Challenges in the Locomotion of Self-Reconfigurable Modular Robots

THÈSE N^{O} 6791 (2015)

PRÉSENTÉE LE 11 DÉCEMBRE 2015
À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE DE BIOROBOTIQUE
PROGRAMME DOCTORAL EN SYSTÈMES DE PRODUCTION ET ROBOTIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Massimo VESPIGNANI

acceptée sur proposition du jury:

Prof. P. Dillenbourg, président du jury
Prof. A. Ijspeert, Prof. R. Pfeifer, directeurs de thèse
Prof. F. Iida, rapporteur
Prof. A. Ishiguro, rapporteur
Prof. J. Paik, rapporteuse

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2015

Abstract

Self-Reconfigurable Modular Robots (SRMRs) are assemblies of autonomous robotic units, referred to as modules, joined together using active connection mechanisms. By changing the connectivity of these modules, SRMRs are able to deliberately change their own shape in order to adapt to new environmental circumstances. One of the main motivations for the development of SRMRs is that conventional robots are limited in their capabilities by their morphology. The promise of the field of self-reconfigurable modular robotics is to design robots that are robust, self-healing, versatile, multi-purpose, and inexpensive.

Despite significant efforts by numerous research groups worldwide, the potential advantages of SRMRs have yet to be realized. A high number of degrees of freedom and connectors make SRMRs more versatile, but also more complex both in terms of mechanical design and control algorithms. Scalability issues affect these robots in terms of hardware, low-level control, and high-level planning. In this thesis we identify and target three major challenges: (i) Hardware design; (ii) Planning and control; and, (iii) Application challenges.

To tackle the hardware challenges we redesigned and manufactured the Self-Reconfigurable Modular Robot Roombots to meet desired requirements and characteristics. We explored in detail and improved two major mechanical components of an SRMR: the actuation and the connection mechanisms. We also analyzed the use of compliant extensions to increase locomotion performance in terms of locomotion speed and power consumption.

We contributed to the control challenge by developing new methods that allow an arbitrary SRMR structure to learn to locomote in an efficient way. We defined a novel bio-inspired locomotion-learning framework that allows the quick and reliable optimization of new gaits after a morphological change due to self-reconfiguration or human construction.

In order to find new suitable application scenarios for SRMRs we envision the use of Roombots modules to create Self-Reconfigurable Robotic Furniture. As a first step towards this vision, we explored the use and control of Plug-n-Play Robotic Elements that can augment existing pieces of furniture and create new functionalities in a household to improve quality of life.

Key words: Self-reconfigurable modular robots, mechanical design, locomotion, self-reconfiguration, user-interfaces

Résumé

Les Robots Modulaires Auto-Reconfigurables (RMARs) sont constitués d'unités robotiques autonomes appelés modules reliés entre eux par des mécanismes de connexion. En changeant la connectivité de ces modules, les RMARs sont capables de changer volontairement leur propre forme afin de l'adapter aux nouvelles conditions environnementales. L'une des principales raisons de l'utilisation de ces robots reconfigurables est que les robots conventionnels sont limités dans leurs capacités par leur morphologie. La promesse du domaine de la robotique modulaire auto-reconfigurable est la conception de robots robustes, polyvalents, multi-usages, et peu coûteux.

Malgré les efforts importants de nombreux groupes de recherche, les avantages potentiels des Robots Modulaires Auto-Reconfigurables doivent encore être concrétisés. Le nombre élevé de degrés de liberté et de connecteurs rend les RMARs plus polyvalents, mais également plus complexes à la fois en matière de conception mécanique mais aussi du point de vue des algorithmes de contrôle. Les contraintes liées à cette capacité d'évolutivité affectent ces robots en matière de matériel, de contrôle bas niveau et de planification haut niveau. Dans cette thèse, nous identifions et abordons trois défis majeurs : (i) la conception mécanique ; (ii) la planification et le contrôle ; et, (iii) les défis de l'application.

Pour relever les défis de la conception mécanique nous avons repensé et fabriqué les Robots Modulaires Auto-Reconfigurables Roombots, développés à l'EPFL, pour répondre aux exigences et caractéristiques souhaitées. Nous avons exploré en détail et amélioré deux principaux composants mécaniques d'un RMAR : le système d'actionnement et les mécanismes de connexion. Nous avons également analysé l'utilisation d'extensions élastiques pour augmenter les performances de la locomotion en matière de vitesse de locomotion et de consommation d'énergie.

Nous avons contribué à relever le défi du contrôle de tels robots en développant de nouvelles méthodes qui permettent à une structure de RMARs à la morphologie arbitraire d'apprendre à se déplacer d'une manière efficace. Nous avons défini une nouvelle stratégie d'apprentissage bio-inspirée de la locomotion qui permet d'optimiser rapidement et de manière fiable de nouvelles démarches après un changement morphologique résultant d'un processus d'auto-reconfiguration ou d'assemblage manuel.

Afin de trouver de nouveaux scénarios d'usage appropriés pour les RMARs, nous avons envisagé l'utilisation des modules Roombots pour créer des meubles robotiques adaptatifs capables de se mouvoir librement et de s'adapter à des situations et des environnements de la vie quotidienne. Comme premier pas vers cette vision à long terme, nous avons exploré l'utilisation et le contrôle d'éléments robotiques « Plug-n-Play » qui peuvent augmenter des meubles existants et créer de nouvelles fonctionnalités dans un ménage pour améliorer la qualité de vie.

Mots clef : Robots modulaires auto-reconfigurables, conception mécanique, locomotion, auto-reconfiguration, interfaces utilisateur

Acknowledgements

In full gratitude I would like to acknowledge the following individuals who supported, encouraged, inspired, and sacrificed themselves to help my pursuit of a doctoral degree.

First and foremost I would like to thank my advisor, Prof. Auke Jan Ijspeert, for giving me the fantastic opportunity to work in his laboratory under his guidance. I appreciate all of his contributions of time, ideas, and funding to make my Ph.D. experience productive, stimulating, and fit to my research interests. Not only he is a great and caring supervisor, but also an amazing person with the unique capability of thinking positively in any situation. I would like to also thank my PhD co-supervisor, Prof. Rolf Pfeifer, for inspiring me to explore the fields of morphological computation and embodied intelligence and for all his insightful comments and suggestions. I want to thank the members of my thesis committee, Prof. Fumiya Iida, Prof. Akio Ishiguro, Prof. Jamie Paik, and Prof. Pierre Dillenbourg, for their time and their valuable suggestions and comments on this thesis.

I am thankful to Sylvie Fiaux for warmly welcoming me into the lab, for her kindness, and for her administrative support at all times. I would also like to thank Claire Chabanel and Corinne Nathalie Lebet for assisting me with EDPR reporting and exam registration. I wish to acknowledge the personnel of the EPFL mechanical workshops and in particular André Badertscher, Jean-Paul Brugger, André Guignard, Mitch Heynick, and Alfred Thomas for help and support during hardware development. I am thankful to François Longchamp for patiently supporting me with his technical expertise and for his invaluable help during the manufacture and assembly of the robots. I would like to thank EPFL, the Swiss National Science Foundation (SNSF), and the National Centre of Competence in Research (NCCR) Robotics for supporting my work.

I am honestly thankful to my late electronics professor Dario Malosti for deeply influencing my professional career and personal life, and for all the passion and dedication that he put into his work.

I would like to thank all the past and current BIOROB members who have positively influenced my Ph.D. through collaborations, useful discussions, or for just being there when I needed them the most. I would like to especially thank:

Alexander for teaching me so much about mechanical design and manufacturing, for being a constant source of inspiration, and for getting so many songs stuck in my head.

Luca for all the coffee breaks and nights out, for convincing me to start rock climbing even though I am afraid of heights, and for being so awesome! Just kidding.

Kamilo for motivating me into exploring a different research field, which lead to amazing results, and for showing me videos of his past career as an air bass-guitar player.

Cole for helping me improve this thesis, for introducing me to ultimate frisbee, and for agreeing on running the 20 km of Lausanne together in funny costumes. Perhaps also for setting up silly challenges‽ #ChallengeAccepted

Andrej for all the conversations, the morning coffee, the second morning coffee, the early afternoon coffee, and the afternoon coffee, and for all the dinners out at Chez Xu.

Steve for teaching me some delicious Chinese recipes and for all the Skype calls from across the globe.

Stéphane for being a great office mate, for all the inside jokes, and for all the nights spent in the lab trying to meet deadlines.

Alessandro for always having a secret tool or a hidden webpage ready to satisfy my requests.

Jesse for helping me out multiple times setting up his optimization framework and Cȯdɣn and for keeping up with me on the yearly BIOROB raclette eating challenge.

Mehmet and Simon for all their support during these past months, for helping me finish up experiments for this thesis, and for creating a good office atmosphere.

Andrej G. and Tadej for the fun time spent together at EPFL, during outings, and at conferences.

I would like to thank Elisa for supporting and putting up with me during the tough months of thesis preparation and writing, for cheering me up, and for all the amazing experiences we are sharing together.

I wish to also thank Dora for always encouraging me with her positive attitude.

Last and most importantly I want to express my deep gratitude to my best friend and brother, Marco, and to my parents and family for their love, support, trust, and guidance throughout my life.

Massimo Vespignani
Lausanne, October 2015

Contents

Abstract i

Acknowledgements v

Table of contents vii

I Introduction 1

1 Thesis Outline 3
1.1 Motivation and Challenges . . . 3
1.2 Goals . . . 5
1.3 Outline . . . 5
1.4 Contributions . . . 7

2 Self-Reconfigurable Modular Robots 9
2.1 History and Classification . . . 11
2.2 Roombots: a Self-Reconfigurable Modular Robot for a Living Environment . . 15
2.3 Lola-OP™ Reconfigurable Modular Snake Robot . . . 16

II Mechanical Challenges 17

3 Overview of the Design of Roombots 19
3.1 Roombots Module Design . . . 21
3.2 Roombots DoF and Housing . . . 23
3.3 Motor and Gearbox . . . 24
3.4 Active Connection Mechanism . . . 27
3.5 Electronics . . . 28
3.6 Proposed Mechanical Improvements . . . 29
3.7 Conclusions . . . 33

4 Actuation System 35
4.1 Introduction . . . 36
4.2 Roombots actuation system . . . 36

4.2.1 Gearbox A: Original Design . . . 37
4.2.2 Gearbox B: Compact Design . . . 39
4.2.3 Gearbox C: Double Support Design . . . 41
4.2.4 Gearbox D: Metal Design . . . 41
4.3 Bench Test and Evaluation . . . 44
4.4 Conclusions . . . 47

5 Connection Mechanism 49
5.1 Introduction . . . 50
5.2 Original ACMv3 Design . . . 52
5.3 ACMv4 . . . 53
5.4 Hybrid ACM . . . 55
5.5 Conclusions . . . 56

6 External Attachments 59
6.1 Passive parts . . . 61
6.2 Specialized Parts . . . 61
6.3 Compliant Parts . . . 63
6.3.1 Compliant parts for Roombots modules . . . 66
6.3.2 Adding Compliance to the Lola-OP™ Snake Robot . . . 69
6.4 Conclusions . . . 71

III Locomotion Challenges 73

7 Overview of the Roombots locomotion control 75
7.1 Locomotion Controllers for Self-Reconfigurable Modular Robots . . . 76
7.1.1 Related Work . . . 76
7.2 Central Pattern Generators (CPGs) . . . 78
7.3 Conclusions . . . 80

8 Body/Limb and Symmetry Finder Algorithms 81
8.1 Introduction . . . 82
8.1.1 Related Work . . . 83
8.2 Materials and Methods . . . 84
8.2.1 Body/Limb Finder . . . 84
8.2.2 BLF Rulebook . . . 85
8.2.3 Distance-based Symmetry . . . 88
8.3 Experimental Results . . . 88
8.4 Discussion . . . 90
8.5 Conclusions . . . 94

9 Hybrid and Meta-Optimization 97
9.1 Introduction . . . 99

9.1.1 Related Work 99
9.2 Hybrid Optimization 100
9.2.1 Offline Optimization 101
9.2.2 Hardware Experiments 101
9.2.3 Meta-Optimization 102
9.3 Experimental Results and Discussion 104
9.4 Conclusions 106

10 Locomotion with Compliant Elements 109
10.1 Roombots Locomotion with In-Series Compliant Elements 111
10.1.1 Related Work 112
10.1.2 Materials and Methods 112
10.1.3 Experimental Results and Discussion 115
10.1.4 Conclusions 117
10.2 Lola-OP™ Locomotion with in-series Compliant Elements 118
10.2.1 Related Work 119
10.2.2 Materials and Methods 119
10.2.3 Experimental Results and Discussion 125
10.2.4 Discussion 129
10.3 Lola-OP™ Locomotion on Horizontal Pipes 132
10.3.1 Related Work 133
10.3.2 Materials and Methods 134
10.3.3 Experimental Results 139
10.3.4 Discussion 143
10.4 Conclusions 145

IV Applications 147

11 Towards Self-Reconfigurable Robotic Furniture 149
11.1 Introduction 150
11.2 Examples of Plug-n-Play Robotic Elements 151
11.3 Conclusions 154

12 Gesture-Based User Interfaces for Self-Reconfigurable Modular Robots 155
12.1 Introduction 156
12.2 Vision-based lighting control 156
12.3 Natural Roombots User Interface 158
12.4 Conclusions 160

13 Conclusions and Outlook 161

Appendices **167**
A List of modular robots 169
B On-Grid Locomotion through Self-Reconfiguration 173
C Roombots Quadruped Locomotion 177
D Self-Reconfiguration from Tripedal to Snake Morphology 179
E Particle Swarm Optimization (PSO) 181

List of figures **183**

List of tables **191**

Bibliography **211**

Curriculum Vitae **213**

Introduction Part I

1 Thesis Outline

1.1 Motivation and Challenges

People have long imagined creatures or machines with shapeshifting abilities. While in old mythological stories human beings were physically transformed into animals or monstrous creatures, for example using magic spells or divine intervention as a form of punishment, more recent fictional stories envisioned machines that can morph at will to gain new abilities, self-heal, or adapt to new conditions. A great example from science fiction is the *T-1000 robot* from the movie Terminator 2: Judgment Day, which is made from futuristic liquid-like metal and can change its shape, copy forms, and self-heal when dismembered. *Transformers* can assemble and combine their bodies into a single machine that is larger and more powerful than each of them individually. In the recent movie Big Hero 6, the main character creates *Microbots:* tiny robots that can connect together to form various shapes and perform tasks cooperatively while controlled with a single wearable neurotransmitter.

Since the 1980s, scientists have aimed to develop shapeshifting machines in the form of Self-Reconfigurable Modular Robots (SRMRs). Beyond standard properties common to fixed-morphology robots, SRMRs are able to deliberately change their own shape by rearranging the connectivity of their modules in order to adapt to new environmental circumstances [249]. One of the main motivations for this is that conventional robots are limited in their capabilities by their morphology: a robotic arm has a predefined limited reachable workspace; a search and rescue quadruped robot has a fixed width and height and cannot crawl through tight spaces in the rubble of a collapsed building. With a self-reconfigurable system, a modular legged robot could change to a snake robot to go through small holes. The ability to reconfigure allows a robot to disassemble and reassemble to form new morphologies that are better suited for a new task that was not known in advance.

While this practically-oriented motivation is sufficient for developing and working with Self-Reconfigurable Modular Robots, their advantages and potential capabilities are much deeper [208]. The long term vision is to design robots that are robust, self-healing, versatile, multi-purpose, and inexpensive [250]. If a module fails, other units can keep working toward the achievement of the goal. The defective module can be ejected from the structure and autonomously replaced with a functioning one. Alternatively, the remaining modules can re-arrange their connectivity to form a more suitable configuration. Potentially, Self-Reconfigurable systems can also lower the overall robot cost due to production at scale of many copies of the same modules. However, despite these premises, the potential advantages of Self-Reconfigurable Modular Robots have yet to be realized [209].

Unfortunately, unlike Microbots' fictional unlimited bonding and actuation strength, invisible power source, perfect synchronization and coordination, and simple and effective end-user interface, real systems have to face a number of hardware, planning, and control challenges and limitations. The high number of degrees of freedom (DoF) and connectors make SRMRs more versatile, but also more complex both in terms of mechanical design and control algorithms. Scalability issues affect these robots in terms of hardware, low-level control, and high-level planning. The promised application scenarios envision hundreds or thousands of modules working together; however, to date, the modular robot with most active modules, Polybot, had only 56 units. There are **hardware design challenges** with fundamental limiting factors that prevent large numbers of modules [249], including:

- Limits on power density and efficiency of the actuators, torque and speed output, motion precision and accuracy
- Limits on bonding strength, precision, robustness, and area of acceptance of the connection mechanisms
- Limits on the features and capabilities that can be embedded in each individual module

Mechanical challenges include designing modules that can lift several times their own weight, connectors that can self-align and guarantee a strong bond under load, and structural parts that will not collapse when many modules join together. Design, materials, and assembly techniques should guarantee a scalable manufacturing process to allow for a large number of modules to be mass produced.

Once a Self-Reconfigurable Structure is built, we face **planning and control challenges** in order to efficiently coordinate its large number of degrees of freedom, such as:

- Finding optimal algorithms to plan the self-reconfiguration process
- Handling different failure modes, from malfunctioning modules to misalignment problems
- Defining algorithms that decide when it is necessary to self-reconfigure, and what is the most suitable form for the task and environment
- Creating a fast, efficient, and scalable communication method between modules

- Coordinating the modules' movement to produce effective locomotion

In this context we are particularly interested in this last point, i.e. defining a locomotion-learning framework that would allow a Self-Reconfigurable Modular Robots assembled in any arbitrary morphology to locomote in an efficient way. If a modular robot employed in a time-critical mission undergoes a reconfiguration process, it should be able to find new locomotion-control parameters that would allow it to continue its mission in the shortest time possible.

Moreover, despite the promised versatility, robustness, and reduced cost, there is still the challenge of finding the optimal uses for SRMRs (**application challenge**). Given a known task and environment, a dedicated conventional robot will be simpler and better performing than a comparable reconfigurable robot. As a consequence, Self-Reconfigurable Modular Robots become advantageous only when the task or environment cannot be known in advance, or in situations in which multiple types of robots are needed to complete a task. We yet have to find the "killer application" for these types of robots. Optimal scenarios are space exploration missions, disaster areas, and deep-sea underwater structures, although SRMRs can also prove to be useful in more practical applications such as robot construction kits, educational robotics, and reconfigurable furniture for assisted living.

1.2 Goals

The goal of this thesis is to contribute in tackling some of the current challenges in the field of Self-Reconfigurable Modular Robotics in four different ways. First, we wish to redesign and manufacture a Self-Reconfigurable Modular Robot (Roombots) that meets desired requirements and characteristics. Second, we want to give an arbitrary structure built with our Self-Reconfigurable Modular Robot the ability to locomote in an efficient way. For this purpose, we want to define a locomotion-learning framework which allows for a fast control-parameter optimization to quickly learn new gaits after a morphological change due to self-reconfiguration or human construction. Third, we wish to analyze the effect and effectiveness of external attachments, in particular of passive compliant elements that could be used to improve locomotion capabilities. Fourth, we want to explore new application scenarios for Self-Reconfigurable Modular Robots testing their usefulness as building blocks for Self-Reconfigurable Robotic Furniture and exploring new natural ways to control them.

1.3 Outline

This thesis is divided into four major parts:

- **Part I** contains the **problem statement** with a description of the goals and motivation behind this work.

A review of the state of the art on (Self-) Reconfigurable Modular Robots is presented in Chapter 2, along with an introduction to the Roombots project, which is key to this work. We will conclude this part presenting the Lola-OP™ Modular Snake Robot, which is an easy-to-use Reconfigurable Modular Robot that we adopted for some of our studies on the use of compliant elements for locomotion.

- **Part II** concerns the challenges of a homogeneous modular **mechanical design**, in which every component must be thought and designed to work in the worst possible scenario while trying to keep it compact and lightweight.

  In Chapter 3 we give an overview of the different components contained in a Roombots module.

  We then analyze in detail the mechanical design of two of the most critical parts of a Self-Reconfigurable Modular Robot, namely the gearbox-actuator system, in Chapter 4, and the active connection mechanism, in Chapter 5.

  We conclude this part with Chapter 6, proposing the design and use of external attachments in the form of passive parts, specialized parts, and compliant parts, to enhance the construction of modular structures, provide them with new specialized features, and improving their locomotion capabilities.

- **Part III** is about **locomotion challenges**, learning how to coordinate the large number of degrees of freedom in a modular structure after self-reconfiguration, and how to improve locomotion performance through the use of compliance.

  In Chapter 7 we give an overview of the locomotion and optimization framework that we used for most of the studies herein.

  In Chapter 8 we describe one bio-inspired approach to reduce time needed to optimize locomotion parameters for any Self-Reconfigurable Modular structure. This is useful for quickly learning new gaits after self-reconfiguration.

  In Chapter 9 we present a hybrid optimization algorithm that allows to produce results from offline optimization that are at the same time well matched with the hardware robot. This allows to transfer learned control parameters from simulation to hardware with little reality gap.

  We conclude this part with Chapter 10, proposing the use of compliance to increase the locomotion performance of a (Self-) Reconfigurable Modular Robot, for instance in terms of locomotion speed and power consumption.

- **Part IV** describes possible **applications** for our Self-Reconfigurable Modular Robot.

  In Chapter 11 we envision Self-Reconfigurable Robotic Furniture that can be used for assisted living scenarios and present preliminary experiments using Roombots modules to build Plug-n-Play Robotic Elements that can be used in a household to improve the quality of life.

> In Chapter 12 we describe two intuitive interfaces that allow non-expert users to control Roombots modules and Plug-n-Play Robotic Elements using natural gestures.

We wrap-up the thesis with conclusions and outlook in Chapter 13.

Lastly, in the Appendix we provide additional material and describe some experiments that are not the main focus of this thesis.

1.4 Contributions

The original contributions, which are described in more detail in Chapter 13, of this thesis are:

1. Global analysis of different challenges in modular robotics
2. Improvement of the mechanical design of an existing Self-Reconfigurable Modular Robot, Roombots, to meet application requirements and allow a more scalable production
3. A novel bio-inspired parameter reduction technique that allows quick learning of new locomotion control parameters
4. A new hybrid optimization method to enhance offline optimization by better matching simulation results with hardware “replayability”
5. Investigation of the use of in-series passive compliant elements in modular robots to improve their speed of locomotion and energy efficiency
6. Demonstrations of application of Self-Reconfigurable Modular Robots for creating augmented furniture and new features for a domestic environment
7. Intuitive user interfaces to control Self-Reconfigurable Modular Robots using natural gestures.

2 Self-Reconfigurable Modular Robots

The field of Self-Reconfigurable Modular Robotics addresses the design, fabrication, motion planning, and control of autonomous robots able to deliberately change their own shape by rearranging the connectivity of their parts in order to adapt to new circumstances, perform new tasks, or recover from damage [249]. The goal may be a robot built for a specific task (a snake to pass through a tunnel that then changes into a quadruped structure to quickly cover open uneven ground) or an object/structure designed for a particular job (such as a wrench, a bridge, or a set of chairs and a table). When the task is complete, the modules in the structure can disconnect and be reused for a different task or to create a different object [64].

