forming a connection. Red light means that
845 the ACM on this side of the module is open (not connected). Blue light means that the random search mentioned in section 3.8 is executed to align the ACM on this side. After this stage, the specific ACM can be closed to form the connection, upon which the light switches to green. Finally, when a planning stage has been fully completed, all the rings of the corresponding modules turn yellow.
850

855 **5. Discussion and Future work**

5.1. Demonstrations of self-reconfiguration and adaptive/assistive furniture

The presented hardware demonstrations showcase the capabilities of the up-graded Roombots modules. We are aware that the outcomes of the demonstrations are rather of qualitative than of quantitative nature as this work aims at
860 exploring a potential use of (SR)MR for adaptive and assistive furniture. In this context, we successfully presented proof-of-concept demonstrations of RB modules completing a large and complex self-reconfiguration task involving a significant number of autonomous connections with minimal human assistance. Further, RB modules are used to create mobile furniture that can follow and
865 evade users, adapt to the environment and can overcome obstacles. Manipulation tasks involved RB modules picking up a pen, opening a PET bottle and a proof-of-concept demonstration of passing objects between two modules. At last, RB modules possess various LED lighting capabilities, and we showed examples of how to use simple gestures to control an RB metamodule and de-
870 veloped a dedicated GUI for easy monitoring of modules and creation of demonstrations.

The challenges for the presented experiments were to increase the connection reliability and to develop an interface to quickly create demonstrations with a large number of modules. For the reliability, on the one hand we could signif-
875 icantly improve the movement precision of all DoFs by reducing the backlash of each gearbox from 2° to 0.2° . This was achieved by changing the materials of the main gearbox and ACM from plastic to aluminum and brass and by integrating a new absolute encoder. The increase in weight was compensated for with stronger motors. On the other hand, reconfiguration sequences now
880 use a closed-loop controller to better align an ACM with a neighboring plate thanks to the integration of infrared and hall-effect sensors. Additionally, a ring of permanent magnets locally helps to form a connection while retaining the hermaphroditism of the ACM. Concerning the interface, the developed GUI contains all the key functionalities to rapidly develop Roombots scenarios. It al-

885 lows for the safe creation of motor sequences in its virtual mode, and due to the
improved connection reliability, the switch from virtual to real modules has a
high chance of correctly performing the demonstration despite the disturbances
present in the real world.

5.2. *Challenges in multi-module structures*

890 While a single module now works effectively, reliable position control of
multiple modules in series remains a challenge. With Roombots, an example
is the necessary more complex on-grid locomotion capability of a meta-module
to perform a self-reconfiguration as in task 1, for which the capabilities of a
single module are too limiting. Even though the mechanics have been upgraded
895 to cause less gearbox backlash, there are two main sources of misalignment
that hinders the autonomy of this functionality: (i) the connection between
two modules possesses unidentified mechanical play, and (ii) two modules in
series still cause significant elastic deformation of the module shells due to the
combined weight.

900 For the experiments in this work, only the chair reconfiguration task in-
volved meta-module reconfiguration where human assistance thus was needed
in a few cases. Nevertheless, we regard the successful demonstration of the
autonomous chair formation using 12 modules as a significant milestone in the
Roombots project. It validates both the capabilities of the hardware and the
905 self-reconfiguration framework. It should be noted that relatively minor human
intervention during the hardware demonstration was needed despite the efforts
for autonomy: three manual interventions were needed in 107 executions of mo-
tor commands and connection or disconnection events (of which 16 events are
independent connections only). In particular, the random local search algorithm
910 to align the ACMs does not handle all possible cases equally well and may not
converge because the positions of the two connecting plates are initially too mis-
aligned due to elastic deformations and mechanical play. As a result, the local
motor actions caused by the algorithm do not allow a meaningful quantitative
comparison of the sensory feedback to steer the connection to a more aligned

915 state, in which case the external assistance was needed. A more elaborate algorithm together with a proper modeling of all physical effects could be able to bring the reconfiguration to true autonomy.