In Section 2.1 we present some of the most influential (Self-) Reconfigurable Modular Robots while presenting some of the classification methods used to characterize the different systems. The field started in the 1980s with the cell structured robot CEBOT and today there are more than a hundred different systems.

We briefly present our SRMR, Roombots, in Section 2.2. The Roombots project aims at creating self-assembling furniture-like structures that have morphing and locomotion capabilities. One Roombots module is a fully autonomous robot with 3 active degrees of freedom and active or passive ports that are used to connect to other modules or passive connectors in the environment. A detailed description of the mechanics is provided in Part II.

We conclude the chapter in Section 2.3 with a description of the Lola-OP™ Modular Snake Robot, which is a Reconfigurable Modular Robot (RMR) that we used for some studies presented herein. Each module of Lola-OP™ has one oscillating degree of freedom assembled using the Bioloid Educational Robot Kit [12]. In our work we used a chain configuration with 8 degrees of freedom.

Some parts of this chapter have been adapted from:

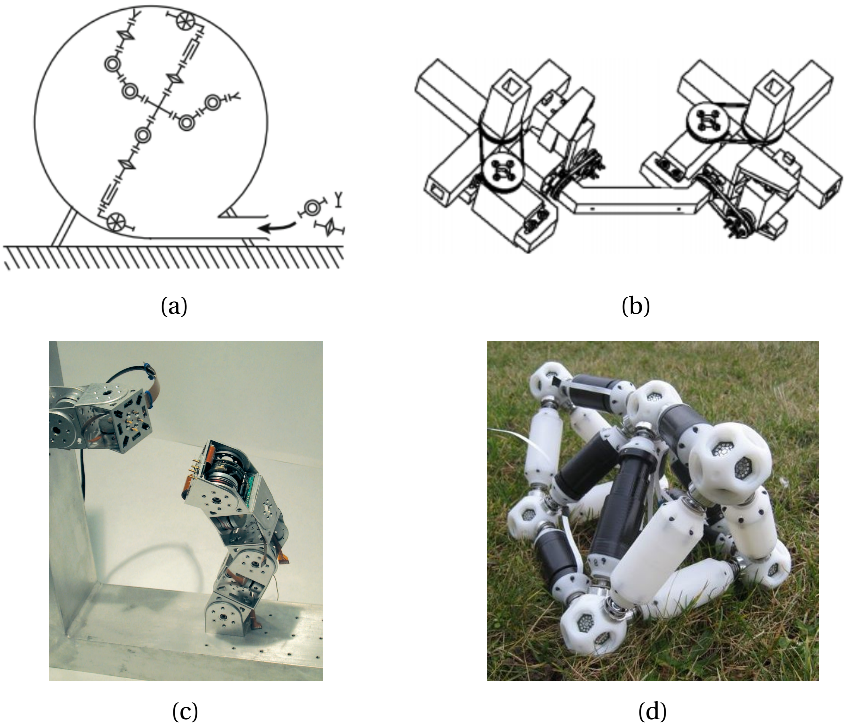
[204] A. Spröwitz, R. Moeckel, M. Vespignani, S. Bonardi, and A.J. Ijspeert. “Roombots: A Hardware Perspective on 3D Self-Reconfiguration and Locomotion with a Homogeneous Modular Robot”. In *Robotics and Autonomous Systems*, Volume 62, Issue 7, July 2014, Pages 1016-1033, 2014.

My original contribution: Conceptual contribution, Experimental setup, Hardware experiments, Partial writing.

2.1 History and Classification

The field of self-reconfigurable modular robotics started with a paper presented by Toshio Fukuda et al. in 1988 [57] in which he introduced the topic of modular robots as “Dynamically Reconfigurable Robotic Systems (DRRS)”. Their robot, CEBOT (CEllular roBOTic system), is the first self-reconfiguring modular robot and was designed to provide a more “optimally configured” robot compared to other one-purpose industrial robots.

CEBOT can be classified as a *heterogeneous* modular robot since it features different “cells” specialized in different tasks (i.e. actuating cells, branching cell, and working cells, shown in Figure 2.1.a). The specialization via a heterogeneous design allows a system to be composed of simpler modules, as not all the functionality has to be integrated into one single module. An example is the Polypod bi-unit modular robot [246] built up of two types of modules: “segments”, containing a 2-DoF parallel mechanism, and “nodes”, whose main purpose is to hold the batteries. This presents a big advantage, both on the design and the debugging level. However heterogeneity can often limit applications as modules cannot be easily replaced with any other (e.g. in case of specialized modules, in case of modules with male-female heterogeneous connectors). The reconfiguration planning of *homogeneous* systems is easier as it requires only to take care of one module type. One of the earliest examples of homogeneous modular robot is Fracta. First published in 1994 by Murata et al. [135], Fracta is made from


(a) (b)

(c) (d)

Figure 2.1 – Examples of Modular Robots: (a) CEBOT. (b) Molecule. (c) PolyBot G3. (d) Odin. Pictures from [57, 99, 249, 111].

multiple, 2-dimensional (planar) Fractum units. Each Fractum is made from layered electromagnetic coils, with which a Fractum can connect to and rotate around neighboring Fractum units.

From a hardware architecture point of view, Fracta can be classified as a *lattice-type* Modular Robot, as its units are arranged and connected in a regular pattern and can change shape by moving into discrete positions on a virtual grid, or lattice [245]. While Fracta and Metamorphic [33] are planar lattice robots, the first 3-dimensional lattice-type robots were the Molecule [99] (Figure 2.1.b) and the 3-d units [100]. Such design makes reconfiguration simple but locomotion more difficult. Lattice robots generate locomotion through a “flow” of modules on top of each other, one by one, which makes the locomotion complex and slow.

Using the hardware architecture as a classifier, another class are the *chain/tree-type* Reconfigurable Robots, which have units that are connected together in a string or tree topology and can move and reach any point in space. In the simplest configuration, they have two connectors (e.g. on both sides of their module’s shell), hence modules can only be connected into chains. Strictly speaking, this limits possible robot configurations to variations of snake-like robots, track- or wheel-like robots, and spiral-shaped robots. Modules with more than two connectors can be employed as “branches”. Among others (e.g. CONRO [237, 31], CK-BOT [183]), this is implemented in the PolyBot modular robot [244] (Figure 2.1.c). A four connector node (in addition to the intrinsic two connectors per module) allows PolyBot to form quadruped configurations.

Mobile Reconfigurable Robots have an elevated degree of self-mobility. They are equipped with wheels/tracks or with many degrees of freedom that allows them to move independently in the environment. Once attached to each other, mobile robots can form either lattice or chain structures. Swarm-Bot modules [132] use tracks for driving and large grippers to grab onto neighboring Swarm-Bot units. Once they are connected in a chain structure, they can overcome large obstacles or gaps.

Truss-type robots use struts that can stretch and contract. Usually this type of robot are composed of active units that can vary their length, passive struts with fixed length, and joint components. Examples of truss-type robots are Odin [111] (shown in Figure 2.1.d), Tetrobot [74], and Morpho [257].

Some modular robotic systems that clearly defy characterization as either a chain or a lattice have been grouped in *free-form-type* robots [64]. They don’t need to form a regular lattice and have the ability to assemble modules in semi-arbitrary positions. Two instances of free-form Modular Robots are Slimebot [196] and Catoms [95].

With the emergence of more modular robots the clean distinction between lattice and chain type robots started to fade, e.g. the SuperBot module is described as a “hybrid chain and lattice architecture” [180]. *Hybrid-type* modules connect to each other and move like chain structures, but can also align to the grid and assume lattice structures. The first *hybrid-type* Modular

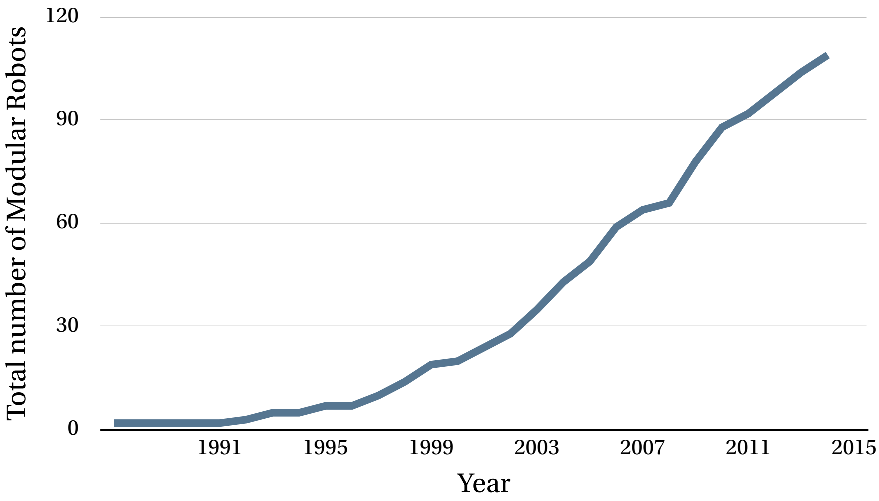


Figure 2.2 – Overall trend of total number of Modular Robots. Data from Table A.1, reported in Appendix A.

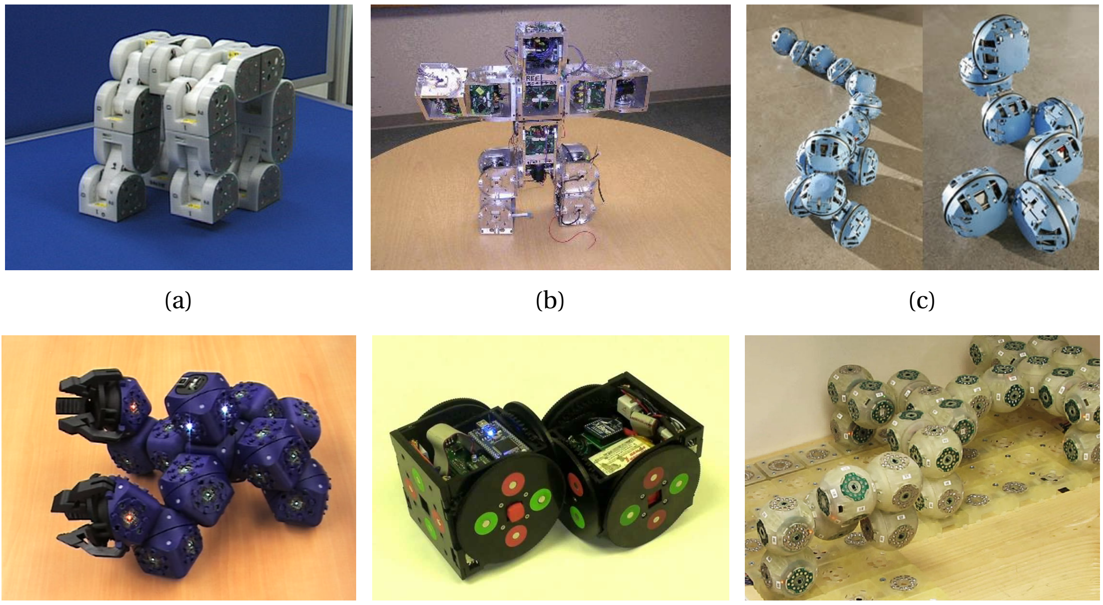
Robot was the M-TRAN (Modular-TRANsformer) robotic system [136, 102] (Figure 2.3.a). The M-TRAN design is clearly one of the most sophisticated and successful ones and it was influential for a number of upcoming modular and self-reconfigurable robots (e.g. Em-cube [4], SuperBot [180, 194]).

In the last decade the number of Reconfigurable Modular platforms has been growing steadily, as shown in Figure 2.2. While the development of a few of these platforms reached only early prototyping stages, some proved to be robust robotic systems used for research or as educational kits.

SuperBot modules (Figure 2.3.b) integrate the necessary features from both M-TRAN and CONRO to accomplish multimodal locomotion [194]. Each SuperBot module has three joints; the middle joint can rotate continuously in both directions, largely increasing 3D reconfiguration capabilities. ATRON (Figure 2.3.c) has 1-DoF modules that align to a lattice inspired by the Rhenium Trioxide crystal lattice [146]. Thanks to its spherical shape and unlimited rotation, ATRON is capable of performing fixed topology locomotion (e.g. cars, walker, snake) and locomotion by self-reconfiguration (cluster flow) [23, 22].

Molecubes [264] feature a diametrical joint inside a regular cube, i.e. with the axis along the cube's largest diagonal. This leaves space for six possible connector faces, i.e. all available sides of the cube. Molecubes have been redesigned to be relatively easy to manufacture and assembled, have been made open-source [266, 265], and new specialized modules have been developed (e.g. gripper modules shown in Figure 2.3.d).

Sambot [231, 232] and SMORES (Self-assembling MOdular Robot for Extreme Shape-shifting) [46, 114] have both mobile and hybrid-type configuration properties. SMORES is capable of


(a) (b) (c)

(d) (e) (f)

Figure 2.3 – More examples of Modular Robots: (a) M-TRAN. (b) SuperBot. (c) ATRON. (d) Molecubes. (e) SMORES. (f) Soldercubes. Pictures from [138, 180, 134, 265, 223, 142]

emulating many of the other existing systems and promises to be a step towards a universal modular robot.

Using the CKBOT [183], Sastra et al. showed for the first time dynamic legged locomotion driven only by body articulation, using passively compliant leg attachments [181]. Recently, White et al. used CKBOT to demonstrate the efficacy of a method to experimentally determine the stiffness matrix for chain style reconfigurable robots [235].

Soldercubes are designed with manufacturability for large batch production in mind [142]. They integrate a self-soldering connector which weights only 2g and use three types of modules (actuation, structural, and energy module).

The Cubelets modular robotics construction kit toy is a commercial product designed to introduce the ideas of tangible distributed computing [188]. The same company, Modular Robotics, released MOSS: a highly heterogeneous system that uses spheres and magnets to connect modules, create ball joints and hinges, and transfer power between units [129].

The recent widespread availability of rapid prototyping machines, cheaper robotic kits, and open source electronic boards and microcontrollers pushed forward the development of printable modular robots [48], on-demand custom printable robots [117], and design-build-test-disassemble robots [24].

An exhaustive review on Modular Robotic Systems was recently published by Ahmadzadeh et al. [2].

2.2 Roombots: a Self-Reconfigurable Modular Robot for a Living Environment

We are developing Roombots (RB) with the aim of assembling furniture-like structures that can provide morphing and locomotion capabilities to the elements that compose our living environment. We are working towards the idea of an active environment, where typical "roomware" components are merged and enhanced by elements from robotics and information technology to build intelligent furniture and other components of our daily life. The ultimate goal of the Roombots project is to build adaptive robotic modules that can autonomously connect into different shapes, such as stools, tables, or sofas. Using intuitive interfaces, end-users are be able to give high-level commands to trigger reconfiguration sequences or ask the robot to move around. Roombots structures could also automatically adapt to the user's needs, for instance by following an elderly person around the house to serve as support in case of sudden loss of balance.

Roombots is a hybrid-type Self-Reconfigurable Modular Robot that uses a standard cubic lattice. Each of its modules occupies two grid units and has three degrees of freedom actuated by DC motors. The clever placement of the rotating axes of the actuators allows a single Roombots module to perform lattice locomotion on a planar surface and over concave corners. Collaboration between two modules, forming a Roombots metamodule, is needed to pass convex edges. Roombots modules connect using 4-way symmetric, hermaphrodite

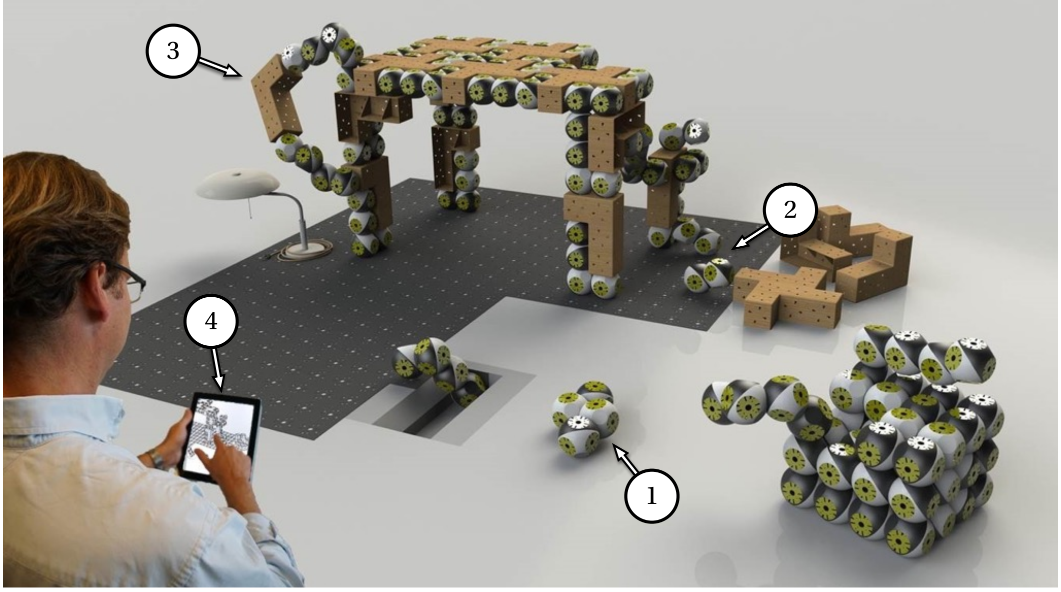


Figure 2.4 – Main areas of research of the Roombots project shown for the automatic assembly of a table made of Roombots modules and lightweight passive elements: (1) Locomotion on non-structured ground (off-grid). (2) (On-grid) locomotion through self-reconfiguration. (3) Collaborative manipulation of passive parts. (4) User interfaces. Tiles on the floor are connector ports that can be used to perform on-grid locomotion. Image adapted from [204].

Active Connection Mechanisms based on mechanical latches. Each robotic unit has 10 sockets that can be equipped with passive or active connectors. Roombots modules are autonomous robots, each with its own power source, computation boards, and wireless communication device. In the usual configuration, a computer acts as central host to generate motor commands and coordinate the movement of the SRMR.

A key element for the success of our robotic furniture is the combination of active units (Roombots modules) with passive objects, as shown in Figure 2.4. These passive objects can be anything from structural elements to lightweight blocks, to aesthetic parts, to existing furniture equipped with connectors. The combined use of active and passive parts helps reduce the overall number of active robotics modules required to construct a specific structure, also making it lighter, structurally stronger, and simpler than one entirely made of Modular Robots. Connector "ports" play an important role in this scenario. Some of these ports are part of the Roombots modules as active and passive connectors. Others are distributed around the room as passive connectors in the floor, walls, ceilings, and on the lightweight elements.

2.3 Lola-OP™ Reconfigurable Modular Snake Robot

For some studies reported in Chapter 10 we used different versions of the Lola-OP™ robot. Lola-OP™ is a Modular Snake Robot [120, 122] composed of a series of n 1-DoF modules connected to each other with a twist shift of 90° on their rotating axes (Figure 2.5). This robot is built using off-the-shelf Dynamixel AX-12A® actuators and Bioloid plastic structural frames [12]. The choice of off-the-shelf building parts allowed us to dramatically reduce the development time, to have commercial-grade robustness for the different components, and to decrease the overall cost and assembly-time for one copy of the robot compared to custom-made solutions.

We considered 8-DoF Lola-OP™ robots, with eight actuated modules. As explained in Chapter 6.3, we modified the original design of the robot by including seven in-series compliant elements between the different actuators.

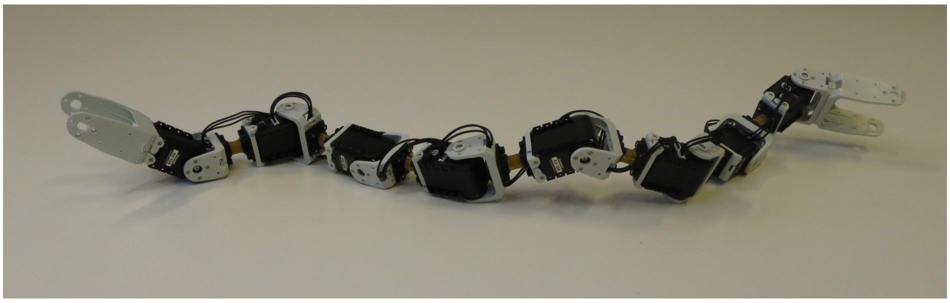


Figure 2.5 – The hardware implementation of the 8-DoF Lola-OP™ robot with eight degrees of freedom and in-series compliant elements.

Mechanical Challenges Part II

3 Overview of the Design of Roombots

In this chapter we give a short overview of the hardware components that are included in each Roombots module. Most of the design choices presented in this chapter were already defined by other members of the Biorobotics Laboratory (EPFL) before the work presented herein. They are reported here for clarity. These choices defined certain constraints that we had to consider during the mechanical redesign and improvement of some sub-parts of the robot. We will point to the next chapters for detailed descriptions of our contributions.

In Section 3.1 we give a description of the principles that lead to the first implementation of the Roombots modules.

Section 3.2 gives details about the specific orientation of the three degrees of freedom (DoF). These allow one module to use on-grid locomotion to move to any place on a flat (horizontal or vertical) surface and overcome concave corners. Collaboration between two Roombots modules is required only to pass convex corners.

The actuation system is briefly described in Section 3.3. More details and improvements to this system are presented in Chapter 4. The Roombots's DoF are actuated by electric DC motors capable of continuous rotation, which is potentially advantageous for motion planning and locomotion tasks.

Details about the Active Connection Mechanism (ACM), which provides the Roombots modules with Self-Reconfiguration capabilities, are given in Section 3.4. The Roombots ACM is a 4-way symmetric hermaphrodite connector based on mechanical latches. We can include up to ten active or passive connection ports in each module. In Chapter 5 we describe how we improved this crucial sub-part.

In Section 3.5 we shortly explain the electronics contained within each module. We describe the placement and connections of each board. The need for a wired inter-module communication bus, along with the choice of having continuous rotation capabilities, influenced the mechanical design of the gearbox-actuator system.

Finally, in Section 3.6 we analyze the limitations of the original Roombots design and describe our novel contributions giving a list of which sub-parts have been improved in the last four years.

Some parts of this chapter have been adapted from:

[204] A. Spröwitz, R. Moeckel, M. Vespignani, S. Bonardi, and A.J. Ijspeert. "Roombots: A Hardware Perspective on 3D Self-Reconfiguration and Locomotion with a Homogeneous Modular Robot". In *Robotics and Autonomous Systems*, Volume 62, Issue 7, July 2014, Pages 1016-1033, 2014.

My original contribution: Conceptual contribution, Experimental setup, Hardware experiments, Partial writing.

3.1 Roombots Module Design

One of the main challenges in mechatronic system integration is identifying appropriate trade-offs between different potential features of a module and the need to have a compact design that can include all components in a limited volume [208]. With today's technology it is not possible to design a general-purpose Self-Reconfigurable Modular Robot that is good for every possible scenario. Choices have to be made based on the application scenario, and these will greatly influence the whole design process.

The implementation of the Roombots modules has been based on the following potential applications [200]:

- **Reconfiguration and locomotion through reconfiguration.** This application guided many aspects of the design, starting from the choice of lattice (grid) to which the modules align. Self-reconfiguration movements are influenced by the number and choice of connection ports, number and orientation of the degrees of freedom, and the ability of two or more modules to collaborate to achieve a certain goal. The choice of kinematic structure dictates the reachable space and the collision space of each robot. Reconfiguration capabilities are also determined by the type of joints used in the system (e.g. revolution vs. prismatic joints) and their range (for revolute joints, this also means oscillation vs. continuous rotation capabilities). This application has also an influence on the choice of actuator strength: modules are more versatile if their joints have enough torque to lift long levers of modules.
- **Locomotion on unstructured terrains.** For this type of locomotion, which happens outside of the lattice and does not involve the use of grippers to attach to fixed structures, the requirements are common to those of many mobile robots. The modules need enough torque per joint to be able to lift the structure from the ground, but they also need to be fast enough if dynamic movements are required. Continuous rotation can be exploited for locomotion tasks by using parts of the modules as wheels. Considerations on connection ports similar to the ones above also have an effect on the type and number of morphologies that can be created with a limited number of modules, and on the ability to actively change morphology by self-reconfiguration to adapt to different terrains. This type of application also requires fully autonomous modules, also in terms of power source.
- **Roombots as building blocks for furniture.** This target application partially influenced the dimensioning of the robot, including the choice of lattice size and actuator strength. Additional Roombots features that are advantageous for applications as building blocks for furniture include the ability of ACMs to go into a “locked state” that does not require energy to keep the connection and the possibility to intrinsically compensate for tensile and pressure forces.

Based on these considerations and others not listed here, Roombots modules have been designed with the following characteristics:

- **A regular cubic lattice of side 110 mm.** Each Roombots module takes a volume of two cubic units (with a size of 220×110×110 mm).
- **Three continuous rotational degrees of freedom, using electric DC motors.** In the worst condition, these actuators allow a module to lift a lever made of 1.5 Roombots in addition to its own weight. Two of the three degrees of freedom are diametrical. The third one allows neighboring-cube swiveling.
- **10 sockets that can house passive or active 4-way symmetric connection ports.** Active Connection Mechanisms (ACMs) use mechanical latches to create the connection with the target connector. Roombots ACMs are hermaphrodite, meaning that both male and female parts are on the same connector. This means that any ACM can connect to any other ACM. In addition, ACMs can also connect to passive ports, which are basically female connectors.
- **On-board computation and power source.** Each module is fully autonomous and has its own set of control boards and batteries. High level commands are coordinated from a central host that is able to communicate individually with each module.

Each of these sub-parts is explained with more details in the next sections. Figure 3.1 shows an exploded view of one Roombots module. A summary of the specifications of one Roombots module is reported in Table 3.1.

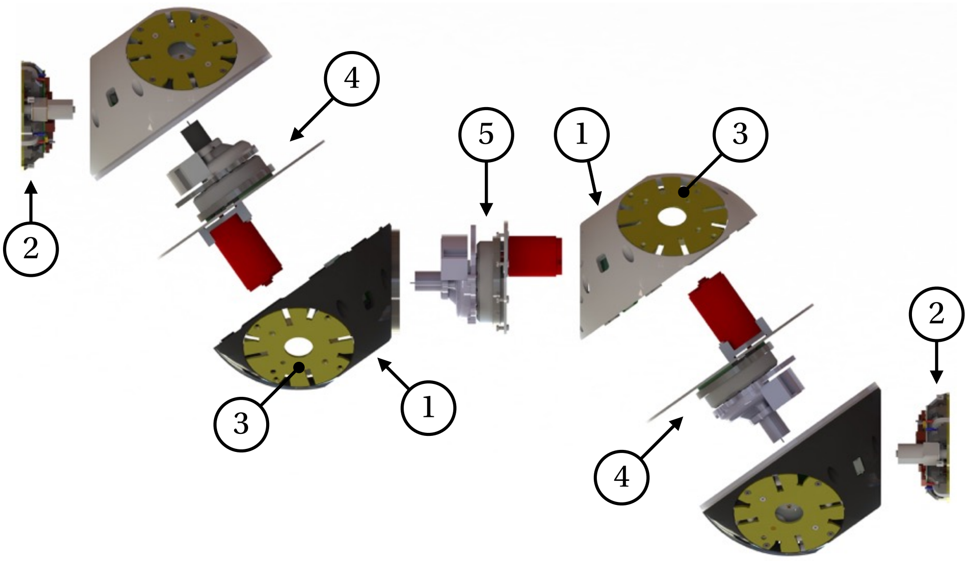


Figure 3.1 – Rendered exploded view of a Roombots module. (1) Rapid prototyped shells. (2) ACMs. (3) Passive ports. (4) Outer (diametrical) actuators with 305:1 reduction and absolute encoder. (5) Inner actuator with 366:1 reduction and absolute encoder.

Table 3.1 – Hardware specifications of a Roombots module.

Specification	**Value**
Overall dimensions	220 × 110 × 110 mm (2 lattice cubes)
Weight	1.4 kg
Degrees of freedom	3 (continuous rotational)
Outer motors	Faulhaber 2342 012 CR
Inner motor	Faulhaber 2232 012 SR
Outer gearboxes reduction	305:1
Inner gearbox reduction	366:1
Outer DoF speed (No load)	26.6 RPM
Inner DoF speed (No load)	19.4 RPM
Outer DoF nominal torque	4.9 Nm
Inner DoF nominal torque	3.6 Nm
Number of connection ports	10 (active or passive)
Active connection type	4-way symmetric, hermaphrodite based on mechanical latches
Communication	Bluetooth
Energy source	4-cell 1200 mAh LiPo battery; autonomy ~1 hour

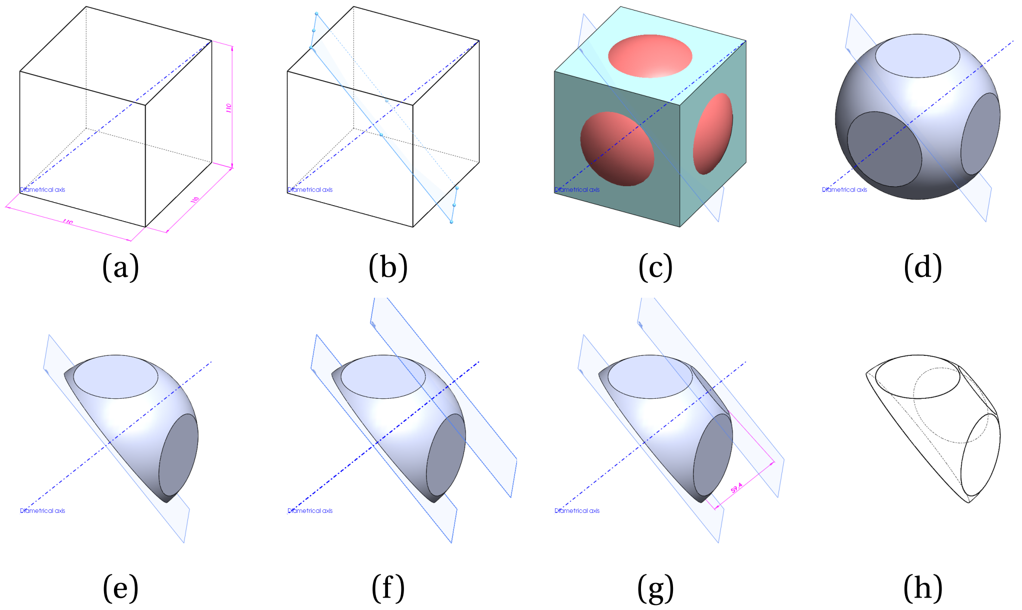
3.2 Roombots DoF and Housing

The Roombots Self-Reconfigurable Modular Robot uses a regular cubic lattice (grid) of size 110×110×110 mm. Each module extends for two lattice cubes, similar to M-TRAN modules [136], and is composed of four coupled hemispheres. Figure 3.2 shows the steps necessary to generate the external shape of each hemisphere. The process starts with a cube that has the lattice dimensions (side equal to 110 mm). We extract a diametrical line[1] from this cube; we then consider a plane p passing through the center of the cube and orthogonal to this line. Using this plane, we cut in half the intersection between the cube of side 110 mm and a sphere of diameter 128 mm. Lastly, to allow two connected Roombots modules to move freely without any possible self-collision, we cut each hemisphere with a plane that is at an offset of 59.4 mm from plane p.

Four Roombots hemispheres are connected together as shown in Figure 3.3. The axes show the location of the three degrees of freedom. Two degrees of freedom (m_0 and m_2) are diametrical, inspired by the DoF of the Molecubes [264]. The other (m_1) is in the center and allows swiveling of one half of the module with respect to the other, similar to the central DoF of SuperBot [194]. This DoF is advantageous for self-reconfiguration [200]. Hemispheres H_0 and H_3 are mechanically identical, differing only by color to allow the user to easily recognize the

---

[1] A diametrical line passes through two opposite vertices of the cube.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3.2 – Steps required to sketch a Roombots hemisphere.

orientation of each module, and have three sockets each for (active or passive) connection ports. In both H_1 and H_2 one of the sockets is used to fit the m_1 motor and gearbox, as seen in Figure 3.1. Thus, only two sockets are available in these two hemispheres, for a total of 10 available sockets for each Roombots module.

This DoF configuration was chosen because it allows one module to use on-grid locomotion to move to any place on a flat (horizontal or vertical) surface. Roombots is one of the very few SRMRs that can do this. The resulting movement is a combination of a zig-zag motion, as we explain in [19]. By using a sequence of DoF movements that we calculated using inverse kinematics, a single Roombots module can also overcome concave corners (Figure 3.4). Collaboration between two Roombots modules is required only to pass convex corners[2]. This requires Roombots actuators to be able to produce sufficient torque to lift at least one additional module.

3.3 Motor and Gearbox

In this section we give an overview of the original Roombots actuation system (referred in the following as Gearbox A) as designed by Dr. Alexander Spröwitz [200]. The motor and gearbox system will be covered with more details in Chapter 4, where we present our original contribution.

The actuation is a critical sub-part of a Roombots module and its design is highly influenced

---

[2] A description of these locomotion through self-reconfiguration tasks is reported Appendix B.

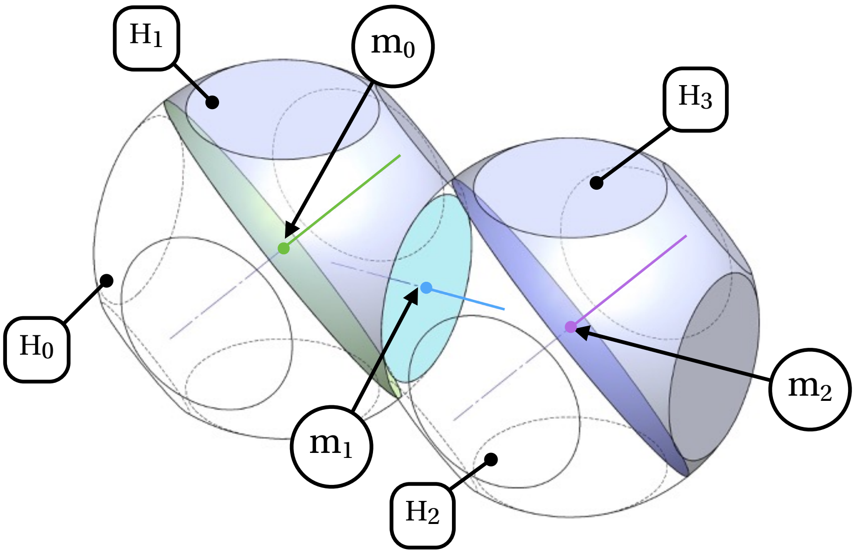


Figure 3.3 – Naming convention and main axes in a Roombots module. Each module is composed of four hemispheres connected together by three DoF. Ten sockets are available for active or passive connection ports.

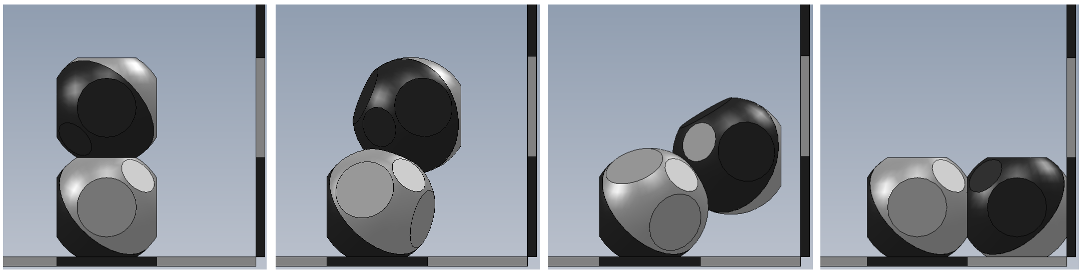


Figure 3.4 – Roombots module approaching a concave corner using its three DoF simultaneously to keep its rightmost connector parallel to the vertical surface at all times. Image from [204].

by other components. Among others, some of the main design constraints are:

- **Size**. There is limited space inside the housing; a change in size affects the design of most of the surrounding components.
- **Weight**. This is closely related to size. A larger motor-gearbox system can usually provide more output power; however it will also increase the overall weight of each module and therefore the required motor torque.
- **Torque vs. speed**. Robotic systems are usually designed with high reduction ratio gearboxes to increase the output torque of the motor to the level required for the specific application. Higher reductions typically require more space. Additionally, while increasing the output torque, they also reduce the output speed, which reduces the dynamics of the system.
- **Hollow design**. In the electronic design of the module (Section 3.5) we decided to use electrical slip rings to transfer power and data between hemispheres. This put a hard constraint on the gearbox design as it must be hollow in order to allow the passage of the wires.
- **Type of actuator**. Each type of actuator has a specific power density, but also requires a number of electronics and components to work.
- **Cost**. The goal of a SRMR is to have many modules collaborating in order to perform a specific task. In order to have a scalable production, the cost of the single module has to be reasonable.

After several design iterations, it was concluded that off-the-shelf gearboxes are not a viable option given the reduced available space and the required output torque. Instead, a custom gearbox was designed using worst case scenario calculations.

Based on the estimated weight of a module (approximately 1.5 kg), the required torque to lift a lever made of two Roombots modules was evaluated to be 4.9 Nm for the diametrical degrees of freedom and 2.5 Nm for the central one [200]. The original Roombots gearbox design [200], which will be referred to as “Gearbox A” in this thesis, was implemented with a double-stage planetary gearbox with a pre-set of spur gears. An exploded view of Gearbox A is shown in Figure 4.1 in Chapter 4. The design uses a low cost combination of 3D-printed ABS and off-the-shelf plastic gears. The outer (diametrical) degrees of freedom are driven by Faulhaber FH2342 DC motors and have a reduction of approximately 305 : 1. The inner (central) DoF has almost an identical design except for one spur gear with a different number of teeth which changes the reduction to 366 : 1, and a smaller more compact motor (Faulhaber FH2232). The motors are shifted off the central axis to allow a hollow axis to pass through the center of the gearbox.

More details on the assembly, new proposed versions of the gearbox, and a stress test of the system are reported in Chapter 4.

3.4 Active Connection Mechanism

Similarly to the previous section, here we briefly introduce the Active Connection Mechanism (ACM) which allows Roombots modules to self-reconfigure. In Chapter 5 we will describe in more detail classification methods for ACMs.

The Roombots ACM (Figure 3.5) was partially inspired by the AMAS [215] connector, based on physical latches. Given the desire to use Roombots to create modular robotic furniture, several possible design options were ruled out. The original Roombots ACM, that we call ACMv3 in accordance with [200], is a 4-way symmetric hermaphrodite connector. The connection is established by four mechanical latches (male part of the connector) that can grab the destination connector through some grooves (female part of the connector). In the ACMv3 both male and female parts coexist in the same connector, meaning that any ACM can connect to any other. The connection is unilateral, meaning that it can be established by just one ACM without the need of a collaborating action by the target connector (although a double connection from both ACMs gives a stronger bond). The downside of unilateral connection is that a non-responsive connected module cannot be disconnected.

In addition to ACMs, we can have connectors that have only the female part. We will call these connectors *passive ports*. They are simple fr-4 (glass fiber) plates cut with the right pattern. The advantage of passive ports is that they are extremely cheap and do not require power nor control. They can be mounted on some of the available sockets on a Roombots module, or included in the environment to create a grid substrate for on-grid locomotion, or added to passive parts or existing furniture, as presented in Chapter 6.

One key feature of the ACMv3 is that, once in the connected state, it is not backdrivable. This is advantageous because no power is required to keep a connection active. This is especially

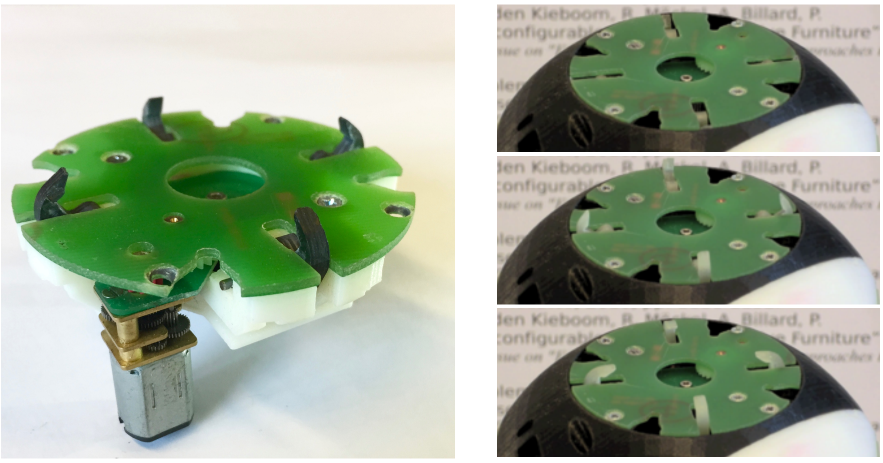

(a) (b)

Figure 3.5 – Active Connection Mechanism (ACM): (a) ACMv4 (more details in Section 5.3). (b) Latches emerging from the connector.

useful in the living furniture scenario, in which a certain shape or configuration needs to be held for a long time.

We can include up to ten ACMs in one module, although we noticed that for most experiments two ACMs placed on H_0 and H_3 are sufficient. We can use the other 8 available sockets as passive ports.

3.5 Electronics

The original version of the Roombots electronics were designed by Dr. Rico Moeckel. A detailed description can be found in [204]. Figure 3.6 shows a simplified diagram of the Roombots electronics.

Each module is autonomous and able to run its locomotion controller with a set of parameters. However the high level control (i.e. motion planning) is done by an external computer (PC). Communication between modules and PC is done via a wireless Bluetooth® link. The Communication Board (BT) transmits the received commands and control-parameter sets to the other electronic boards using a wired communication bus. Electronic slip rings (SR) allow the distribution of the bus signal to every hemisphere.

Motor Boards (MB) can receive locomotion commands and are responsible for the low-level control of the DC motors and for running a central pattern generator controller. There are three motor boards, one for each actuator. Each active connector is equipped with an ACM Control Board (ACM) that controls the opening and closing of the mechanical latches.

Each module is powered by a 4-cell, 1200 mAh LiPo battery (BAT) which guarantees an autonomy of at least one hour. The battery level is regulated and stabilized by a Power Board (PB).

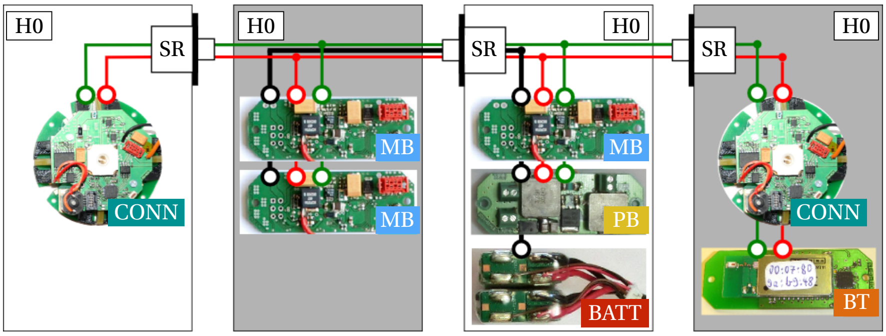
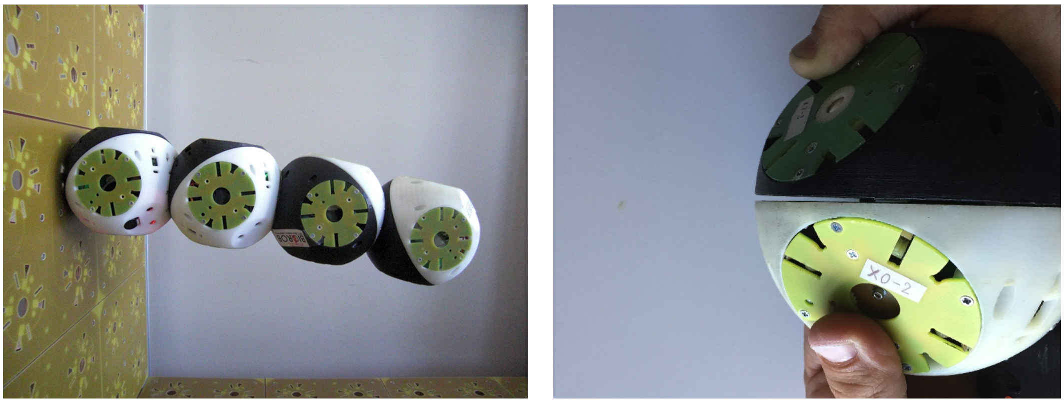


Figure 3.6 – Roombots electronics: active connection mechanism control board (CONN), motor-driver board (MB), power board (PB), batteries (BATT), Bluetooth communication board (BT), and slip ring (SR). Image adapted from [204].



(a) (b)

Figure 3.7 – Bending of a Roombots module. (a) Cantilevered Roombots metamodule under the effect of gravity. (b) Gap between hemispheres caused by the deformation of the main plates.

3.6 Proposed Mechanical Improvements

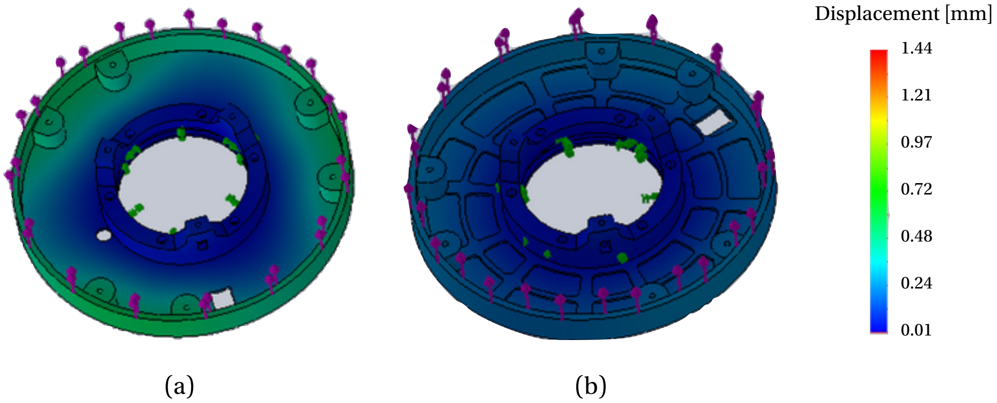
In this section we discuss limitations and improvement opportunities that we observed in the original design of Roombots modules. For each of them we clarify whether the new development was required to fix a technical/performance issue, to provide new features, or to allow for a more scalable production.