5.3. *Limitations of the Roombots platform*

However, the Roombots robotic platform in its current form has rather
920 reached a limit in terms of complexity and weight, making it difficult to investigate such research. In fact, the formed chair is unlikely to be able to support any additional load as the modules in the legs of the chair already need to support the weight of up to 2.5 additional modules (around 4.5 kg) which we regarded as the border of the structural stability of one module. In terms of
925 module lifting power, in the current iteration one module can easily manipulate one other module (resulting in maximally 6 Nm for the driving DoF) but runs into stall torque when two modules are attached in an extended lever configuration (the outer DoFs have a nominal torque of 8.4 Nm but this configuration would require approximately 15 Nm; details in [63]). The table structure thus is
930 able to carry some additional load (e.g. the attached gripper meta-module), but it is not capable to lift and transport heavier load. Lastly, improvements in the speed of a module need to be taken into consideration, especially for designing a consumer product where the formation of a piece of furniture is likely expected to happen within a shorter time than demonstrated in this work.

935 5.4. *Directions of future research*

Further proof of concept experiments to improve on the points mentioned above would likely require a new platform which could be part of future research in the Roombots project and the topic of SRMRs. Moving to lighter, faster and mechanically less complex modules means rethinking the module design,
940 presumably starting from scratch, developing new technologies for connection, disconnection and module actuation.

However, there are certainly areas for improvement of the current hardware iteration. A major future work within the Roombots project is the integration

of additional structural passive parts. In the full Roombots vision, furniture
945 consists not only of RB modules but also of lightweight structural parts that
will help to reduce the weight of the formations and allow the creation of larger
pieces of furniture such as a table. The addition of such passive parts requires a
close collaboration of RB modules to position them appropriately in a structure
for which new reconfiguration algorithms will have to be developed and tested,
950 such as e.g. in [11].

By operating the upgraded Roombots design at the limitations of the plat-
form through successfully running the many different sub-tasks, we could iden-
tify the following additional hardware and control challenges:

1. Exploration of machine vision capabilities (such as e.g. in visual servoing
955 for autonomous docking) to allow modules to perceive their environment.
This could e.g. allow meta-modules in free space to find and move towards
each other and self-reconfigure independent of an assisting grid.
2. Autonomous adaptation of morphology according to the task, either with
visual feedback and/or other means. This could include e.g. classification
960 processes on images of the surrounding to define the task, and selection of
a structure from an internal library that has the best chances of completing
the task.
3. Exploration of safe human-module interaction for a safe integration of
modular robots into our living space. This could be implemented by e.g.
965 the use of compliant materials in the module, which has to be considered
in the self-reconfiguration. Alternatively, moving from position to torque
control could increase the interaction safety. One could even imagine
implementing an artificial skin around the modules to react to tactile
feedback from users.
- 970 4. Study of distributed control algorithms inspired from biological swarms
such as ants or termites. A preliminary study with Roombots on this
topic is presented in [76].
5. There are numerous ways to achieve easy-to-use UIs for SRMRs. For

example, mixed reality applications can enable functionalities that has
975 not been explored yet in SRMR perspective. A preliminary augmented
reality approach with Roombots has already been explored in [16].

The continuous development of SRMRs gives us insights into current chal-
lenges and limitations of the general concept of reconfigurable systems. In the
long term, we could imagine the technology being used in many more applica-
980 tions, e.g. reconfigurable factory lines where it could be used both for reducing
factory space and manufacture costs through machine reconfiguration, or futur-
istic visions of reconfigurable satellites, space exploration robots or even space
stations where autonomous reconfiguration systems could prove invaluable for
their versatility and robustness. This work is another steppingstone towards the
985 grand vision of SRMRs.

6. Conclusion

In this work, we presented the capabilities of the latest generation of our
self-reconfigurable modular robotic system “Roombots”. We outlined five key
tasks that we consider relevant to the vision of modular robotics for adaptive
990 and assistive furniture and were able to successfully demonstrate various sub-
tasks in hardware. This required various design improvements - especially in
the mechanics of the modules - which are described in detail.