The main limitation of having successful on-grid locomotion and self-reconfiguration experiments are various sources of deflection, backlash, and position uncertainty within a module or in the connection with other modules or grid elements. These uncertainties add up and cause the open-loop controller to fail to accomplish the desired task. The typical mode of failure is a misalignment between an ACM and its target connector, which prevents the module from successfully executing the next step of a self-reconfiguration process.

While exploring ways to also compensate for this misalignment via software [9], we identified three main mechanical sources of failure: bending of the "main plates" (as seen in Figure 3.7.b), backlash in the gearboxes, and limited range and self-alignment properties of the ACM.

Main Plates

Figure 3.7.a shows the deflection of a cantilevered Roombots metamodule under the effect of gravity. One of the most evident causes for this is the deformation of the main plates, which are fr-4 (glass fiber) plates used in the diametrical degrees of freedom to connect the motor-gearbox system to the external housing (shells). Each DoF has two main plates that slide on top of each other. Their deformation produced a wide gap between hemispheres, as seen in Figure 3.7.b.



(a) (b)

Figure 3.8 – Finite Element Analysis (FEA), using SolidWorks Simulation, of a main plate under load. (a) Original flat design in glass fiber. (b) Proposed design in aluminum with reinforcement ribs.

As a first attempt, we tried to analyze the geometry and material properties of these main plates. We used Finite Element Analysis (FEA) to evaluate different designs using reinforcement ribs and aluminum as building material. We simplified the CAD design by including just one main plate, locked at the center (at the points where it is attached to the gearbox), and a section of the external housing. Although Figure 3.8 shows a theoretical threefold reduction in the deformation caused by purely normal forces applied to the shell element, tests performed on a prototype of this new design revealed that this approach was not sufficient to reduce the gap between hemispheres.

As an alternative solution, we constrained the main plates at their edges, at the point of maximum lever arm. After evaluating several design options, we modified the main plates and shell design to include a POM ring that holds the two plates together (Figure 3.9). This method effectively eliminated the gap between hemispheres and is currently part of the Roombots module design.

Additionally, using a clear material for this plastic ring allowed us to create a semi-transparent band around each diametrical degree of freedom. This feature was used in the work presented in Chapter 12 to provide visual feedback to the user through LED lights mounted inside the module.

Gearboxes

Gearboxes have been the object of several design improvements, described in details in Chapter 4. One of the main reasons that led the redesign process was the amount of backlash for Gearbox A, estimated at approximately 1.5 degrees [9], increasing over time, and variable between samples. As already mentioned before, this was considered one of the main causes of misalignment during self-reconfiguration experiments.

We adapted the design, materials, and production techniques to reduce backlash by improving the matching between gears, but also to allow for a faster and more scalable production. To further decrease the effect of backlash, we included absolute encoders at the output of each gearbox to compensate for the backlash via closed-loop control. This required a complete redesign of the system to reduce the height of the gearbox in order to fit the absolute encoder. We took this redesign as an opportunity to also increase the DoF torque and speed by using more powerful motors.

Chapter 4 gives an in-depth description of all these mechanical changes.

ACMs

The last sub-part that we identified as limiting for on-grid locomotion and for the self-reconfiguration process is the Active Connection Mechanisms.

The ACMv3 design is already quite packed, taking most of the available space in each socket. This does not leave too much room for a complete redesign without affecting all the rest of Roombots's components. Instead in the ACMv4 we tried to get a smaller degree of improvement by changing the latches design and actuation. This improvement alone allowed us to complete a completely autonomous on-grid locomotion sequence with a Roombots module on a vertical grid for the first time.

In order to increase the self-aligning properties of the connector, we included small magnets at the center of each ACM or passive port. This hybrid ACM dramatically increased repeatability

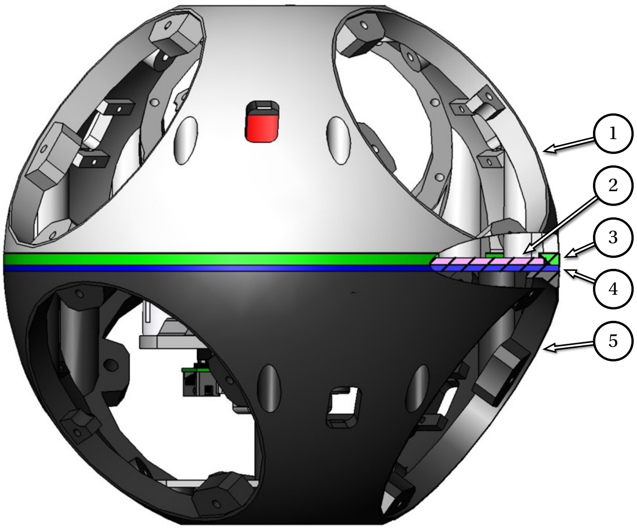


Figure 3.9 – Broken out section of the POM ring. (1) H_2 hemisphere. (2) Main plate fixed to H_2 (pink). (3) POM ring (green). (4) Main plate fixed to H_3 (blue). (5) H_3 hemisphere. Colors shown for descriptive purposes only. Only half of a Roombots module is shown.

for on-grid locomotion experiments. This is analyzed in Chapter 5.

Shells

The original Roombots shells (external housing) were manufactured using 3D-printed ABS plastic. After a minor redesign of the internal features in order to accommodate the POM ring that holds the main plates together, we decided to produce the new ones with laser sintered Polyamide (PA2200). We also considered the possibility of using injection molding; however, given the complexity in our parts and the small number required, we discarded the idea because it was not cost-effective.

Electronics

The Roombots electronic boards have been completely redesigned over the last years and their new versions are currently being tested. We will not go into the details of the changes as it was work done by other members of the team, namely Dr. Rico Moeckel and Mehmet Mutlu, with the assistance of Dr. Alessandro Crespi and André Badertscher. The main features of these new electronics are more stable wireless Bluetooth chips (the previous ones had disconnection issues), a better power management board, and the possibility for Motor Boards to connect to the new absolute encoders that have been included at the output of the gearboxes.

External Attachments

One aspect that we investigated is the possibility of attaching external parts to Roombots structures.

We imagine structures not only made of active robotic units but also of passive elements to which Roombots can attach using their ACMs. The most simple example is structural passive parts equipped with passive connection ports. These parts would be particularly useful in the “Roombots as building blocks for furniture” scenario, as they can be used to reduce the total number of Roombots modules needed to build a structure (thus decreasing the overall cost and weight of said structure). Moreover, by cleverly placing these structural parts, Roombots modules could create constraining connections that would lock their degrees of freedom [200], allowing them to hold their position even without actuation.

We can also “augment” existing furniture by placing compatible connectors on it and attaching Roombots modules to provide it with locomotion and manipulation capabilities. Some examples of this use are shown in Chapter 6. We thought about possible specialized parts that could be directly connected to Roombots modules to provide them with specific new functionalities. These parts could include exteroceptive sensors or new types of actuators.

Finally, we explored the use of compliant passive extensions that could be used to improve locomotion performance. We give more details about the implementation of these compliant extensions in Section 6.3 and describe locomotion studies in Chapter 10.

3.7 Conclusions

In this chapter we described design choices and limitations of the first version of the Roombots modules [200]. The mechatronic design of a Self-Reconfigurable Modular Robot is challenging because each component of its modules (e.g. actuators, connection mechanisms, control and communication electronics, power source) is tightly linked with the others and with the desired final application, and the smallest change affects the whole system. Each design choice becomes an integral part of the system, limiting the possibility of future non-invasive modifications.

In Section 3.6 we described some of the limitations and issues that we encountered with the original Roombots module design. We then briefly reported our novel contributions on the mechanical redesign of critical sub-parts. In the next chapter we will cover in more detail the Roombots actuation system and the improvements made to the gearbox.

4 Actuation System

In this chapter we describe the challenges related to the design and manufacturing of a custom gearbox-actuator system.

In Section 4.1 we explain the challenges in the actuation design for a SRMR. In homogeneous systems the actuators have to be sized by taking into account worst conditions within the expected application scenarios (e.g. long levers of connected modules).

In Section 4.2 we describe the requirements that led to the development of the custom Roombots actuation. We first explain in detail the original Roombots gearbox (Gearbox A). We then report three new design iterations (Gearboxes B, C, and D) in which we reduced the overall thickness of the gearbox, changed manufacturing methods and materials of most of the components to improve quality and reduce assembly time, and added an absolute encoder at the output to measure the correct DoF position after backlash. In Gearbox D, built using brass and aluminum parts, we also changed motors to more powerful ones.

In Section 4.3 we report bench tests used to evaluate and compare the four different designs. The first two designs could not pass a load-holding test at 95% of the motor's nominal torque. Gearbox C successfully passed all the tests. Gearbox D was able to withstand a twofold increase in output load compared to the original design.

4.1 Introduction

Actuation is a fundamental component of a SRMR. Actuators and transmission mechanisms are embedded in each module and allow the overall structure to perform locomotion, self-reconfiguration, and in general to move and interact with the environment. Contrary to traditional robots, the general design rule in modular robotics is to keep modules simple and provide them with a restricted number of features. Modules normally have limited mobility and just a few actuators, and only when connected to other units they are able to generate more complex motion. Typically actuators occupy more than 50% of the volume and weight of a module [2]. and are one of the major obstacles to module miniaturization.

In homogeneous SRMRs each module is (at least mechanically) exactly alike the others. Contrary to biological systems, in which proximal actuators (e.g. muscles close to the center of the body) are stronger than distal ones in order to carry the different load, in homogeneous SRMRs all the modules are identical and provide the same power output. Therefore a clear limitation of this type of systems is scalability of the actuators' power. As the number of modules used in a structure increases, homogeneous actuators cannot provide enough power to produce a useful output (e.g. effective locomotion, manipulation of objects). Modules have to be designed with the application scenario in mind and according to the worst possible configuration (e.g. they must be able to provide adequate torques to manipulate long chains of other modules connected to them). It is therefore essential to select the right tradeoff between actuation capability of a module and the weight and size of its actuators.

Several types of actuation mechanisms have been used in modular robotics. However, the large majority of systems uses DC electric motors, either brushless [244], brushed [46], or stepper [77]. Electric motors are commonly available, their technology is mature, they are easy to control, and have a good power density at the centimeter scale. These types of actuators can be implemented as (i) servomotors with integrated gearbox, (ii) DC motors with off-the-shelf gearboxes, or (iii) DC motors with custom-designed gearboxes.

4.2 Roombots actuation system

Roombots modules feature three rotational degrees of freedom operated by DC brushed electric motors. This choice was influenced by application constraints (considering the envisioned domestic use for the modules) and by the list of design constraints reported in Section 3.3. Roombots's degrees of freedom allow for not just oscillating, but also continuous rotatory movement. This feature can be advantageous during self-reconfiguration and on-grid locomotion tasks, as a DoF can rotate in either direction to reach a desired angular position. Continuous rotation can be used also for off-grid locomotion; given the rounded shape of Roombots shells, the outer hemispheres can be used as wheels.

With self-reconfiguration tasks in mind, we designed Roombots's actuators with sufficient output torque to allow one module to lift at least a second additional module (four lattice grids).

A explained in Section 3.3, to fit this actuator requirement, we combined off-the-shelf motors with a custom-designed gear train. The gearbox is lightweight compared to commercially available spur or planetary gear trains of similar size.

The three DoF use two variations of the motor-gearbox system. The two diametrical DoF use more powerful motors (Faulhaber 2342) with a 305 : 1 reduction. This results in a theoretical nominal output torque of 4.9 Nm and no load speed of 26 rpm. The central degree of freedom has instead a smaller motor (Faulhaber 2232) but with a higher reduction of 366 : 1 that provides a nominal output torque of 3.6 Nm and no load speed of 19 rpm.

The construction of the gearboxes for diametrical and central DoF is exactly identical (to simplify the parts production), except for the first spur gear fixed to the motor's shaft which has 12 teeth for the outer motors and 10 teeth for the central one. For the rest of the chapter we will not make the distinction between the two variations. Given the higher output torque of the diametrical DoF, we use this configuration to explain the assembly and evaluate robustness and performance.

4.2.1 Gearbox A: Original Design

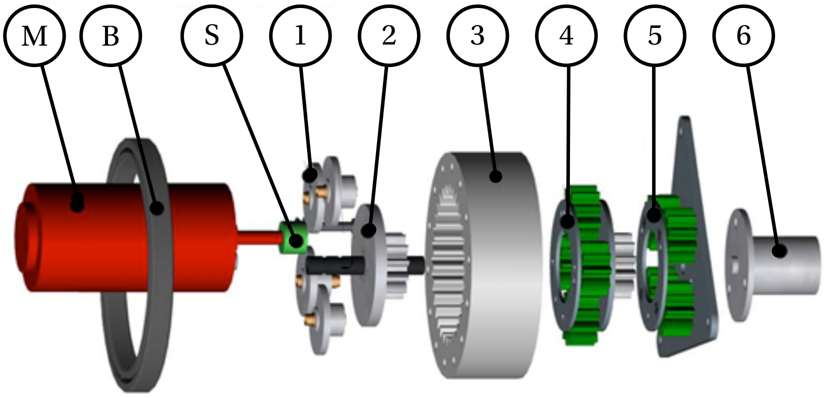


Figure 4.1 – Exploded view of the Gearbox A design. A DC brushed motor (M) drives a first drive chain of spur gears (S-1-2). A two-stage planetary gear train (3-4-5) further increases the reduction ratio. An internal hollow axis allows electrical slip ring wires (6) to pass through the gearbox. A thin-section ball bearing (B) guarantees smooth motion. Image adapted from [200].

We present here the design of the Gearbox A (shown in Figure 4.1), as implemented by Dr. Alexander Spröwitz [205].

The DC brushed motor (M), positioned off-axis compared to the center of the gearbox, drives an initial three-stage spur gear set (S-1-2). These are all off-the-shelf POM plastic gears with a module of $M = 0.5$. The gear (S), mounted on the motor's shaft, has $n = 12$ teeth. The motor torque is initially transmitted in parallel by two pairs of gears (1) to a $n = 50$ gear (2) placed centrally on the main axis of the gearbox. Each of the two pairs of gears is composed by two plastic double spur gears with $n = 28 : 12$. The transmission ratio of this first part of the gearbox is:

$$i_{spur} = \frac{28}{12} \cdot \frac{28}{12} \cdot \frac{50}{12} \approx 22.7 : 1$$

An in-series double-stage planetary gear set continues to gear down the output. The module is increased to $M = 1$ using an off-the-shelf 12-tooth spur gear that acts as sun-gear for the first planetary stage. This gear is manually fixed to the $n = 50$ gear. In both stages of the planetary gearbox the annulus (i.e. the outer ring with inward-facing teeth (3), with $n = -32$, 3D-printed ABS) is held stationary, the sun gear is used as the input, and the planet carrier is the output. For both stages, the planet carrier holds 4 off-the-shelf plastic planetary gears with $n = 12$. The sun gear of the second stage is directly fixed to the planet carrier of the first stage (4). This configuration gives the lowest gear ratio attainable with a two-stage planetary gear train, which is calculated as:

$$i_{planetary} = \left(1 - \frac{-32}{12}\right) \cdot \left(1 - \frac{-32}{12}\right) \approx 13.4 : 1$$

The number of teeth in the planet gears is irrelevant for this calculation.

The output gear ratio of the complete gearbox is therefore

$$i_{gearbox} = \frac{28}{12} \cdot \frac{28}{12} \cdot \frac{50}{12} \cdot \left(1 - \frac{-32}{12}\right) \cdot \left(1 - \frac{-32}{12}\right) = \frac{4743200}{15552} \approx 305 : 1$$

Gearbox A fits most of the requirements listed in Section 3.3. Its main characteristics are a high reduction in a compact design, a lightweight construction, a reduced cost compared to commercial products, and a hollow design that allows electrical slip ring (6) wires to pass through the gearbox. There are however some limitations with this design:

- Although most of the spur gears are off-the-shelf (as shown in Figure 4.2) their bores have to be manually reamed out to the correct size. The sun gears of the two planetary stages also have to be manually drilled in order to insert dowel pins necessary for their assembly. Not only these manual operations require time to prepare and assemble the components together, but they are also prone to imprecisions that affect the quality and the proper alignment of the assembly.
- Gearbox A has approximately 1.5 degrees of backlash caused by gaps and imperfect matching between gears. In this design the effects of backlash (i.e. a variation of the joint output position around the desired value) cannot be measured during run-time and greatly affect the module alignment during self-reconfiguration tasks.
- Despite its compact design, Gearbox A is 26 mm thick (motor and slip ring excluded) and occupies almost all the available space inside the hemispheres, limiting the introduction of additional components inside a module.

For these reason we redesigned the gearbox over several iterations. We report here only the major ones that have been used in the robot, that are all based on the implementation used for Gearbox A, although other designs have also been tested (e.g. with thee planetary stages and different modules, with a cycloidal drive).

4.2.2 Gearbox B: Compact Design

In the Gearbox B design (Figure 4.3) we aimed at reducing backlash by using better matched gears. We manufactured the gears that compose the two planetary stages in POM plastic, machined with a CNC mill. We iterated the design a number of times to find the optimal distance between gears. The machining of some parts helped to dramatically reduce the complexity and the time required for the assembly.

In this design we also changed the thickness of the gears (3), (4), and (5) in order to reduce the total height of the gearbox to 20 mm. Additionally, at this point we also included in the assembly the POM ring described in Section 3.6, which prevents bending of the main plates. The elements that compose Gearbox B are shown in Figure 4.4.

This design effectively reduced backlash, allowing to increase the alignment between modules in self-reconfiguration tasks. However, during extensive experimental sessions, friction caused the gearbox to heat up. Given the thermal properties of POM, this resulted in an expansion of some of the gears which increased even more the internal friction. Occasionally this increase in friction produced excessive tangential forces on the dowel pins in the assembly (5), which

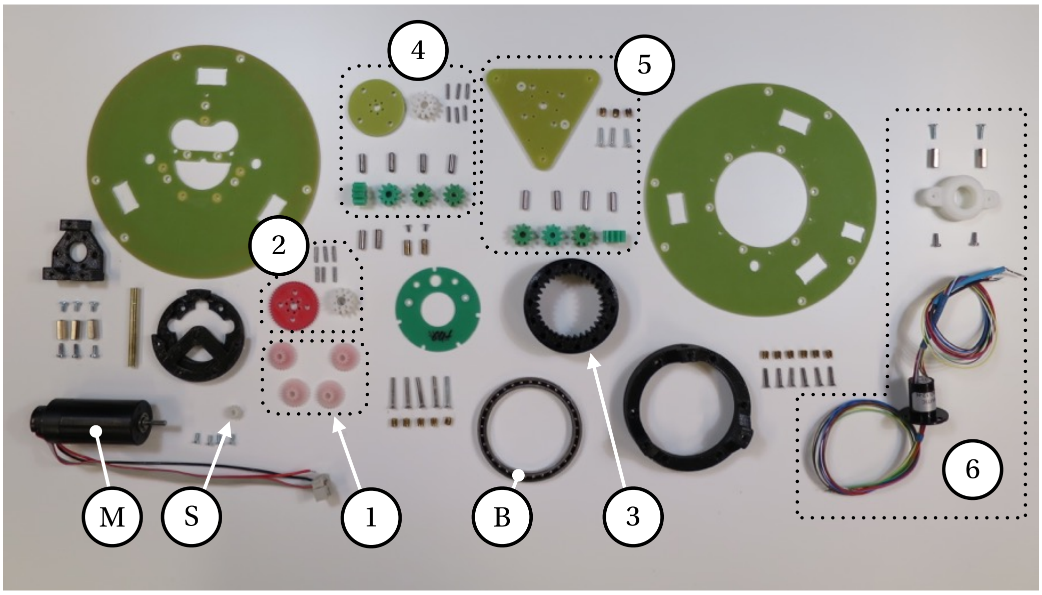


Figure 4.2 – 104 elements necessary to assemble Gearbox A. The labels correspond to the labels of the exploded view in Figure 4.1. Gears (S), (1), and in the assemblies (2), (4), and (5) are off-the-shelf. Annulus gear (3) is 3D printed.

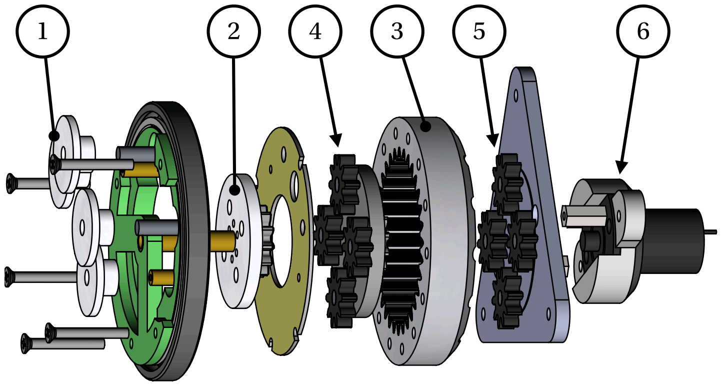


Figure 4.3 – Exploded view of the Gearbox B design (motor not shown). The labels correspond to the labels of the exploded view in Figure 4.1.

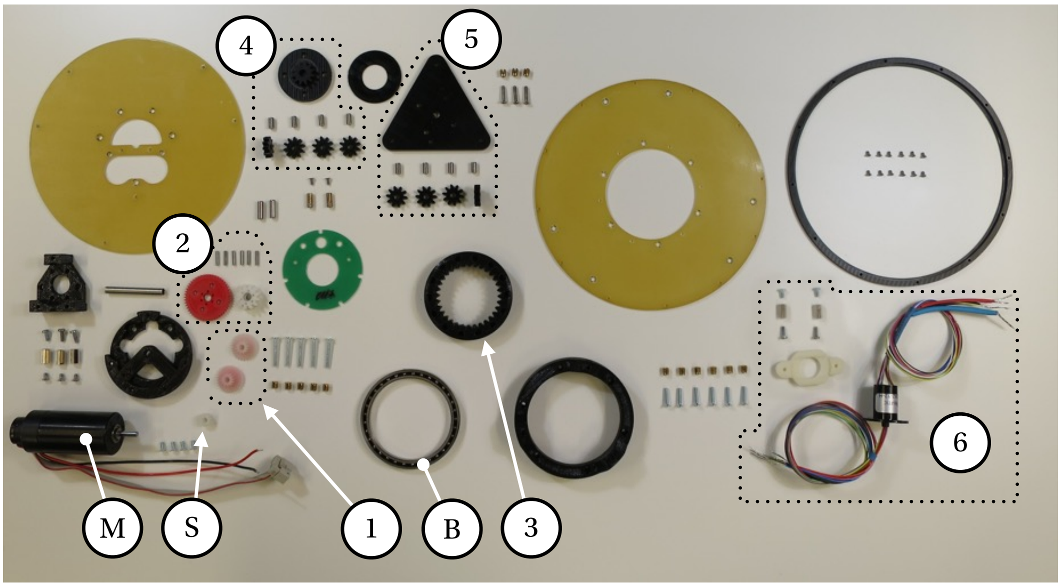


Figure 4.4 – 95 elements necessary to assemble Gearbox B. 13 additional elements are needed to assemble the POM ring (top-right of the figure). The labels correspond to the labels of the exploded view in Figure 4.1. Gears (S), (1), and in the assembly (2) are off-the-shelf. Gears in the assemblies (4) and (5) are machine-milled in POM. Annulus gear (3) is 3D printed.

detached them from their triangular support piece.