Concerning the demonstrations, modules performed a large-scale self-reconfi-
guration into a chair and provided mobility capabilities - such as following and
995 evading a patient and overcoming obstacles - to off-the-shelf pieces of furniture
by being used similarly to omnidirectional wheels. Specialized gripper modules
were used to demonstrate basic manipulation capabilities such as picking up a
pen, and could show the passing of objects between two modules. Interaction
functionalities with colored LED and with the use of a Kinect depth camera
1000 were presented and we discussed the development of an easy-to-use Graphical
User Interface (GUI) to control and monitor groups of modules.

Much of the presented results are of a qualitative nature with a set of Room-
bots modules achieving a defined sub-task. Our goal in this work was not to
optimize a single task leading to quantitative data to analyze, but to show the
1005 versatility of our SRMR system in many different scenarios. We were interested
in the capabilities as well as limitations of the current iteration of our system
to build new experience in hardware experiments that can be used for future
development of this or another SRMR, creating a benchmark of demonstrations
that other systems can compare to. Even though some of the demonstrations
1010 were partly controlled by a human operator, autonomy is only one aspect of
SRMRs and all of the presented demonstrations have been crucial milestones
for this project.

While the majority of the presented demonstrations is focused on adaptive
and assistive furniture, the tasks in general share the main challenges present in
1015 generic SRMR functionalities in mechanics (connection, disconnection, move-
ment, alignment), electronics (autonomy, communication, sensors), software
(control, computation, robustness, safety) and interaction (user interface, user
feedback). Roombots as a system presents one approach in tackling these vari-
ous and interdisciplinary challenges, which could inspire other systems with the
1020 solutions described here, and we hope that this work stimulates the general field
of modular robots.

As for the future of the Roombots project, we envision the design and inte-
gration of lightweight passive, structural parts that can be used in conjunction
with active RB modules to form larger furniture for our everyday environment.
1025 New collaboration algorithms (i.e. adapting the existing control framework or
creating a new distributed control framework) will need to be developed, espe-
cially regarding safety when interacting with modules, coming one step closer
to the integration of such robotic systems into our living space.

Acknowledgements

1030 We gratefully acknowledge the help of Ahmet Safa Öztürk, Stéphane Bussier, Quentin Golay and Théo-Tim Denisart for their contribution to the Roombots project. Special thanks to Jérémy Blatter for the development of the GUI. Further, we are enormously thankful to François Longchamp for his technical support in the design of the modules.

1035 This project is funded by the Swiss National Science Foundation (Project 200021_153299) and the Fundação para a Ciência e Tecnologia (FCT) agency of Ministry for Education and Science of Portugal (PD/BD/105781/2014).

Appendix A. Additional materials

See Table A.1 and Fig. A.20.

Table A.1: The video is available as supplementary material for this article (see link)

	Description	Location
1	RB sub-tasks	https://tube.switch.ch/videos/8254ba7f

1040 **Vitae (in separate file)**

References

- [1] K. Stoy, D. Brandt, D. J. Christensen, Self-reconfigurable robots: an introduction, Intelligent robotics and autonomous agents, MIT Press, Cambridge, Mass, 2010.
 - 1045 [2] H. Ahmadzadeh, E. Masehian, M. Asadpour, Modular Robotic Systems: Characteristics and Applications, Journal of Intelligent & Robotic Systems (2015) 1–41.
 - [3] P. Swisler, M. Rubenstein, FireAnt: A Modular Robot with Full-Body Continuous Docks, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 6812–6817, iSSN: 2577-087X.
- 1050