4.2.3 Gearbox C: Double Support Design

We further improved the design of the Roombots actuation by developing Gearbox C, shown in Figure 4.5. The characteristics of this design are the following:

- All the gears are now machined, which helped improve their teeth profiles and, consequently, their matching.
- The height of the gears is dependent on their position in the gear train. Gears on the second planetary stage are thicker than those on the first stage because they transmit approximately 3.6 times more torque.
- The dowel pins that connect the planet gears to the carrier now have a double support design that keeps them in place also when the tangential forces increase.
- Instead of trying to minimize the overall gearbox backlash by reducing the play between mating teeth, we included in the design a 12-bit capacitive absolute encoder (AMT203-V by CUI, Inc) at the output. This encoder can be used in combination with or in replacement of the DC motor encoder to precisely control the desired DoF position[1].

4.2.4 Gearbox D: Metal Design

Before mass-producing the actuation system to create many Roombots modules, we modified the design to provide more output torque and speed. We replaced the Faulhaber motors with more powerful Maxon motors. For the diametrical degrees of freedom we chose Maxon RE 25 DC motors, which can increase the output torque to 8.4 Nm (70% increase) and the no-load speed to 31.5rpm (21% increase). Similarly, for the central DoF we used a Maxon RE-max 24 DC motor that results in 4.3 Nm of nominal torque (19% increase) and 20.4 rpm (7% increase).

In order to hold such an increase in power we adapted the Gearbox C design to be manufactured in metal, using brass gears and aluminum support pieces. The redesign and the manufacturing of the first prototypes of Gearbox D were done by the lab's technician, François Longchamp, whom we thank for his support.

After a successful evaluation of this new design (reported in the next section), 30 copies of Gearbox D have recently been ordered from one of EPFL's professional mechanical workshops.

Table 4.1 summarizes the features of the different gearbox versions.

---

[1] This feature is currently being tested and has not yet been used to control the robot during self-reconfiguration tasks.

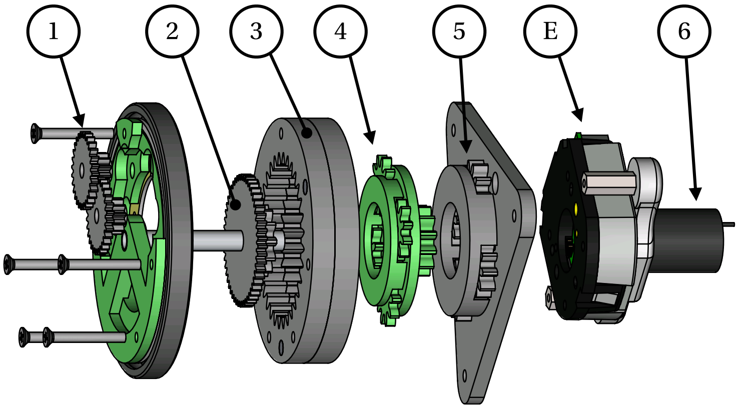


Figure 4.5 – Exploded view of the Gearbox C design (motor not shown). The labels correspond to the labels of the exploded view in Figure 4.1. This design also incorporates a 12-bit absolute encoder (E) at the gearbox output.

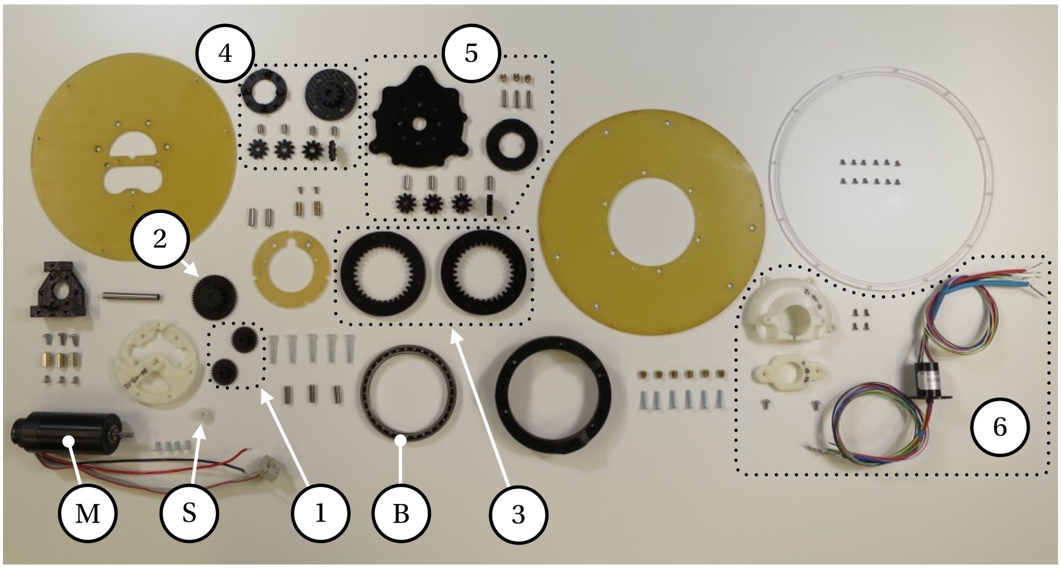


Figure 4.6 – 102 elements (13 for the POM ring). 89 elements necessary to assemble Gearbox C. 13 additional elements are needed to assemble the POM ring. The labels correspond to the labels of the exploded view in Figure 4.1. All the gears are machine-milled in POM.

Table 4.1 – Summary of the four Roombots gearbox versions. Test torque based on the results presented in Section 4.3. Easiness of production, easiness of design, and cost evaluations are qualitative, based on own experience. Gearbox A is not easy to produce because the off-the-shelf (ots) components have to be manually modified to fit the design. Some of these issues have been addressed in Gearbox B with custom designed milled parts. The double support design in Gearbox C makes the assembly much easier. Gearbox D has the production and design advantages of the previous versions; however the metal design makes the machining process much harder, significantly increasing the overall production cost.

	Gearbox A	**Gearbox B**	**Gearbox C**	**Gearbox D**
Designed by	A. Spröwitz	M. Vespignani		M. Vespignani + F. Longchamp
Gear materials	ots POM + 3D printed ABS	ots + milled POM + 3D printed ABS	ots + milled POM	Brass + Aluminum
Housing materials	3D printed ABS + glass fiber	3D printed ABS + glass fiber	3D printed ABS + POM + glass fiber	Aluminum + POM
Weight	230 g	240 g	270 g	370 g
Height	26 mm	20 mm		
Position sensors	Input relative encoder		Input relative encoder + output absolute encoder	Output absolute encoder
Driving motor	Faulhaber 2342			Maxon RE 25
Nominal speed	26 rpm			31.5 rpm
Nominal torque	4.9 Nm			8.4 Nm
Test torque	95% nom. torque	95% nom. torque	115% nom. torque	115% nom. torque
Easiness of production	- -	+ +	+ +	+
Easiness of design	-	+	+ +	+ +
Cost	$	$$	$$	$$$

4.3 Bench Test and Evaluation

In order to evaluate the different gearbox designs, we bench tested them according to a custom protocol. Figure 4.7 shows the experimental setup[2]. The gearboxes were isolated from the Roombots modules and fixed horizontally to a frame. At their output we mounted a pulley which was used to lift different payloads of increasing weight proportional to the nominal output torque. The pulley allows to invert the direction of the torque produced by the load.

We adapted the design of Gearboxes A[3] and B to also include the absolute encoder (E) for measuring the output position.

We controlled the motor using an Arduino Mega board with a DC motor shield. Additionally we used a thermocouple, taped on the motor housing, to measure the variation in temperature over time as an estimation of motor effort to drive the load and contrast the internal friction of the gearbox. Since temperature changes with a slow timescale, we also included a current probe that logs the current at the output of the Arduino motor shield. During the bench tests we logged the following variables:

- Desired trajectory (motor command)
- Input position, measured from the motor encoder (I)
- Output position, measured from the absolute encoder (E)
- Motor current
- Body temperature of the motor

---

[2] We thank François Longchamp for building the mechanical setup; Simon Hauser for helping to interface the sensors and to write the code, and for running the experiments; Mehmet Mutlu and Stéphane Bonardi for their technical assistance.

[3] With this simple design change (i.e. mounting holes for the absolute encoder and relocation of the spacers for the electrical slip ring) Gearbox A would not fit inside the Roombots hemispheres.

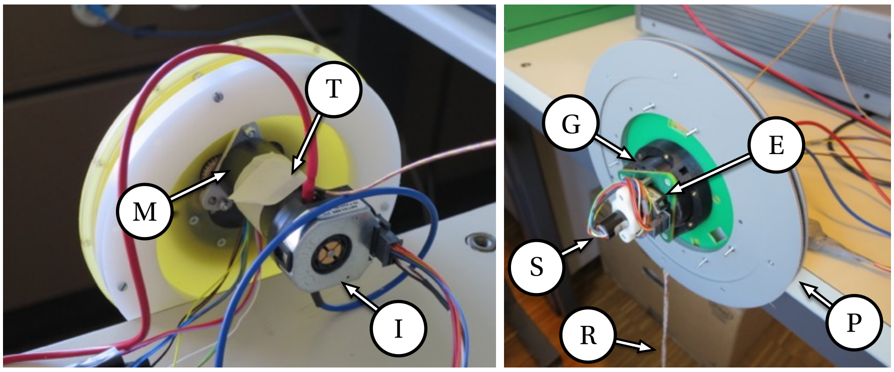


Figure 4.7 – Bench test setup: (M) DC motor. (T) Temperature probe. (I) Input encoder. (G) Gearbox. (E) Output encoder. (S) Electrical slip ring. (P) Pulley. (R) Rope. Not pictured: payload, regulated power supply, Arduino control board with DC motor shield.

We defined the following experimental protocol:

Test 1: Continuous rotation with no load

- 5 minutes of rotation at maximum speed in clockwise direction
- 5 minutes of rotation at maximum speed in counter-clockwise direction
- 5 minutes of rotation at maximum speed in clockwise direction
- 5 minutes of rotation at maximum speed in counter-clockwise direction
- 5 minutes of rotation at maximum speed in clockwise direction
- 5 minutes of rotation at maximum speed in counter-clockwise direction

Test 2: Oscillation with no load

- 2 minutes of ±360 degrees of oscillation at maximum speed
- 2 minutes of ±180 degrees of oscillation at maximum speed
- 2 minutes of ±90 degrees of oscillation at maximum speed
- 2 minutes of ±45 degrees of oscillation at maximum speed
- 2 minutes of ±22.5 degrees of oscillation at maximum speed

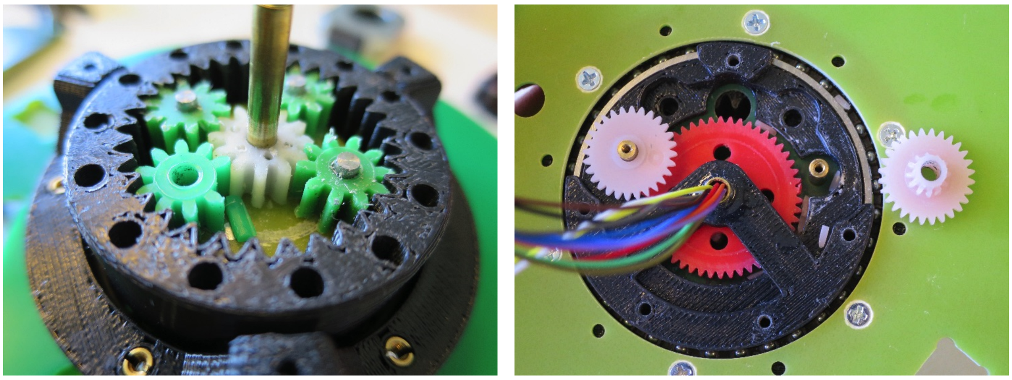
Test 3: Payload holding

- 50% of the nominal torque (2.5 Nm) for 5 seconds, 10 repetitions for each direction
- 70% of the nominal torque (3.5 Nm) for 5 seconds, 10 repetitions for each direction
- 95% of the nominal torque (4.7 Nm) for 5 seconds, 10 repetitions for each direction
- 105% of the nominal torque (5.2 Nm) for 5 seconds, 10 repetitions for each direction
- 115% of the nominal torque (5.7 Nm) for 5 seconds, 10 repetitions for each direction

Test 4: Increased payload holding with Maxon motor (only for Gearbox D)

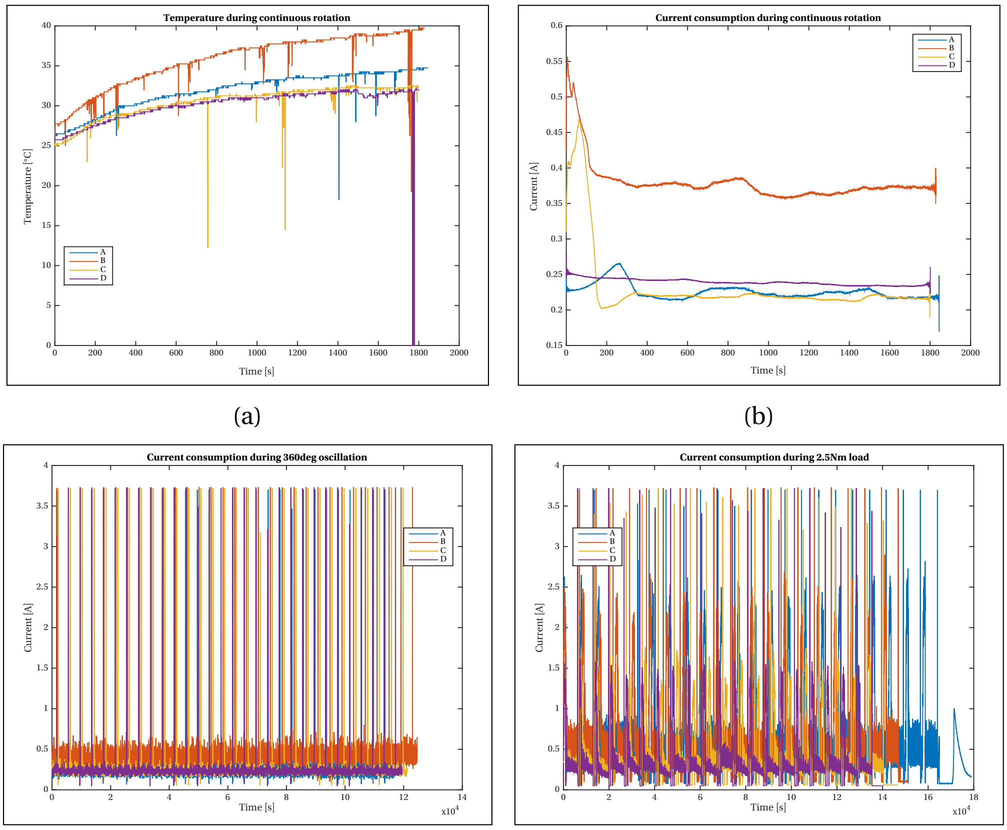
- 70% of the nominal torque (6 Nm) for 5 seconds, 10 repetitions for each direction
- 90% of the nominal torque (7.5 Nm) for 5 seconds, 10 repetitions for each direction
- 100% of the nominal torque (8.5 Nm) for 5 seconds, 10 repetitions for each direction
- 115% of the nominal torque (9.5 Nm) for 5 seconds, 10 repetitions for each direction

We followed the experimental setup for the four different gearbox designs. All of them successfully passed tests 1 and 2. Both Gearbox A and Gearbox B failed load holding test at 95% of the load. For Gearbox A the reason was one tooth of a second-stage planet gear that broke and fell loose inside the gearbox. This caused a locking of the other gears and the damage of one of the gears in the three-stage spur gear set, as seen in Figure 4.8. For Gearbox B one of



(a) (b)

Figure 4.8 – Damage to Gearbox A after failing to hold a payload equivalent to 95% of the nominal torque of the motor: (a) Broken tooth from the second planetary stage. (b) Damage to an $n = 28:12$ gear from the three-stage spur gear set.



(a) (b)

(c) (d)

Figure 4.9 – Comparison of the four gearbox designs for four selected tests: (a) Temperature increase and (b) current consumption during continuous rotation with no output load tests. The spikes are noise in the sensor readings. (c) Current consumption for the ±360 deg oscillation with no load tests. (d) Current consumption for 20 cycles of payload holding (2.5 Nm). From the plot we can infer that Gearbox B has more internal friction.

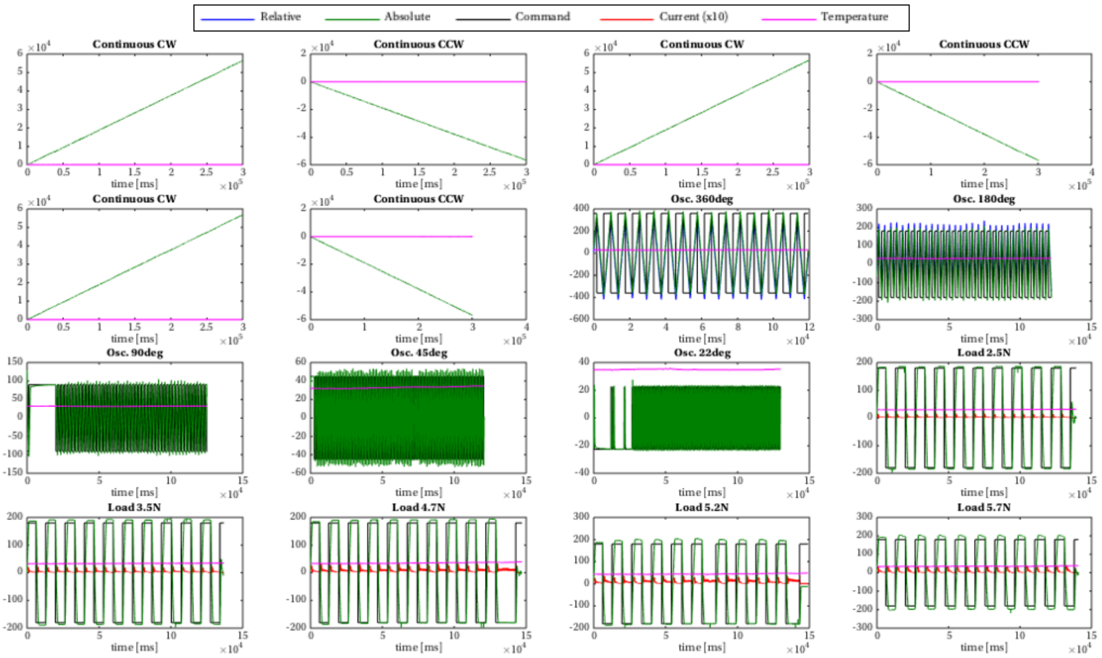


Figure 4.10 – Results from the stress test of Gearbox D: six continuous rotation with no load tests, 5 oscillation with no load tests, and 5 load-holding tests. Units are: [degrees] for relative, absolute, and command positions, [A] for the current, and [°C] for the temperature.

the dowel pins detached as explained in Subsection 4.2.2 with similar consequences for the gearbox. On the other hand, gearbox designs C and D successfully completed all the tests of the experimental protocol.

Figure 4.10 shows an example of the data that we gathered for each gearbox design. From the logged data we can derive useful information such as the backlash value and an estimation of internal friction. In Figure 4.9 we show a comparison between the different designs that shows the predicted high internal friction for Gearbox B.

4.4 Conclusions

In this chapter we described actuator scalability issues and the challenges in the design of a custom actuator-gearbox system. We detailed changes in four design iterations of the Roombots gearbox in which we adapted dimensions, manufacturing and assembly techniques, and materials. We reported experimental protocol and results of a bench test of the four designs. The newest version (Gearbox D) is able to withstand a twofold increase in output load compared to the original design and is now ready for mass production.

In the next chapter we will analyze in detail the mechanical improvements made to another fundamental component of a SRMR: the Active Connection Mechanism.

5 Connection Mechanism

Connection mechanisms are one of the key components of any Self-Reconfigurable Modular Robot. They allow the modules to connect to each other or to connectors placed in the environment.

In Section 5.1 we describe common classification methods. The most common type of connector is based on mechanical forces, followed by those based on magnetic forces. We list some of the desired properties of an ideal connector.

In Section 5.2 we explain the working principles of the ACMv3, the Active Connection Mechanism that was originally developed for Roombots. ACMv3 is a 4-way symmetric, hermaphrodite connector based on mechanical latches.

Although covering most of the desired properties, ACMv3's robustness against misalignments was somehow limited. In Section 5.3 we describe an improved implementation, ACMv4, which has a larger gripping range and a better latch profile. This new connector allowed us to perform single Roombots module locomotion through self-reconfiguration for the first time on a vertical surface.

Even with the ACMv4, the robustness of the on-grid locomotion experiments was still limited. In Section 5.4 we propose a hybrid connector which combines the advantages of the ACMv4 design with the use of magnets in order to improve alignment. This new design allowed us to dramatically increase the number of successful on-grid locomotion trials.

5.1 Introduction

Connection mechanisms are fundamental components for any (Self-) Reconfigurable Modular Robot as they provide attachment points between modules, guaranteeing a strong and reliable physical connection.

Connectors can be active, manual, or passive. *Active Connection Mechanisms* (ACMs) are equipped with the necessary hardware to autonomously establish or unlock a connection between two modules. *Manual connectors* such as ModLock [47] require human intervention to operate. This property differentiates Reconfigurable Modular Robots from SRMRs. On the other hand, *passive connectors* (or passive ports) can only receive a connection from an active or manual connector.

Based on how neighboring mechanisms connect, connectors can be classified as either gendered, hermaphrodite, or genderless. In *gendered* systems [137, 214, 146] there are two distinct variants that mate together. One variant is active while the other is passive and they are usually referred to as male and female. *Hermaphrodite* connectors combine both male and female features on the same connector. Hermaphrodite connectors can mate to each other, with the connection that can be establish by just one or both sides. They can usually also connect to passive ports; this is for example the case in the Roombots ACM. The term *genderless* is often used in the literature as a synonym for hermaphrodite. We prefer however to use it to distinguish systems in which active elements mate in such a way that either side can disconnect without action from the other [152]. Some examples of genderless connectors are SINGO [191], HiGen [152], and the self-soldering connector used in Soldercubes [141]. Hermaphrodite and genderless connectors help simplify the self-reconfiguration planning because all connection points are compatible [201].

Different coupling principles have been proposed in the literature. The vast majority of modular robots uses mechanisms based on *mechanical* coupling, i.e. physical latching [2]. Common means of physical connections are pins and holes [237, 244, 47], latches and chamfered notches [231, 216, 146], sliders [191], screws [178, 151], and grippers [132]. The second most widely-used type of connectors use *magnetic* forces. These forces can be generated by permanent magnets [264, 188, 148], electro-magnets [99], or switchable permanent magnets [66]. Some systems use *magneto-mechanical* connectors, in which permanent magnets are used to make the connection between modules and a mechanical element is needed to separate them [103, 46, 174]. Pneumatic forces are used in [84] and [221]. [86] use hydrodynamic forces to keep modules connected. More recently, a number of novel connection methods that use a phase changing binder material between connecting modules have been proposed: [127] uses Peltier cells to freeze water between floating modules; [228, 24] use hot melt adhesives; [141] describes self-soldering connectors based on Field's Alloy which also automatically establish an electrical connection between modules.

The ideal connection mechanism should have the following desirable properties:

1. **Mechanical strength.** Connectors must function under all loading scenarios, which can include holding long beams made of modules.
2. **Size and weight.** Since a reconfigurable robot can be equipped with multiple connectors, their size and weight should be limited to keep the overall design compact and lightweight.
3. **Tolerance to misalignment.** Perfect alignment of the connecting elements is not always guaranteed; therefore, connectors should be able to withstand a certain degree of misalignment. Passive (using specific face geometries, magnetic or electrostatic forces, or simply chamfered edges) or active guidance (using sensor data in the module's DoF control loop) can help reduce the alignment before the connection process takes place.
4. **Speed of connection and disconnection.** Self-reconfiguration tasks should be performed in a short amount of time and often the bottleneck is the connection/disconnection process. This property is crucial for lattice-type modules that perform flow locomotion[1].
5. **Energy efficiency.** The connector should not require a constant draw of energy to keep the connection locked. Energy should be required only in transition phases.
6. **Symmetry and gender.** Connectors should be able to connect using multiple orientations (e.g. the Roombots ACM is 4-way symmetric, allowing for rotations of 90° in the orientation). With hermaphrodite and genderless connectors, self-reconfiguration planning can be simplified because of a higher number of possible connection ports available. Genderless connectors additionally allow for unilateral disconnection in case one module is unresponsive.
7. **Communication and power sharing.** In SRMRs there is a large amount of information that is shared between modules or with the centralized controller. Transferring data through connectors is more favorable than establishing wireless networks. Sharing power across modules can allow for a better power management.
8. **Production scalability.** As every other component in a modular robot, design, working principle, assembly, and choice in materials should favor a scalable production.

These desirable properties guided the design of Roombots ACMv3 connection, explained in the next section.

---

[1] See Chapter 7 for a definition of flow locomotion.

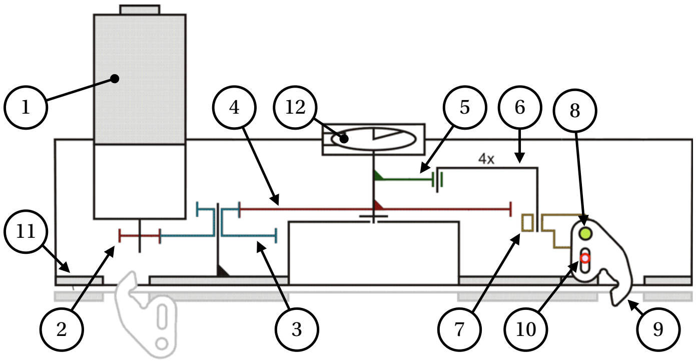


Figure 5.1 – Actuation schematics of an ACMv3 unit. (1) DC brushed motor with 150:1 reduction. (2) Spur gear ($n = 10$). (3) Spur gear ($n = 28 : 12$). (4) Spur gear ($n = 50$). (5) Eccentric disk. (6) Crank drive. (7) Slider. (8) Moving axis. (9) Latch. (10) Fixed axis. (11) Connector board. (12) Position sensor. Part names as in [200]. Target connector greyed out at the bottom. Image adapted from [200].

5.2 Original ACMv3 Design

The ACMv3 connector is a 4-way symmetric, hermaphrodite connector based on physical, retractable latches [200]. It has been designed by Dr. Alexander Spröwitz in order to have a simple, compact, robust, and efficient connector for Roombots. While initially inspired by the AMAS [216] and M-TRAN III [101] coupling mechanisms, the ACMv3 aims at improving the trajectory of the coupling latches. These perform a “scooping” movement that allows them to actively pull closer the target connector, thus improving tolerance to misalignment.

Figure 5.1 shows a schematic view of the actuation of the ACMv3. This is composed of three main parts:

1. **Spur-gear train**. The output shaft of a DC motor with 150:1 reduction (1) is connected to a gear train (2-3-4). The number of teeth n of each spur gear is reported in the caption of Figure 5.1. The total gear ratio is 1750 : 1, calculated as:

   $$\textit{gear ratio} = \frac{150}{1} \cdot \frac{28}{10} \cdot \frac{50}{12} = \frac{1750}{1}$$

   This high reduction is in itself already blocking.

2. **Crank-slider mechanism**. This mechanism converts the rotative motion of the spur gear (4) into linear motion of the slider (7). The system is shown in Figure 5.2. The system is designed in such a way that, after the slider reaches the top dead center (i.e. the position which is farthest from the crankshaft), an additional rotation of the spur gear (4) brings it to a self-blocked state. In this state the ACM can keep the connection locked without requiring power from the motor.

3. **Slider-latch mechanism**. This part converts the linear motion of the slider (7) into a

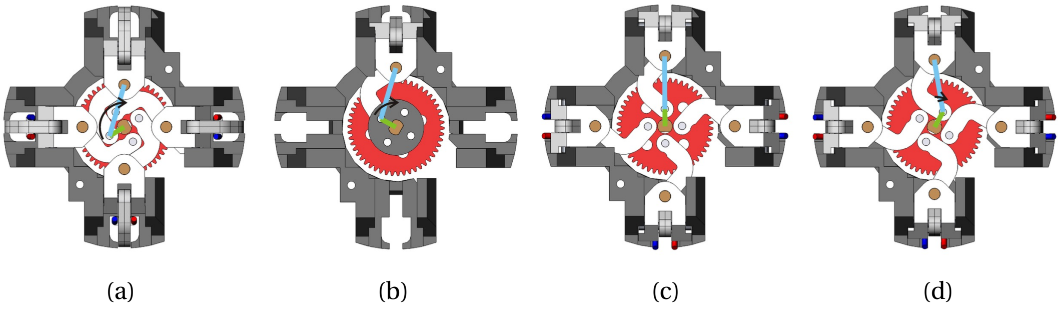

(a) (b) (c) (d)

Figure 5.2 – Crank-slider mechanism (top view) in four states: (a) Fully retracted. (b) Partially retracted. (c) Fully protracted. (d) Self-locked. Image from [200].

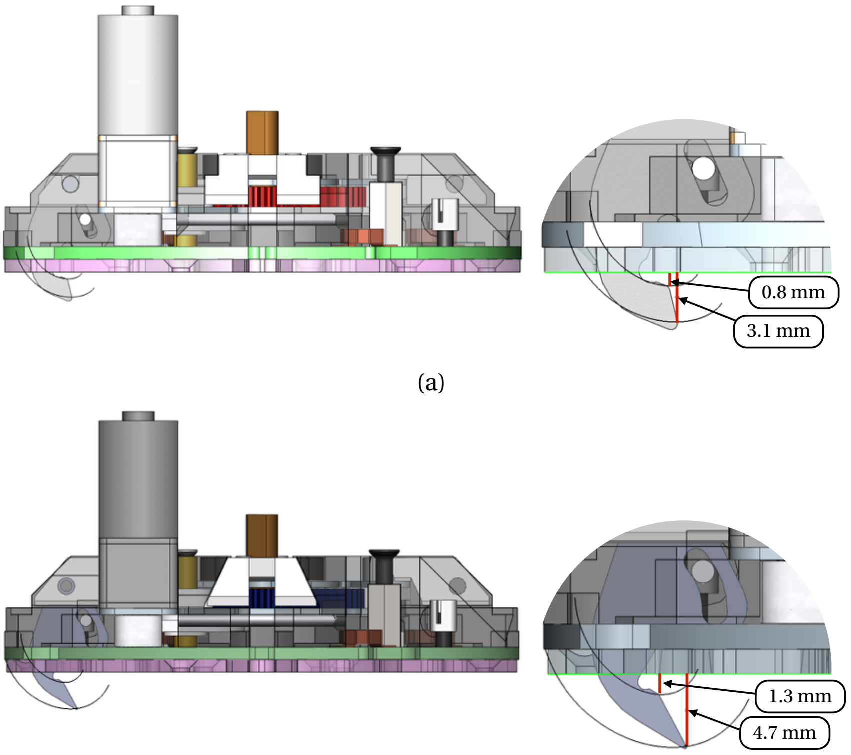
> scooping movement of the tip of the latch (9). The latch motion is constrained by two axes: a moving axis (8) that follows the same trajectory of the slider (7) and hinges the latch, and a fixed axis (10), locked to the main frame of the connector, which guides the latch with a pin-and-slot mechanism.

A potentiometer (12) mounted on the shaft that connects the spur gear (4) to the eccentric disk (5) is used as latch position sensor for the electronic board that controls the ACMv3. The latches create the *active part* of the hermaphrodite connector. They engage the target connector on its *passive part* which is created by 4-way symmetric chamfered slots on the connector board (11). The connection process requires approximately 2 seconds to complete. The same time is also needed for the disconnection.

The ACMv3 is built using a combination of off-the-shelf components, machined glass fiber plates, and ABS 3D-printed parts. The connector weights 57 g, is 65 mm wide, and between 19 mm (top of the electronic board) and 32 mm (top of the DC motor) high. Overall, considering the applications that we envision for Roombots, this design satisfies the desirable design properties 1 (mechanical strength), 2 (size and weight), 4 (speed of connection and disconnection), 5 (energy efficiency), and 6 (symmetry and gender). Property 7 (communication and power sharing) was planned but not implemented in the final product. The connector is relatively cheap and simple to manufacture, but requires some time for the manual assembly. Property 2 (tolerance to misalignment) is satisfied to certain extent. In the next section we will discuss in more detail this property and the corrective actions that we took to improve it.

5.3 ACMv4

Although the ACMv3 has been designed with robustness against misalignments in mind, this property could only be partially exploited during hardware experiments such as self-reconfiguration tasks. A combination of backlash, structural deformations, and calibration inaccuracy often caused the selected ACM to have poor alignment compared to the target ACM, both in terms of offset and angle.



(a)

(b)

Figure 5.3 – Improvement of the scooping movement of the connection mechanism: (a) ACMv3. (b) ACMv4. In green: own's connector plate. In pink: target connector. For clarity, only one of the four latches is shown.

This deviation limited the capabilities of the hardware robot. As we reported in [19], when performing locomotion through self-reconfiguration on a vertical grid *"we could not climb further than position (h) because of elasticity effects in the RB hardware."* Position (h) in [19] corresponds to the R3 movement shown in Figure B.2 (Appendix B).

We considered the ability to perform self-reconfiguration sequences and on-grid locomotion to be of vital importance for the success of the Roombots project. For this reason we redesigned the ACMv3 to improve its misalignment tolerance. Considering that most of the other desired features were already satisfied and that the ACMv3 compact shape was already at the limit of the available volume inside the module, we aimed at a least invasive redesign. This involved an improvement of both the crank-slider and the slider-latch mechanisms. The new design, ACMv4, increases the gripping range and changes the scooping movement trajectory, as shown in Figure 5.3.

The tip of the latch can now enter 4.7 mm inside the target connector (compared to 3.1 mm with the ACMv3) before starting to pull it closer. Therefore, the scooping movement is now more accentuated. More importantly, the "leading edge" of the latch has now a sharper profile that can actively pull the opposite connector.

This redesign allowed us to autonomously complete the two on-grid locomotion sequences for the first time as reported in Appendix B.

5.4 Hybrid ACM

Under standard laboratory conditions, the ACMv4 gripping range was sufficient to successfully perform on-grid locomotion through reconfiguration on a vertical surface. The robustness was however still limited. Experiments had to be restarted several times in order to have flawless runs with no human intervention.

For this reason we created a *hybrid connector* that combines physical latches with permanent magnets (Figure 5.4). We modified the assembly of the ACMv4 to fit a permanent neodymium disc magnet (diameter 15 mm, height 3 mm, maximum strength 3.2 kg) at the center of the connector. In this design the latches are used to align, grip, and maintain the connection. The magnets are instead only used for passive guidance and to attract the target connector. Contrary to magneto-mechanical connectors, the hybrid connector does not have an active way to break the magnetic connection. The magnetic field is however weak enough compared to the Roombots's actuators so that an escaping movement is sufficient to pull the connectors apart.

The hybrid ACM allowed us to dramatically increase the success rate of single Roombots module locomotion through self-reconfiguration trials, making it an almost trivial task. It is easy to understand that the passive alignment properties of the hybrid connector can compensate misalignment only to a certain degree. This is particularly relevant in the case of

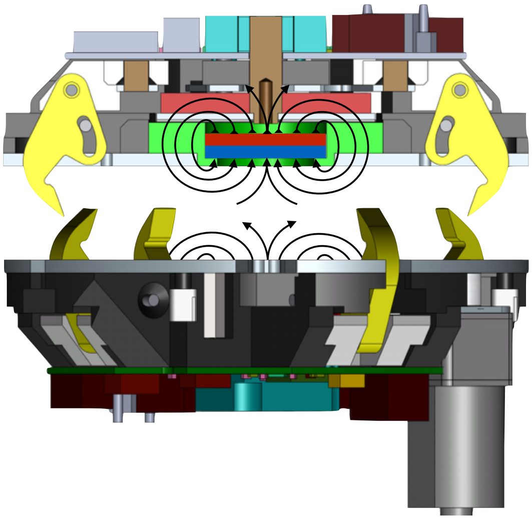


Figure 5.4 – Hybrid ACMs. Top connector in cross sectional view. The central magnet is used for passive guidance only. This version of the ACM is gendered because we used magnets with axial magnetization.

on-grid Roombots metamodule locomotion[2]. In this case the misalignment produced by the long lever, combined with the doubled weight of the structure, limits the effectiveness of the hybrid connector. We believe however that other causes such as backlash in the actuation and structural deflection are responsible for the failure of the on-grid metamodule locomotion.

In this version of the hybrid ACM we use magnets with axial magnetization (i.e. with their north and south pole on the flat circular surfaces). This magnetization produces the strongest magnetic field on the main axis of the disk. However, a connector can only have one of the two polarities pointed towards the outside of the module. This affects its gender property. This first version of the hybrid AMC is therefore not hermaphrodite but gendered. When preparing the setup to perform self-reconfiguration experiments we have to take this into account and distribute “north” and “south” connectors accordingly. This can be avoided by exchanging the axial magnets with freely rotating diametrically magnetized magnets, with a mechanism similar to the one used to connect mockup modules in [200].

As mentioned earlier, the hybrid ACM uses a physical connection, maintained by the four latches. The magnets are only used to improve the alignment with the target connector and are not strong enough to keep two modules connected together. However we thought that the magnet could also be used to snap small external parts to Roombots modules[3]. To be magnetically connected (without using the physical connection), these parts should have a magnet with the opposite polarity on their surface, be light enough, and should also not require a specific orientation. Since the hybrid ACM is not a magneto-mechanical connector, these external parts cannot be automatically disconnected. Human intervention or the use of an additional external device would be needed to pull the external parts apart.

5.5 Conclusions

In this chapter we presented the connection mechanism, one of the most fundamental components of a Self-Reconfigurable Modular Robot. We explained how connectors can be classified and we listed several desired properties that an ideal connection mechanism should have. We described the Roombots ACMv3 connector as it was designed by Dr. Alexander Spröwitz and explained its limits in terms of tolerance to misalignment. We then proposed two improvements to this design. ACMv4 has improved kinematics that create a larger scooping movement of the latches, increasing the gripping range from 3.1 mm to 4.7 mm. This allowed us to autonomously perform locomotion through self-reconfiguration for the first time on a vertical surface, as reported in Appendix B. The Hybrid ACM combines the mechanical coupling system of ACMv4 with small magnets used solely for passive guidance. This improved the robustness of the connection process, allowing us to dramatically increase connection success rate.

Active Connection Mechanisms can be used to change the connectivity of the SRMR, to attach

---

[2] On-grid locomotion is not the main focus of this thesis.

[3] This feature is discussed here but was never used for any of the studies reported herein.

to connector plates fixed in the environment to perform on-grid locomotion, or to connect and manipulate external parts and objects. In the next Chapter we will discuss some examples of external attachments that can be used in conjunction with Roombots modules.

6 External Attachments

Although we designed Roombots to be a homogeneous SRMR, we believe that it could take advantage of its ACMs for attaching to and manipulating different types of external attachments. In this chapter we describe three types of attachments that we thought could be useful for a SRMR and more specifically for our "Roombots as building blocks for furniture" scenario and for improving locomotion.

In Section 6.1 we present passive parts, which are structural elements that can be used in conjunction with active Roombots modules to create structures with increased mechanical stability compared to those built entirely from active modules.

In Section 6.2 we list some possible specialized parts that could be directly connected to Roombots modules to provide them with specific new functionalities. The use of these parts would make Roombots a kind of heterogeneous system in which some complex tasks are executed by dedicated modules. Some of these could include for example specific types of actuators that would be hard to implement in a compact way just with Roombots modules, additional sensors, or devices as displays, lights, or speakers.

Finally, in Section 6.3 we describe the use of compliant passive parts that could be used to improve locomotion performance. Several different types of compliant external attachments could be included in our SRMR. Here we focus on the construction of in-series intermodule passive compliant elements. In Chapter 10 we report locomotion experiments with compliant elements with two Roombots modules and with the 8-DoF Lola-OP™ Modular Snake Robot.

Some parts of this chapter have been adapted from:

[224] M. Vespignani, E. Senft, S. Bonardi, R. Moeckel and A.J. Ijspeert. “An experimental study on the role of compliant elements on the locomotion of the self-reconfigurable modular robots Roombots". In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Nov. 2013, pp. 4308-4313.

My original contribution: Conceptual contribution, Experimental setup, Hardware experiments, Data analysis, Writing.

[226] M. Vespignani, K. Melo, S. Bonardi, and A.J. Ijspeert. “Role of Compliance on the Locomotion of a Reconfigurable Modular Snake Robot". In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sept. 2015.

My original contribution: Conceptual contribution, Software development, Simulation experiments, Hardware experiments, Data analysis, Writing.

[225] M. Vespignani, K. Melo, M. Mutlu, and A.J. Ijspeert. “Compliant snake robot locomotion on horizontal pipes". In *Safety, Security, and Rescue Robotics (SSRR), 2015 IEEE International Symposium on.* Oct. 2015.

My original contribution: Conceptual contribution, Experimental setup, Hardware experiments, Data analysis, Writing.

6.1 Passive parts

The long term vision of the Roombots project is to create Self-Reconfigurable Modular Furniture, as presented in the rendering in Figure 2.4 (Chapter 2). We believe that larger Roombots structures are best built from a mix of active Roombots modules and lightweight passive elements with embedded connection ports. Passive parts can be used to constrain the robotic modules and allow a larger load to be applied to the overall structure, even when the modules are switched off.

Figure 6.1 shows conceptual examples of structures built by active modules and passive parts. The use of passive parts and multiple connection points can help increase mechanical stability. These elements can be small blocks, the size of one or multiple lattice cells, manipulated and transported in the environment by active Roombots modules or metamodules, as we demonstrated in [15].

In the realization of the Self-Reconfigurable Modular Furniture scenario we also considered passive parts created by larger structural parts (such as tabletops or long beams) or entire standard pieces of furniture equipped with passive connectors. In Chapter 11 we will provide some examples of this “augmented furniture”.

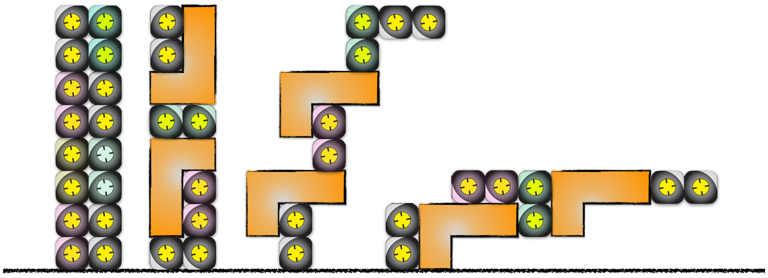


Figure 6.1 – Use of passive parts to increase mechanical stability and reduce the weight of a structure. The leftmost structure is built using 8 Roombots modules. The others combine 4 Roombots modules and 2 passive parts.

6.2 Specialized Parts

For the design of specialized parts we brainstormed a list of possible items that we imagined could be integrated in a Roombots structure. We put a particular focus on specialized parts that could be useful in a domestic environment, such as those that we used in Chapter 11. All these specialized parts should be equipped with Roombots passive ports.

The number and type of devices that can be included as specialized parts depends on the application and type of task (e.g. locomotion, reconfiguration, manipulation). We report here a partial list of objects sorted by their main function.

- Structural support
    - Passive stiff elements
    - Cable systems (e.g. tensegrity)
    - Pre-constructed body parts (e.g. table)
- Useful for locomotion
    - Wheels, Swedish wheels, Whegs
    - Articulated actuated legs
    - Tracks
    - External passive extensions, rubber feet
    - Sticky materials, grippers, anisotropic materials, claws, to create high friction with the ground on a specific place
    - Inertial units, flywheels
- Impact protection
    - Bumpers
    - Soft enclosures
    - Compliant extensions
- Sensors or end effectors
    - Exteroceptive sensors
    - Grippers
    - Tools
    - Lights, speakers, monitors
- Computation and power
    - Computation units
    - Data logging and storage units
    - Extra battery packs
- User's input
    - Control panel
    - Communication board for input devices (e.g. IR remote controllers)
    - Exteroceptive sensors for voice control or for gesture recognition
- Ergonomics
    - Handles

    - Soft structures
    - Inflatable pillows

- Aesthetics
    - External covers

6.3 Compliant Parts

In Chapter 4 we explained how scalability affects homogeneous SRMRs. Actuators have to be designed with the worst possible configuration in mind. One of the challenges in the field of Reconfigurable Modular Robotics is to design modules that can push as much as possible this scalability boundary, allowing modules to lift several times their own weight. For this reason, many robots in literature are designed with large torque capability (e.g. with high reduction ratio gearboxes), resulting in rather slow speeds and less dynamic gaits [182].

One approach that we explored to push this scalability boundary is by using compliance in the structure. Compliance could be included in a number of ways, for instance in series with the actuators or as softness in the body of the robot. In our work we used compliant extensions, a special kind of specialized parts that can be connected to a modular structure. These extensions can be plugged to any ACM available to create compliant links between modules or soft protrusions that contact with the environment. We focused our work on the first type, creating inter-module passive compliant elements.