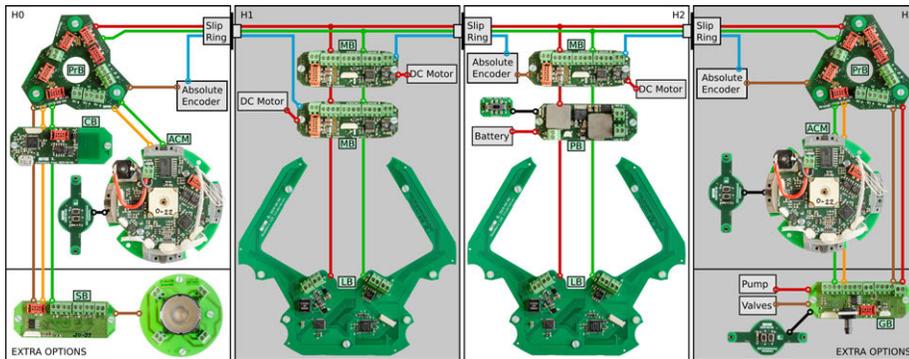


Figure A.20: Roombots electronics is also modular and distributed throughout a module in hemispheres H0-H3. The background of each hemisphere region is shaded to match colors of the real hemispheres. In each Roombots module, there exists 3 motor driver boards (MB), 2 active connection mechanism boards (ACM), 2 proxy boards (PrB, used to convert voltages and ease assembly), 2 LED boards (LB, used to give feedback to the operator), 1 communication board (CB, provides Bluetooth connection and drives the inner RS-485 bus) and 1 power regulation board (PB). Furthermore, each module can support extra options such as spotlight board (SB) or Universal Gripper control board (GB) in the outer hemispheres. Extra option slots are identical and each electro-mechanical subsystem can be placed in either of the outer hemispheres. The extra options are not considered as essential for the basic operation of Roombots modules; only selected modules are equipped with them. All of the boards that are connected to the communication bus except the proxy boards have a dsPIC33FJ128MC802-I/MM to handle low level control tasks. Electrical connections within a module are illustrated with different colors: Red (15V), orange (6V), brown (5V), green (RS-485 bus), blue (SPI) and black (support boards for easy user access). Each physical connection is shown with a circle whereas unconnected lines pass through a board without a circle.

- [4] J. Germann, M. Dommer, R. Pericet-Camara, D. Floreano, Active Connection Mechanism for Soft Modular Robots, *Advanced Robotics* 26 (7) (2012) 785–798.
- [5] M. Malley, M. Rubenstein, R. Nagpal, Flippy: A soft, autonomous climber with simple sensing and control, in: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 6533–6540, ISSN: 2153-0866.
- [6] A. Sproewitz, A. Billard, P. Dillenbourg, A. Ijspeert, Roombots-mechanical

- design of self-reconfiguring modular robots for adaptive furniture, in: IEEE
1060 International Conference on Robotics and Automation, 2009. ICRA '09,
2009, pp. 4259–4264.
- [7] A. Sproewitz, S. Pouya, S. Bonardi, J. V. D. Kieboom, R. Mockel, A. Bil-
lard, P. Dillenbourg, A. J. Ijspeert, Roombots: Reconfigurable Robots
for Adaptive Furniture, IEEE Computational Intelligence Magazine 5 (3)
1065 (2010) 20–32.
- [8] A. Sproewitz, Roombots: Design and Implementation of a Modular Robot
for Reconfiguration and Locomotion, Ph.D. thesis, EPFL (2010).
- [9] A. Sproewitz, P. Laprade, S. Bonardi, M. Mayer, R. Moeckel, P.-A. Mudry,
A. J. Ijspeert, Roombots - Towards decentralized reconfiguration with self-
1070 reconfiguring modular robotic metamodules, in: 2010 IEEE/RSJ Interna-
tional Conference on Intelligent Robots and Systems, IEEE, Taipei, 2010,
pp. 1126–1132.
- [10] A. Sproewitz, R. Moeckel, M. Vespignani, S. Bonardi, A. Ijspeert, Room-
bots: A hardware perspective on 3d self-reconfiguration and locomotion
1075 with a homogeneous modular robot, Robotics and Autonomous Systems
62 (7) (2014) 1016–1033.
- [11] S. Bonardi, M. Vespignani, R. Moeckel, A. J. Ijspeert, Collaborative manip-
ulation and transport of passive pieces using the self-reconfigurable mod-
ular robots roombots, in: Intelligent Robots and Systems (IROS), 2013
1080 IEEE/RSJ International Conference on, Ieee, 2013, pp. 2406–2412.
- [12] A. Sproewitz, R. Moeckel, J. Maye, A. J. Ijspeert, Learning to Move in
Modular Robots using Central Pattern Generators and Online Optimiza-
tion, The International Journal of Robotics Research 27 (3-4) (2008) 423–
443.
- 1085 [13] S. Bonardi, R. Moeckel, A. Sproewitz, M. Vespignani, A. J. Ijspeert, Lo-
comotion through Reconfiguration based on Motor Primitives for Room-

bots Self-Reconfigurable Modular Robots, in: Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on, VDE, 2012, pp. 1–6.

- 1090 [14] S. Bonardi, M. Vespignani, R. Moeckel, J. Van den Kieboom, S. Pouya, A. Sproewitz, A. Ijspeert, Automatic generation of reduced CPG control networks for locomotion of arbitrary modular robot structures, in: Robotics: Science and Systems X, Robotics: Science and Systems Foundation, 2014, pp. 1–9.
- 1095 [15] M. Vespignani, E. Senft, S. Bonardi, R. Moeckel, A. J. Ijspeert, An experimental study on the role of compliant elements on the locomotion of the self-reconfigurable modular robots Roombots, in: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, Ieee, 2013, pp. 4308–4313.
- 1100 [16] S. Bonardi, J. Blatter, J. Fink, R. Moeckel, P. Jermann, P. Dillenbourg, A. Ijspeert, Design and evaluation of a graphical iPad application for arranging adaptive furniture, in: 2012 IEEE RO-MAN, 2012, pp. 290–297.
- [17] A. Ozgur, S. Bonardi, M. Vespignani, R. Mockel, A. J. Ijspeert, Natural user interface for Roombots, in: Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on, IEEE, 2014, pp. 12–17.
- 1105 [18] E. Brown, N. Rodenberg, J. Amend, A. Mozeika, E. Steltz, M. R. Zakin, H. Lipson, H. M. Jaeger, Universal robotic gripper based on the jamming of granular material, Proceedings of the National Academy of Sciences 107 (44) (2010) 18809–18814.
- 1110 [19] M. Mutlu, S. Bonardi, M. Vespignani, S. Hauser, A. Bernardino, A. J. Ijspeert, Natural user interface for lighting control: Case study on desktop lighting using modular robots, in: 2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), 2016, pp. 288–293.

- 1115 [20] V. Nigolian, M. Mutlu, S. Hauser, A. Bernardino, A. Ijspeert, Self-reconfigurable modular robot interface using virtual reality: Arrangement of furniture made out of roombots modules, in: 2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), 2017, pp. 772–778.
- 1120 [21] H. Ahmadzadeh, E. Masehian, Modular robotic systems: Methods and algorithms for abstraction, planning, control, and synchronization, *Artificial Intelligence* 223 (2015) 27–64.
- [22] M. Yim, D. Duff, K. Roufas, PolyBot: a modular reconfigurable robot, in: IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA '00, Vol. 1, 2000, pp. 514–520 vol.1.
- 1125 [23] J. Davey, N. Kwok, M. Yim, Emulating self-reconfigurable robots - design of the SMORES system, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012, pp. 4464–4469.
- [24] X. Cui, J. Zhao, Y. Zhu, S. Tang, UBot: a new reconfigurable modular robotic system with multimode locomotion ability, *Industrial Robot: the international journal of robotics research and application* 39 (2) (2012) 178–190.
- 1130 [25] B. Salemi, M. Moll, W.-M. Shen, SUPERBOT: A deployable, multifunctional, and modular self-reconfigurable robotic system, in: Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, IEEE, 2006, pp. 3636–3641.
- 1135 [26] J. Liedke, R. Matthias, L. Winkler, H. Worn, The Collective Self-reconfigurable Modular Organism (CoSMO), in: 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2013, pp. 1–6.
- 1140 [27] J. W. Romanishin, K. Gilpin, S. Claiici, D. Rus, 3d M-Blocks: Self-Reconfiguring robots capable of locomotion via pivoting in three dimen-