We explored the use of these elements for off-grid locomotion tasks with (Self-) Reconfigurable Modular Robots, under the hypotheses that added in-series compliance could bring benefits in terms of:

- Improved energy efficiency by using the compliance to store and release elastic energy
- Increased system dynamics
- Better adaptation to rough terrain features and maximizations of the contact points with the ground
- Reduced shock loads on the actuators and on the whole structure

Related Work

Running animals can save energy and reduce unwanted heat production by bouncing along on elastic structures, using the principle of the pogo stick. For example, horses can generate 40% of positive mechanical work for trotting and galloping from the elastic coil of their tendons [11]. They can save even more energy by using return springs to halt the legs at the end of each

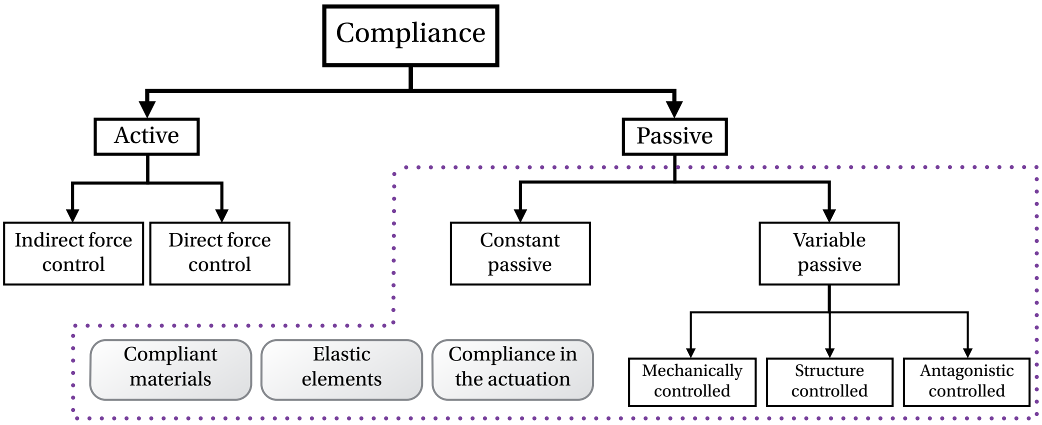


Figure 6.2 – Types of compliance

forward or backward swing. Most mammals have pads on their paws, which help moderating the impact force of the foot on the ground and improve road holding by preventing chatter [3].

In classical robotic applications (e.g. industrial robots), designers have traditionally tried to maximize interface stiffness between actuators and loads and minimize joint compliance [162]. This makes sense in industrial high-precision systems because stiffness improves accuracy, stability, and bandwidth of position control. Stiffer interfaces allow increasing the bandwidth of a position control feedback loop without compromising stability. However, mechanically compliant systems have advantages as well, including:

- Greater shock tolerance and reduced damage in case of accidental collision
- Lower reflected inertia
- More accurate and stable force control
- Potential for energy storage and restitution

It is therefore not surprising that, especially for application such as human-robot interaction or dynamical legged robotics, several robots with compliant actuators, flexible links, and/or compliant joints have been designed. Figure 6.2 gives a brief overview of how compliance or compliant behavior can be included in a robot.

One way to introduce compliance in a robotic structure is by adding passive elastic elements. This can be done, for instance, by constructing components or entire links using flexible materials [94, 167]. These elements bend under the effect of an external force, converting it into elastic potential energy that will then be partially fed back to the system. Smart Composite Microstructures (SCM) technology [239] allows creating a layered heterogeneous structure with local variations in materials properties which give high specific strength and stiffness in selected areas, while providing compliance in others. This offers the possibility to include non-linear properties such as bio-inspired compliance gradients.

Passive compliance can also be lumped in elastic elements such as springs. These can be created in many shapes (e.g. coil springs, leaf springs, cantilever springs), but there are three basic types of springs:

- Compression springs, designed to operate with compression loads
- Extension springs, attached at both ends, designed to operate with tension loads
- Torsion spring, designed to operate with torques

An alternative to (steel) springs can be the use of air springs, or the use of elastomers [126] as compliant medium.

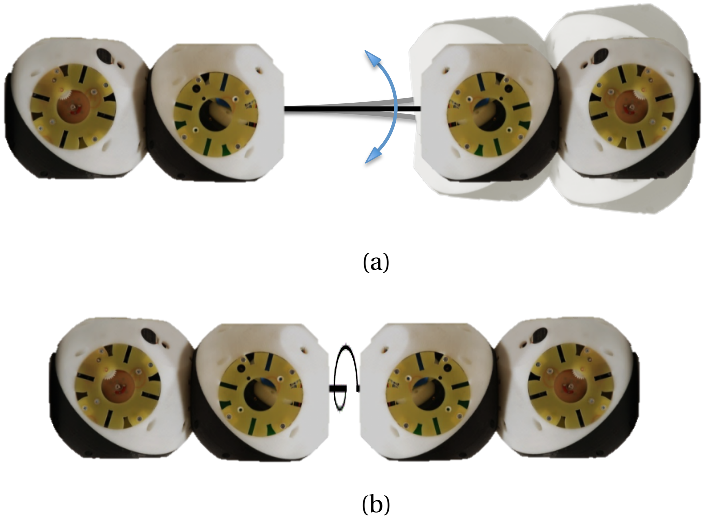
Passive elastic elements can be added in series to the actuation. Many applications use electric motors with gear reduction; while this can provide power at low speed, it also increases reflected inertia, backlash, torque ripple and noise. Introducing series elasticity reduces interface stiffness and can help reduce many of the previous problems. Among the many advantages, series elasticity:

- Provides low-pass filtering of shock loads
- Turns force-control problems into position-control problems
- Provides for the possibility of energy storage

One notable implementation are the Series Elastic Actuators (SEA) developed by Pratt and Williamson [161]. In alternative, when possible, using pneumatic or hydraulic actuators can give already a certain degree of compliance and impact robustness [175, 190]. Soft Pneumatic Actuators (SPA) are compact and lightweight actuators fabricated almost entirely out of highly compliant elastomers such as silicone rubber [55].

Passive elastic elements are relatively easy to be added to a robotic structure; they provide high bandwidth and great protection against impacts. However, their main limitation is that the spring constant k is fixed once for all.

To overcome this limitation, it is possible to employ adjustable stiffness elements. If applicable, one simple way to change the behavior of a spring is by changing its pre-compression. Alternatively, in case of cantilever springs, it is possible to reduce the length of the beam to increase the stiffness (e.g. with a rigid slider [59, 60]). Passive, controllable stiffness actuators can be implemented using for example antagonistic-controlled stiffness (using non-linear springs), Variable Stiffness Actuators (VSA) or Variable Stiffness Joints (VSJ). A good review on compliant actuator design is presented in [73]. Tunable stiffness can also be achieved with


(a)

(b)

Figure 6.3 – Types of in-series compliance explored in this work: (a) “Omnidirectional” compliance. (b) Torsional compliance.

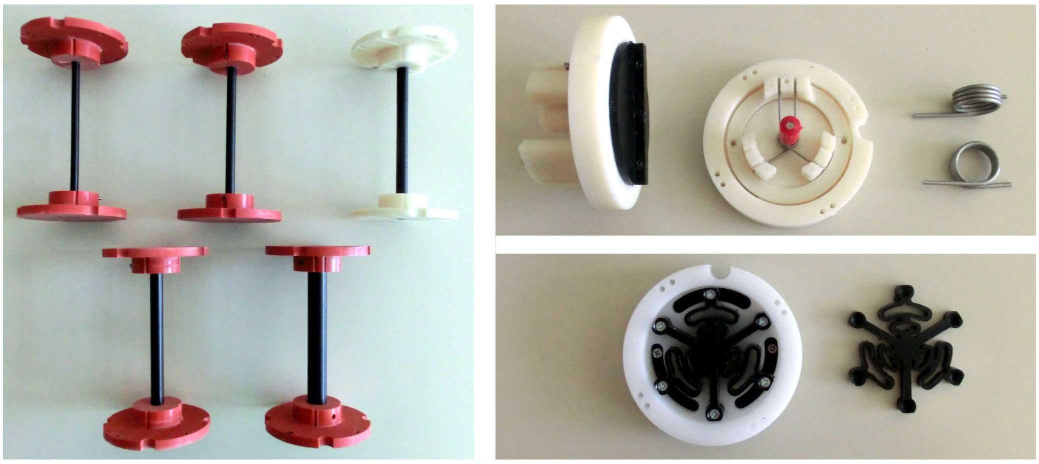
structures based on electroactive polymers (EAP) that are lightweight and simple and can change their stiffness and damping properties very quickly [45]. Shape Memory Alloy (SMA) wires can be coated with thin layers of Shape Memory Polymer (SMP) and sewn onto pieces of fabric to create a variable stiffness fabric for wearable devices [32].

Finally, compliance can also be included as active compliance achieved through the robot’s control loop. Common approaches are Compliance Control, Impedance Control, Force Control and Hybrid Force/Motion Control [197]. The main drawback of active compliance is that it is not an intrinsic property of the robotic structure, but it is a response of the control to an external perturbation. Active compliance depends on the sampling rate of the system and therefore the response cannot be instantaneous. This means that it has a limited response rate and the mechanical structure could be damaged from peaks of impact force.

6.3.1 Compliant parts for Roombots modules

In order to study the effect of compliance in a reconfigurable modular structure we analyzed the locomotion on flat ground of two Roombots modules with in-series compliant elements. Details on the work and results are reported in Section 10.1. Here we describe the implementation and manufacturing of the elements.

We focused on two different types of compliance, namely “omnidirectional” and torsional compliance (Figure 6.3). The basic element of the first set, referred as compliant rod (*CR*), is a cylindrical beam made of polyoxymethylene (POM), blocked on each side (Figure 6.4.a). By changing the diameter d of the beam, we were able to produce different stiffness values. The length of the beam is fixed at $\ell = 0.081$ m so that the total length of the element (including



(a) Compliant Rods (CR)

(b) Torsional Springs (TS)

Figure 6.4 – Compliant elements used for the experiments: (a) Five compliant rods made in POM. (b) Top: torsional spring construction for TS_1–TS_4, using off-the-shelf springs; bottom: torsional spring for TS_5, with a custom designed POM spring. Image from [224].

the fixation on each side) is equal to the basic Roombots grid size (0.11 m). Given the axial symmetry, the *CR* can bend in every direction perpendicular to its longer axis. In addition, *CR* elements have also some torsional stiffness (torsion of a beam).

To recreate pure torsional compliance, we developed two different designs of torsional springs (*TS*). The first design (mechanism type 1, used for TS_1–TS_4) was made using pairs of off-the-shelf torsional springs, slightly pre-compressed, mounted in a mirrored configuration so that they act one against each other (top row of Figure 6.4.b). This way, the resulting compliant element has the same stiffness value when rotated clockwise or counterclockwise. The range of motion of this design is ±60 deg. The second *TS* design (mechanism type 2, used to get a higher stiffness value for TS_5) was inspired by the work of Carpino et al. [30] (bottom row of Figure 6.4.b). This design was chosen because it is symmetrical along the direction of rotation, the stiffness value can be varied by changing the thickness, and it is quite compact and easy to manufacture. TS_5 was made in POM and has a range of motion of ±30 deg.

For each type of compliance we manufactured five elements with different compliance. Stiffness values of these elements are reported in Table 6.1. These values where chosen among a larger selection so that, during the locomotion of the Roombots structure, the most compliant elements (CR_1 and TS_1) naturally bent almost to their full range. Each of the next three elements (CR_2–CR_4 and TS_2–TS_4) roughly doubles the value of the previous one. The stiffest elements (CR_5 and TS_5) are of an order of magnitude higher than the previous, to provide an “almost stiff” test case. For the *CR*, the stiffness values were calculated using beam theory. If we consider the material as homogeneous, we define two types of stiffness:

$$\text{Flexural stiffness: } k_F = \frac{EI}{\ell} = \frac{E\pi d^4}{64\ell} \tag{6.1}$$

Table 6.1 – Compliant elements specifications. $k_{CR,F}$: flexural stiffness (from Equation 6.1); $k_{CR,T}$: beam's torsional stiffness (from Equation 6.2); k_{TS}: torsional spring stiffness (from datasheet (TS_1–TS_4)) or FEA (TS_5)).

Item name	d [mm]	Flexural stiffness [Nm/rad]	Torsional stiffness [Nm/rad]
Compliant Rod 1	5	$k_{CR_1,F} = 1.06$	$k_{CR_1,T} = 0.78$
Compliant Rod 2	6	$k_{CR_2,F} = 2.19$	$k_{CR_2,T} = 1.63$
Compliant Rod 3	7	$k_{CR_3,F} = 4.07$	$k_{CR_3,T} = 3.02$
Compliant Rod 4	8	$k_{CR_4,F} = 6.95$	$k_{CR_4,T} = 5.15$
Compliant Rod 5	12	$k_{CR_5,F} = 35.2$	$k_{CR_5,T} = 26.1$

Item name	Torsional stiffness [Nm/rad]	Mechanism type
Torsional Spring 1	$k_{TS_1} = 0.44$	1
Torsional Spring 2	$k_{TS_2} = 0.58$	1
Torsional Spring 3	$k_{TS_3} = 1.18$	1
Torsional Spring 4	$k_{TS_4} = 1.63$	1
Torsional Spring 5	$k_{TS_5} = 16.6$	2

$$\text{Torsional stiffness: } k_T = \frac{GJ}{\ell} = \frac{G\pi d^4}{32\ell} \tag{6.2}$$

where E is the Young's modulus for the material (we used 2.8 GPa for POM), G is the shear modulus calculated as $\frac{E}{2(1+\nu)}$ (Poisson's coefficient $\nu = 0.35$ for POM), and I and J are respectively the area moment of inertia and the polar moment of inertia.

For the first *TS* design, it can be demonstrated that the total stiffness is equal to the sum of the two springs. Since we used off-the-shelf components, their stiffness value was provided by the manufacturer. For the second *TS* design, the stiffness value was estimated using finite elements analysis (FEA). It is worth pointing out that we are mostly interested in the order of magnitude and the relative stiffness value between different compliant elements rather than their precise value.

We imagined these passive elements as extensions that can be easily connected to Roombots modules using Active Connection Mechanisms. However, in order to simplify the long experimental work presented in Section 10.1, we manually and tightly fixed them to available sockets on the Roombots shells using screws.

6.3.2 Adding Compliance to the Lola-OP™ Snake Robot

We decided to extend our study on passive compliant extensions using a Modular Snake Robot, a special case of Reconfigurable Modular Robot. This decision was mainly driven by the readily availability of the Bioloid platform [12], which allows to build modular robots built entirely using off-the-shelf components and controlled using standard Dynamixel libraries. The system has a commercial-grade robustness which makes it perfect for running extensive hardware experiments. Its modularity and reconfigurability allows to quickly construct different configurations.

We assembled Bioloid modules in the shape of a snake robot, using the Lola-OP™ configuration described in Section 2.3. In Sections 10.2 and 10.3 we will analyze the effect of added in-series compliance in the interconnection between each module of an 8-DoF Lola-OP™. In the following we we give details on the manufacturing of the compliant elements.

Compliance has been added in the form of cylindrical beams, fixed on each side, able to bend in any tangential direction and twist on their longitudinal axis. These compliant elements are equivalent to the Compliant Rods for Roombots modules presented in the previous section. The stiffness level of these elements depends on their length, ℓ, their diameter, d, and the mechanical properties of the materials used for construction.

We manufactured elements with 4 different levels of compliance from relatively soft to stiff. For the first three elements (denoted as C1, C2, and C3) the compliant elements are made from super-elastic Nitinol wires with diameters of $d = 1.0$, 1.5, and 2.0 mm. These elements

Table 6.2 – Compliant elements specifications.

Item	d [mm]		Flexural stiffness [Nm/rad]	Torsional stiffness [Nm/rad]
C1	1	(Nitinol)	$k_{C1,F} = 0.46$	$k_{C1,T} = 0.35$
C2	1.5	(Nitinol)	$k_{C2,F} = 2.3$	$k_{C2,T} = 1.75$
C3	2	(Nitinol)	$k_{C3,F} = 7.3$	$k_{C3,T} = 5.54$
C4	8	(POM)	$k_{C4,F} = 70.4$	$k_{C4,T} = 52.3$

have a total length of 25 mm, but are mechanically fixed on each side to a supporting structure of thickness 8.5 mm, as shown in Figure 6.5. Thus, their effective length is $\ell = 8$ mm. The fourth element (denoted as C4) is made using the exact same principle, but with a 8mm-wide cylindrical rod made in POM (polyoxymethylene).

Each of these elements can bend in any direction tangential to their longitudinal axis, and also twist about it. We calculated flexural and torsional stiffnesses using Equations 6.1 and 6.2. For the Young's modulus E we used 75 GPa for Nitinol and 2.8 GPa for POM. For the Poisson's coefficient, needed to calculate the shear modulus, we used $\nu = 0.33$ for Nitinol and 0.35 for POM. The resulting values of stiffness are reported in Table 6.2.

We chose $C1$ as a lower bound, as it corresponds to a soft element that highly deforms under

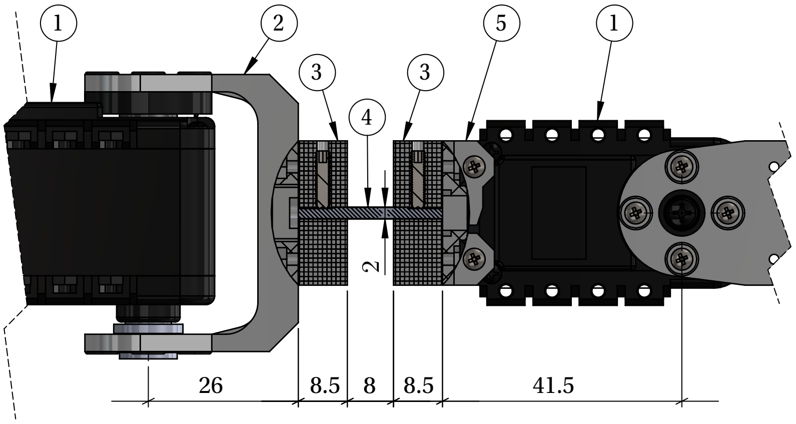


Figure 6.5 – CAD model of a segment of the Lola-OP™ robot, with a broken-out view of a C3-type compliant element: (1) Dynamixel AX-12A® actuators. (2) Bioloid Flange F2. (3) Machined POM parts. (4) Super-elastic Nitinol wire (for C1, C2, and C3) or POM rod (for C4). (5) Bioloid Flange F3. Dimensions are expressed in mm. Image from [225].

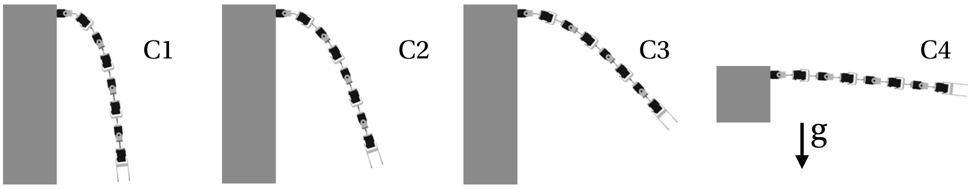


Figure 6.6 – Simulation of compliant 8-DoF Lola-OP™ robots under the effect of gravity. The first module is fixed to a frame. All servos are locked at zero position to show only the bending of the compliant elements. Image adapted from [225].

the nominal forces applied by the motors. *C*2 and *C*3 have higher levels of stiffness, but are still partially deformed during runtime. *C*4 is chosen as upper bound and has a rigid behavior.

To get an intuition of the chosen range of flexural values of stiffness, in Figure 6.6 we show the effect of gravity on simulated cantilevered 8-DoF Lola-OP™ robots.

6.4 Conclusions

In this chapter we proposed the use of external attachments in conjunction with active modules. Passive parts create mechanical stability while reducing weight and price compared to structures built completely from active modules. Specialized parts can create new features, for example by adding new end effectors to the modular structure, or improve existing ones, as in the example of the Swedish wheel part that could replace the use of roombots modules as wheels. In Chapter 11 we show some examples of how these specialized parts can be used in a household to improve quality of life. A third type of external attachments, compliant parts, can be useful for locomotion to store and release elastic energy, to loosen internal constraints, or to reduce shock loads on the structure or its actuators. Compliant parts have been used in several studies reported in Chapter 10.

This concludes our contribution to the hardware challenges for SRMRs. In the next part we will tackle locomotion challenges by learning how to coordinate large numbers of degrees of freedom after self-reconfiguration and improving locomotion performance through the use of compliance.

Locomotion Challenges

Part III

7 Overview of the Roombots locomotion control

Locomotion is one of the basic tasks of a Self-Reconfigurable Modular Robot. We are particularly interested in the off-grid locomotion (which will also be referred to as gait locomotion), in which the different degrees of freedom move together in a coordinated manner to produce a translation of the center of mass of the structure along a desired direction.

In Section 7.1 we describe the challenges in designing a locomotion controller for a (Self-) Reconfigurable Modular Robot and discuss common methods used in literature.

In Section 7.2 we present the locomotion controller that we used in our experiments which uses a network of coupled non-linear oscillators representing a bio-inspired Central Pattern Generator (CPG).

Some parts of this chapter have been adapted from:

[224] M. Vespignani, E. Senft, S. Bonardi, R. Moeckel and A.J. Ijspeert. “An experimental study on the role of compliant elements on the locomotion of the self-reconfigurable modular robots Roombots". In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Nov. 2013, pp. 4308-4313.

My original contribution: Conceptual contribution, Experimental setup, Hardware experiments, Data analysis, Writing.

[130] R. Moeckel, Y.N. Perov, A.T. Nguyen, M. Vespignani, S. Bonardi, S. Pouya, A. Spröwitz, J. van den Kieboom, F. Wilhelm, and A.J. Ijspeert. “Gait optimization for Roombots modular robots - Matching simulation and reality". In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Nov. 2013, pp. 3265–3272.

My original contribution: Conceptual contribution, Experimental setup.

7.1 Locomotion Controllers for Self-Reconfigurable Modular Robots

Locomotion is an essential skill of any autonomous robot and one of the basic tasks of a (Self-) Reconfigurable Modular Robot. Lattice-type systems move by dynamically changing the shape of the structure, generating the so-called *flow locomotion*. Modules move individually by rearranging their connectivity on a substrate that can be made of static grid environment or other modules. Although flow locomotion can be modeled as a Self-Reconfiguration Problem (SRP) and solved adapting methods from this domain [199], the resulting motion is extremely slow and inefficient [2]. Moreover flow locomotion is highly affected by hardware imperfections and limitations: alignment problems or unwanted structural deflection can cause two neighboring modules to not be able to successfully dock, thus blocking the self-reconfiguration process.

In our research we are interested in the locomotion of chain- and hybrid-type robots. These systems can use the more energy-efficient *gait motion*, in which off-grid coordinated actuation of the joints results in a translation of the center of mass along a desired direction. In gait motion no attachment/detachment actions take place and the morphology of the modular robot remains fixed during locomotion[1]. Reconfiguration occurs only in order to adapt to a new environment or situation.