- sions, in: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, IEEE, 2015, pp. 1925–1932.
- 1145 [28] J. Neubert, H. Lipson, Soldercubes: a self-soldering self-reconfiguring modular robot system, *Autonomous Robots*.
- [29] Y. Terada, S. Murata, Automatic Modular Assembly System and its Distributed Control, *The International Journal of Robotics Research* 27 (3-4) (2008) 445–462.
- 1150 [30] H. Kurokawa, K. Tomita, A. Kamimura, S. Kokaji, T. Hasuo, S. Murata, Distributed Self-Reconfiguration of M-TRAN III Modular Robotic System, *The International Journal of Robotics Research* 27 (3-4) (2008) 373–386.
- [31] D. Brandt, D. J. Christensen, H. H. Lund, ATRON Robots: Versatility from Self-Reconfigurable Modules, in: *2007 International Conference on Mechatronics and Automation*, 2007, pp. 26–32.
- 1155 [32] C. Liu, M. Whitzer, M. Yim, A distributed reconfiguration planning algorithm for modular robots, *IEEE Robotics and Automation Letters* 4 (4) (2019) 4231–4238.
- [33] T. Tosun, J. Daudelin, G. Jing, H. Kress-Gazit, M. Campbell, M. Yim, Perception-informed autonomous environment augmentation with modular robots, in: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6818–6824.
- 1160 [34] J. Daudelin, G. Jing, T. Tosun, M. Yim, H. Kress-Gazit, M. Campbell, An integrated system for perception-driven autonomy with modular robots, *Science Robotics* 3 (23).
- 1165 [35] D. Pianini, M. Viroli, J. Beal, Protelis: Practical aggregate programming, in: *Proceedings of the 30th Annual ACM Symposium on Applied Computing, SAC '15*, ACM, New York, NY, USA, 2015, pp. 1846–1853.

- [36] M. P. Ashley-Rollman, S. C. Goldstein, P. Lee, T. C. Mowry, Padmanabhan Pillai, Meld: A declarative approach to programming ensembles, in: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, pp. 2794–2800.
- [37] K. Lima, E. R. B. Marques, J. Pinto, J. B. Sousa, Dolphin: A task orchestration language for autonomous vehicle networks, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 603–610.
- [38] C. Pincioli, G. Beltrame, Buzz: An extensible programming language for heterogeneous swarm robotics, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016, pp. 3794–3800.
- [39] M. P. Weller, M. D. Gross, S. C. Goldstein, Hyperform Specification: Designing and Interacting with Self-reconfiguring Materials, *Personal Ubiquitous Comput.* 15 (2) (2011) 133–149.
- [40] M. Camplani, A. Paiement, M. Mirmehdi, D. Damen, S. Hannuna, T. Burghardt, L. Tao, Multiple human tracking in rgb-depth data: a survey, *IET Computer Vision* 11 (2017) 265–285(20).
- [41] J. Han, E. J. Pauwels, P. M. de Zeeuw, P. H. N. de With, Employing a rgb-d sensor for real-time tracking of humans across multiple re-entries in a smart environment, *IEEE Transactions on Consumer Electronics* 58 (2) (2012) 255–263.
- [42] J. Liu, Y. Liu, G. Zhang, P. Zhu, Y. Q. Chen, Detecting and tracking people in real time with rgb-d camera, *Pattern Recognition Letters* 53 (2015) 16–23.
- [43] A. Ramey, V. Gonzalez-Pacheco, M. A. Salichs, Integration of a low-cost rgb-d sensor in a social robot for gesture recognition, in: 2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI), 2011, pp. 229–230.

- [44] M. Ye, Q. Zhang, L. Wang, J. Zhu, R. Yang, J. Gall, A Survey on Human Motion Analysis from Depth Data, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, Ch. 2, pp. 149–187.
- 1200 [45] H. S. Koppula, R. Gupta, A. Saxena, Learning human activities and object affordances from rgb-d videos, *The International Journal of Robotics Research* 32 (8) (2013) 951–970.
- [46] A. Magnusson, M. Pacheco, M. Moghadam, H. H. Lund, D. J. Christensen, Fable: Socially Interactive Modular Robot, *The Eighteenth International Symposium on Artificial Life and Robotics* (2013) 9.
- 1205 [47] J. Forlizzi, C. DiSalvo, Service robots in the domestic environment: A study of the roomba vacuum in the home, in: *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-robot Interaction, HRI '06*, ACM, New York, NY, USA, 2006, pp. 258–265.
- 1210 [48] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, P. Abbeel, Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding, in: *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 2308–2315.
- [49] J. van den Berg, S. Miller, K. Goldberg, P. Abbeel, Gravity-Based Robotic Cloth Folding, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, Ch. 6, pp. 409–424.
- 1215 [50] J. Broekens, M. Heerink, H. Rosendal, Assistive social robots in elderly care: a review, *Gerontechnology* 8 (2).
- [51] S. Martin, G. Kelly, W. G. Kernohan, B. McCreight, C. Nugent, Smart home technologies for health and social care support, *Cochrane Database of Systematic Reviews* 2008 4 (2) (2009) 13.
- 1220 [52] Y. Demiris, Knowing when to assist: Developmental issues in lifelong assistive robotics, in: *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2009, pp. 3357–3360.

- 1225 [53] M. Beetz, D. Jain, L. Mosenlechner, M. Tenorth, L. Kunze, N. Blodow,
D. Pangercic, Cognition-Enabled Autonomous Robot Control for the Real-
ization of Home Chore Task Intelligence, *Proceedings of the IEEE* 100 (8)
(2012) 2454–2471.
- [54] S. S. Srinivasa, D. Berenson, M. Cakmak, A. Collet, M. R. Dogar,
1230 A. D. Dragan, R. A. Knepper, T. Niemueller, K. Strabala, M. V. Weghe,
J. Ziegler, Herb 2.0: Lessons Learned From Developing a Mobile Manipu-
lator for the Home, *Proceedings of the IEEE* 100 (8) (2012) 2410–2428.
- [55] T. Asfour, N. Vahrenkamp, D. Schiebener, M. Do, M. Przybylski, K. Welke,
J. Schill, R. Dillmann, ARMAR-III: Advances in Humanoid Grasping and
1235 Manipulation, *Journal of the Robotics Society of Japan* 31 (4) (2013) 341–
346.
- [56] T. Carlson, Y. Demiris, Collaborative Control for a Robotic Wheelchair:
Evaluation of Performance, Attention, and Workload, *IEEE Transactions*
on Systems, Man, and Cybernetics, Part B (Cybernetics) 42 (3) (2012)
1240 876–888.
- [57] Y. Sugahara, T. Hosobata, Y. Mikuriya, H. Sunazuka, H.-o. Lim, A. Takan-
ishi, Realization of dynamic human-carrying walking by a biped locomo-
tor, in: *IEEE International Conference on Robotics and Automation, 2004.*
Proceedings. ICRA '04. 2004, Vol. 3, 2004, pp. 3055–3060 Vol.3.
- 1245 [58] D. Sirkin, B. Mok, S. Yang, W. Ju, Mechanical ottoman: How robotic fur-
niture offers and withdraws support, in: *Proceedings of the Tenth Annual*
ACM/IEEE International Conference on Human-Robot Interaction, HRI
'15, ACM, New York, NY, USA, 2015, pp. 11–18.
- [59] OriLiving, Robotic wardrobes (2019).
1250 URL <https://oriliving.com>
- [60] C. Georgoulas, T. Linner, T. Bock, Towards a vision controlled robotic
home environment, *Automation in Construction* 39 (2014) 106 – 116.

- [61] M. D. Gross, K. E. Green, Architectural robotics, inevitably, interactions 19 (1) (2012) 28–33.
- 1255 [62] C.-H. Yu, F. Willems, D. Ingber, R. Nagpal, Self-organization of environmentally-adaptive shapes on a modular robot, in: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, pp. 2353–2360.
- [63] M. Vespignani, Challenges in the Locomotion of Self-Reconfigurable Modular Robots, Ph.D. thesis, EPFL (2015).
- 1260 [64] H. Wei, Y. Chen, J. Tan, T. Wang, Sambot: A self-assembly modular robot system, IEEE/ASME Transactions on Mechatronics 16 (4) (2011) 745–757.
- [65] Y. Zhu, H. Jin, X. Zhang, J. Yin, P. Liu, J. Zhao, A multi-sensory autonomous docking approach for a self-reconfigurable robot without mechanical guidance, International Journal of Advanced Robotic Systems 11 (9) 1265 (2014) 146.
- [66] F. Mondada, L. M. Gambardella, D. Floreano, S. Nolfi, J. Deneuborg, M. Dorigo, The cooperation of swarm-bots: physical interactions in collective robotics, IEEE Robotics Automation Magazine 12 (2005) 21–28.
- 1270 [67] J. Amend, E. Brown, N. Rodenberg, H. Jaeger, H. Lipson, A Positive Pressure Universal Gripper Based on the Jamming of Granular Material, IEEE Transactions on Robotics 28 (2) (2012) 341–350.
- [68] J. Amend, N. Cheng, S. Fakhouri, B. Culley, Soft Robotics Commercialization: Jamming Grippers from Research to Product, Soft Robotics 3 (4) 1275 (2016) 213–222.
- [69] H. Khodr, M. Mutlu, S. Hauser, A. Bernardino, A. Ijspeert, An Optimal Planning Framework to Deploy Self-Reconfigurable Modular Robots, IEEE Robotics and Automation Letters 4 (4) (2019) 4278–4285.

- 1280 [70] M. Mutlu, S. Hauser, A. Bernardino, A. Ijspeert, Playdough to Roombots: Towards a Novel Tangible User Interface for Self-reconfigurable Modular Robots, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 1970–1977.
- [71] A. J. Ijspeert, Central pattern generators for locomotion control in animals and robots: A review, *Neural Networks* 21 (4) (2008) 642–653.
- 1285 [72] D. Krupke, G. Li, J. Zhang, H. Zhang, H. P. Hildre, Flexible modular robotic simulation environment for research and education., in: European Conference on Modelling and Simulation (ECMS), 2012, pp. 243–249.
- [73] H. Kurokawa, K. Tomita, E. Yoshida, S. Murata, S. Kokaji, Motion simulation of a modular robotic system, in: IEEE 26th International Conference on Industrial Electronics, Control and Instrumentation., Vol. 4, 2000, pp. 2473–2478 vol.4.
- 1290 [74] G. Jing, T. Tosun, M. Yim, H. Kress-Gazit, An end-to-end system for accomplishing tasks with modular robots., in: *Robotics: Science and Systems*, 2016, pp. 1–5.
- 1295 [75] R. Gross, E. Tuci, M. Dorigo, M. Bonani, F. Mondada, Object transport by modular robots that self-assemble, in: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 2558–2564.
- 1300 [76] J. Nguyen-Duc, M. Mutlu, S. Hauser, A. Bernardino, A. J. Ijspeert, Cooperative bridge building by self-reconfigurable modular robots based on ants’ stigmergic behaviour, in: *9th International Symposium on Adaptive Motion of Animals and Machines (AMAM 2019)*, 2019.