Despite the large number of developed gait controllers for bipedal, multi-legged, crawler, caterpillar, and snake robots, their direct adaptation to modular robots is not straightforward [2]. "Conventional" robots are usually designed (e.g. in terms of morphology and number, type, and placement of the DoF) to suit a particular locomotion gait; on the other hand modular robots are normally designed to be general purpose, they are expected to exhibit multi-modal locomotion, and their shape and distribution of degrees of freedom is highly dependent on the number and type of modules used in the structure. Hence, achieving locomotion in modular robots usually imposes more challenges than in conventional robots.

7.1.1 Related Work

In order to realize a particular gait, it is necessary to define two elements: a *control method* and a *synchronization method*. The control method defines the cyclic set of actions that each module has to perform at each time-step. Synchronization methods are needed to coordinate the motion of modules so that the overall structure creates smooth gait motion.

One of the first and simplest control methods for reconfigurable robots was *gait control tables*, in which modules have a list of predesigned mappings from a set of states to a set of actions [243, 211]. The control usually runs in open-loop, with the assumption that all modules are synchronized. The main weaknesses of this method are that it is not scalable as the number of modules in the structure increases, it cannot generate non-periodic gaits, and it is not adaptable to environmental changes.

---

[1] Unless explicitly stated, for the rest of the thesis we will use the term *locomotion* as a synonym for gait motion.

Recently Christensen et al. [36] presented a *distributed reinforcement learning* strategy for morphology-independent lifelong gait learning for modular robots. All modules run identical controllers that optimize their action selection based on the robot's velocity as a global, shared reward signal. This method can be used to learn simple control tables and can recover the gait in case of self-reconfiguration or module failure.

Zhang et al. introduced *Phase Automata* [261], a way of programming scalable locomotion gaits using an event-driven state automaton with an initial phase delay. Phase automata can produce periodic and non-periodic gaits, can create terrain-adaptive gaits, and it is scalable. It was used to control the locomotion of hardware structures composed of more than 50 PolyBot modules. However, transition rules have to be manually designed.

Found in many vertebrates and invertebrates, *Central Pattern Generators* (CPGs) are neural circuits capable of producing coordinated patterns of high-dimensional rhythmic output signals while receiving only simple, low-dimensional, input signals, even in isolation from motor and sensory feedback [81]. CPGs can be modeled as distributed systems of coupled non-linear oscillators, usually with one oscillator per degree of freedom [116]. CPGs can produce both periodic and non-periodic gaits, are robust against perturbations, and can produce coordinated patterns in absence of feedback. Their parameter optimization however does not scale well as the number of degrees of freedom in the modular robot increases [158].

Neuro-Evolution of Augmenting Topologies (NEAT) is a Genetic Algorithm that evolves at the same time weighting parameters and structure of an Artificial Neural Network [206]. *HyperNEAT*, an evolution of NEAT, has been used to create efficient gaits for modular robots [71, 43]. *Reinforcement Learning* (RL) has been used to solve the "Control Your Own Body" (CYOB) problem on simulated YaMoR and Roombots modules [44, 230]. D'Angelo recently showed that the RL PoWER algorithm is much faster in achieving high performance compared to HyperNEAT [43]. The quick convergence of RL does not come at the cost of solution quality.

SRMRs are a perfect platform to use co-evolutionary algorithms in which morphology and controller are evolved simultaneously to suit a particular task [115]. In nature there is never an "empty" organism, but brain and body co-evolve [155]. Pfeifer and Bongard [154] emphasized the advantages that can be achieved by exploiting morphology in parallel with control aspects. The co-evolution of body and mind was first investigated by Sims in his seminal paper [198], and later used by several other authors in simulation [34, 116, 72, 159, 54] and directly on hardware robots [24].

As discussed earlier, in order to realize a particular gait it is necessary to also define a synchronization method. There are two main categories: *blocking* and *non-blocking*.

Blocking methods keep modules synchronized at all time, but can result in discrete movements with pauses between each action. Examples include the *Master Control*, used by Yim [247], and *Hormone-Based* methods such as those developed by Shen et al. [192, 195]. Moreno and Gomez [133] developed a hybrid system that uses hormone messages to propagate sensory feedback information, in combination with a CPG for motion generation.

On the other hand, non-blocking methods avoid interruptions in the module movements caused by delays in the synchronization messages. In [210] joint angles are calculated with a cyclic function with period T; after a time interval d (delay), each parent module sends a synchronization message to its child modules to ensure that their action is delayed by d. The simplest form of continuous synchronization would be to have one clock signal shared between all modules; however in many applications it is assumed that all the modules' clocks stay synchronized for the duration of the experiments [247, 208].

Unless otherwise stated, for the locomotion studies presented in the next chapters we used bio-inspired control methods based on **Central Pattern Generators** combined with learning techniques. The chosen design of the CPG controller is described in the next section. The CPG control parameters are found using **Particle Swarm Optimization** (PSO), as explained in Appendix E. For the experiments with hardware robots in Chapters 9 and 10, a "start experiment" signal is broadcasted via Bluetooth to all the modules. Therefore we assume that the clocks of the different modules are sufficiently synchronized and use a non-blocking control.

7.2 Central Pattern Generators (CPGs)

We control Roombots structures through Central Pattern Generators [81], networks of coupled nonlinear oscillators. CPGs were preferred to other control methods for the following reasons:

1. They are capable of generating synchronized movement patterns using only a few control parameters (as in Loeb's marionette puppets analogy [110]). Thus only a few parameters have to be learned during the optimization process to achieve complex behavior.
2. In principle, CPGs provide a scalable control scheme that can be easily adapted to the number of modules forming the robotic structure by simply adding or removing oscillators from the network[2].
3. CPGs ensure smooth transitions of motor set points after a modification or perturbation of the state variables. This avoids abrupt changes of the motor states which is typically preferred to achieve stable gaits and expand motor and robot life time.
4. They can be easily expanded to generate different types of locomotion patterns. In this work we only considered sinusoidal waves.
5. State variables can be influenced by sensory feedback signals. This feature however was not exploited in the work presented in this thesis.

---

[2] In practice, as discussed later in this section and in Section 7.3, parameters optimization becomes challenging as the number of DoF increases.

Similar to bio-inspired lamprey models, each CPG is modeled using a *phase oscillator*, as in [82]:

$$\dot{\phi}_i = 2\pi \cdot f + \sum_j w_{ij} \cdot r_j \cdot \sin\left(\phi_j - \phi_i - \psi_{ij}\right) \tag{7.1}$$

$$\dot{r}_i = a_i \left(R_i - r_i\right) \tag{7.2}$$

$$\theta_i = r_i \cdot \sin\left(\phi_i\right) + X_i \tag{7.3}$$

where i and j are the indexes of the oscillators, θ_i is the oscillator output controlling the position set-point of the degree of freedom i, and r_i and ϕ_i are state variables that encode amplitude and phase of the oscillation. The parameters w_{ij} and ψ_{ij} are respectively the coupling weight and phase bias of the coupling between oscillators i and j. a_i is a positive constant which determines the rise time of the amplitude r_i to the desired value R_i. We use one common frequency for all oscillators ($f = 0.2Hz$), which fits well with the nominal speed of the hardware actuators.

Each oscillator i has the following open parameters that are subject to optimization:

- The desired amplitude R_i
- The offset X_i
- The phase lags ψ_{ij} with respect to the other oscillators

While each oscillator has only one amplitude and offset, it is clear that the number of parameters for the phase lag ψ_{ij} can increase exponentially with the total number of degrees of freedom. In order to limit this number, in our experiments we use symmetric bi-directional couplings such that ψ_{ij} = -ψ_{ji}. Additionally, we only consider nearest-neighbor coupling, i.e. only phase oscillators of neighboring DoF are coupled together, as shown in Figure 7.1 for two structures composed of three Roombots modules. All coupling weights are set to $w_{ij} = 2$, as in [158].

With these predefined rules, any arbitrary Roombots structure composed of m modules and $n = 3m$ degrees of freedom has the following open parameters: n amplitudes, n offsets, $2m$ phase lags ψ_{ij} between oscillators belonging to the same modules, and 1 phase lag value for each physical connection between modules. This means $9m - 1$ parameters if the structure does not have closed loops.

CPG's equations were modeled and integrated over time using the Cȯdɣn Framework developed by Jesse Van den Kieboom [39, 93], whom we thank for his valuable support. In order to find optimal values for the open parameters we used Particle Swarm Optimization, as described in the next section.

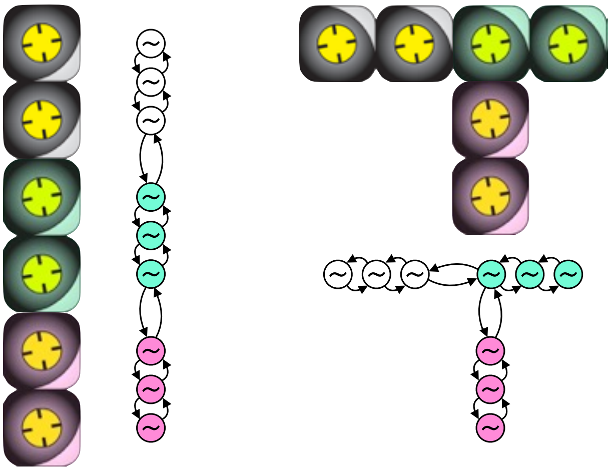


Figure 7.1 – Examples of Roombots CPG topologies with one oscillator per degree of freedom and nearest-neighbor coupling. Each structure has 26 parameters subject to optimization: 9 amplitudes, 9 offsets, and 8 phase lags.

7.3 Conclusions

In this chapter we described and analyzed existing control methods for (Self-) Reconfigurable Modular Robots. For our locomotion experiments we generate joint angle positions using Central Pattern Generators (CPGs), bio-inspired networks of coupled nonlinear oscillators.

As discussed in Section 7.2, despite using some simplification rules to design the CPG topology such as nearest-neighbor coupling, the locomotion controller for a Roombots structure composed of m modules has at least $9m-1$ open parameters that require optimization. This number does not scale well with the number of modules in the structure. As the search space of the optimization becomes larger, more particles and more iterations (i.e. more optimization time) are needed to explore it thoroughly. In the next chapter we will describe two bio-inspired methods that we developed in order to reduce the number of parameters that need optimization.

8 Body/Limb and Symmetry Finder Algorithms

Designing a locomotion controller for a Self-Reconfigurable Modular Robot is challenging because the morphology and connectivity of the robot can change arbitrarily at any time during the completion of a task. In this chapter we consider the case of a robotic structure self-reconfiguring into a “never seen before” morphology, meaning that the robot does not have any prior knowledge on how to locomote with this structure (i.e. it cannot use a gait table). As discussed in Section 8.1, one typical example would be a SRMR that faces a hardware failure of some modules during a time critical mission and is forced to reconfigure into a new shape and re-optimize its locomotion controller. Given the time criticalness of the mission, the robot cannot wait until full convergence of the optimization algorithm, but has to accept a “good enough” quick solution.

In Section 8.2 we propose a new method (BLF) to speed up the learning process by automatically extracting morphological features in order to reduce the number of parameters to be optimized. The proposed method is based on the detection of bio-inspired sub-structures, i.e. body, limbs, and articulation joints, inside any arbitrary modular structure to create reduced network topologies. Additionally, information on structural symmetries (SYM) is used to further reduce the open parameters.

We test combinations of the proposed methods on three simulated structures (Section 8.3), showing in Section 8.4 that they can lead to significantly better results until at least the 30^{th} iteration of the optimization process, with fitness values on average three times higher than the complete unreduced optimizations.

Some parts of this chapter have been adapted from:

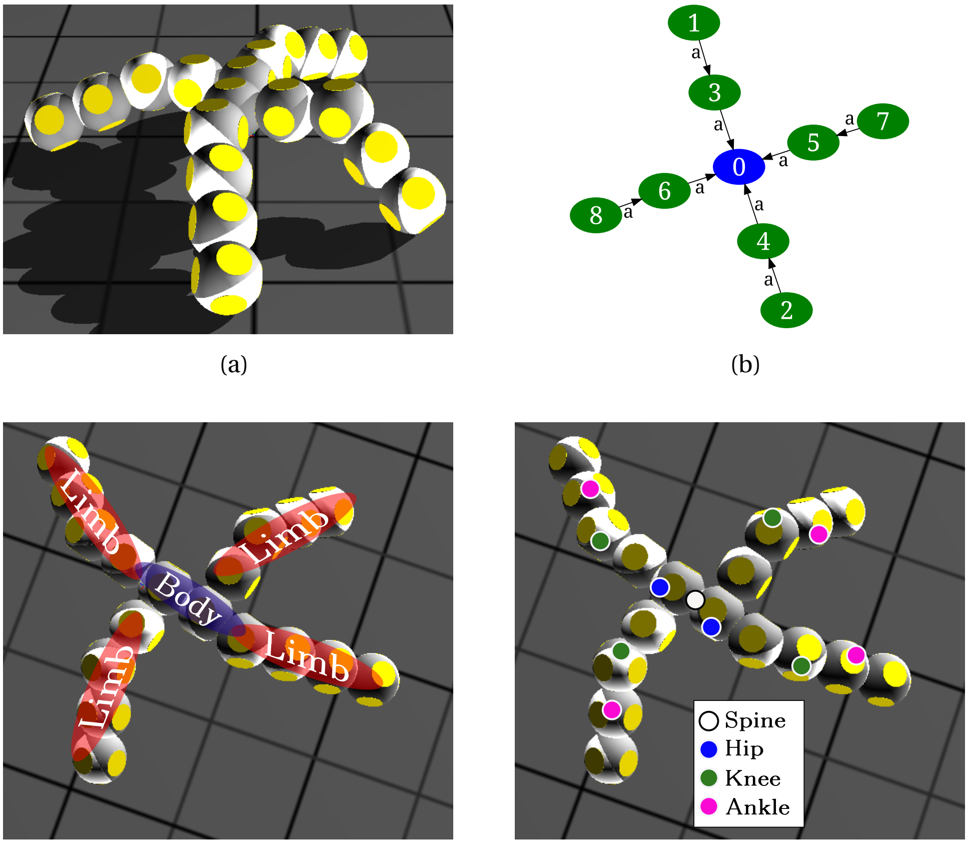
[17] S. Bonardi, M. Vespignani, R. Moeckel, J. Van den Kieboom, S. Pouya, A. Sproewitz, and A.J. Ijspeert. “Automatic generation of reduced CPG control networks for locomotion of arbitrary modular robot structures". In: Proceedings of Robotics: Science and Systems. 2014.

My original contribution: Conceptual contribution, Simulation experiments.

8.1 Introduction

As already discussed in Chapter 2, one of the potential advantages of SRMRs is their ability to autonomously modify their structure to adapt to changes in their surroundings. These morphologic changes bring the burden of having to also adapt the locomotion controller to each new situation. If the new structure was known beforehand, it is possible to use previously learned parameter sets, e.g. using *gait tables*. However, if the robot reconfigures into a new structure that was never seen before, it is necessary to employ some type of locomotion learning algorithm.

A possible way to find optimal sets of control parameters is to use model-free optimization methods such as Powell's Method [160] or Particle Swarm Optimization [90, 157]. These methods have been used for example to find optimal CPG parameter sets for several Roombots structures [158]. Their main downside is the time required to sufficiently explore the parameter space, which is tightly linked to the number of parameters that needs to be optimized.



(a) (b)

(c) (d)

Figure 8.1 – Example of application of the BLF algorithm: (a) A quadruped structure composed of 9 Roombots modules. (b) Corresponding graph representation. (c) Detected body and limbs. (d) Articulations detected using the set of rules described herein. Image from [17].

In this work we propose an automated method to generate reduced control networks for the locomotion of arbitrary structures made of modular robots[1]. This method can speed up the learning of new parameters and it is particularly efficient in those situations in which there is limited time available for the optimization process. Using predefined sets of rules we reduce the number of active degrees of freedom and therefore of optimization parameters. Our approach is based on the decomposition of the robotic structure into morphologically relevant sub-structures (e.g. body and limbs) and on the automated identification of bio-inspired articulation joints. These joints correspond to the actuated DoF, while all the other actuators are kept in a locked state. Additionally, we also considered existing symmetries in the structure to further reduce the number of open parameters.

Our work is driven by the following hypotheses:

1. The use of bio-inspired functional patterns and symmetries to generate a reduced CPG network can help for fast re-optimization after structural changes due to hardware failure or changes in the environment. The advantages in terms of forward speed and collisions, compared to a fully open optimization, are higher during the first few iterations of the optimization process.
2. The quality of the solution (e.g. the locomotion speed after convergence) is not significantly modified in comparison to a fully open optimization.

8.1.1 Related Work

Modular robots offer the advantage of morphology that can change depending on external factors (e.g. changes in the environment) or internal ones (e.g. sudden hardware failure). This flexibility brings an additional challenge in comparison with monolithic robots in terms of design of efficient controllers. Moreover the increase in the number of degrees of freedom with each module added to the structure makes it difficult to hand-design specific gaits. Monolithic robots can also have to cope with a change in their morphology due to hardware issues, requiring as a consequence a re-design of their locomotion controller. The optimization of the set of parameters to generate efficient locomotion is often time consuming.

Since the early work by Yim [243] on the caterpillar locomotion of Polypod robots, several approaches have been proposed for the control of the locomotion of structures made of modular robots. For example, Shen et al. [193] proposed a hormone based method to control the locomotion of CONRO robots, Stoy et al. [212] used role-based control and cellular automata, and Yu et al. [256] described a consensus based approach for the locomotion control of 2D modular robots. CPGs, implemented as systems of coupled oscillators, are well suited for distributed locomotion control and various techniques have been investigated [81,

---

[1] Based on the proposed implementation, the only limitation is that structures cannot be fully linear (e.g. a snake robot or a Roombots metamodule) nor fully cyclic (e.g. in a closed chain). Dedicated rules could however be introduced to cover also these situations.

115, 116, 42, 202]. The main drawbacks with those approaches is that they consider a fixed morphology and require the manual design of the CPG network, which might prove to be a tedious task for large structures. Some authors [198, 21, 107] used evolutionary methods and co-evolution to make the robot discover its own morphology, or used genetic algorithms to evolve possible gaits for given structures [87]. Those methods are often computationally demanding and time consuming, making them difficult to transfer on-board and on-line. More recently, accelerated learning methods have been investigated [37, 38, 36] based on a distributed and morphology independent learning process. The main difference with our approach is that we propose to optimize beforehand the control network itself instead of approximating the learning reward for the different possible actions. Christensen et al. [35] described a control framework to generate full body behaviour based on the decomposition of the structure into bio-inspired parts (like muscle or bones) with pre-defined function (e.g. muscles can contract). The control is then done at the level of those sub-parts, abstracting away their individual components. Although this approach is similar in essence to our method, the main difference is that we propose an automatic detection of bio-inspired joints and symmetries in any arbitrary structure instead of considering predefined structures built from known sub-parts.

8.2 Materials and Methods

For this work we used a simulated model of Roombots developed in the mobile robot simulator Webots [229]. For the locomotion controller we used a CPG formulation similar to the one described in Chapter 7. In order to find the most efficient gait for each structure, we use Particle Swarm Optimization (PSO) to generate (off-line) the set of CPG control input parameters.

In this work we used CPG networks to control the locomotion of the different structures. Our approach can however be generalized to other control strategies (e.g. based on simple sine waves).

8.2.1 Body/Limb Finder

In many vertebrates, body (torso) and limbs are clearly separated and have different roles in the generation of coordinated movement for locomotion. We developed a bio-inspired algorithm, called *Body/Limb Finder* (BLF), that can automatically identify body and limbs sub-parts in arbitrary modular structures.

Figure 8.1 shows an example of application of the Body/Limb Finder algorithm of the *quad9* structure, composed of 9 modules. The robotic structure is first represented as an undirected graph in which each node represents a module and each vertex represents a connection between two modules. The main idea of the BLF algorithm is that the removal of the body from a given structure will lead to several disconnected elements that represent the limbs. For more complex structures, the body could be further decomposed into a linear part and/or a

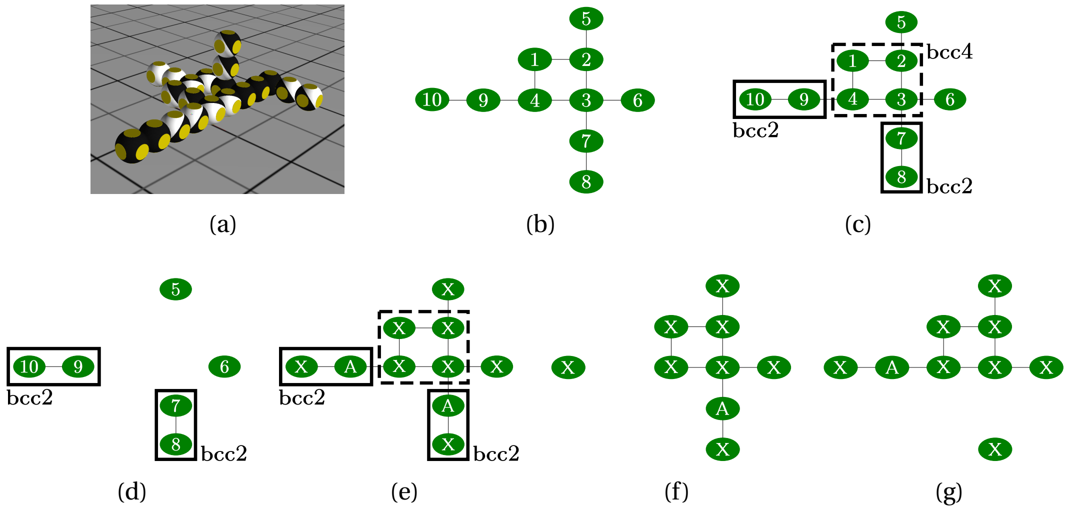


Figure 8.2 – Step by step application of the body-limb finder algorithm on *struct10* (a), a structure with 10 Roombots modules used for our experimental validation. (b) The structure is converted into an indirected graph in which each node represents a module and each edge represents a physical connection. (c) Detection of bi-connected components (bcc) in the graph and sorting depending on the number of nodes they contain. (d) Cycles found in the previous step (i.e. bcc containing more than 2 nodes) are tested to find body parts (bcc4 is one). (e) Detection of articulation nodes (marked with an A) for the remaining bcc. (f, g) Articulation joints are tested to check whether they are still part of the body (if the removal of each articulation joint leads to more than two sub-graphs). The two bcc2 are not part of the body. Image from [17].

cyclic part. A cyclic part is defined as a closed loop of connected modules.

We set specific rules for the actuation strategy (e.g. amplitude of oscillation) of each sub-part. In this work we defined some rules to differentiate between spine, hip, knee, and ankle joints.

8.2.2 BLF Rulebook

Body/Limbs Differentiation

The BLF algorithm is primarily based on the notion of *bi-connected component* (bcc). In a graph, a bi-connected component is one such that, if removed, leads to a disconnected graph. Figure 8.2 shows the three main steps of the BLF algorithm:

1. **Decomposition into bi-connected components**. Starting from the graph representation of a structure (b), we identify all the bi-connected nodes (c). From this classification we can already identify the linear parts in the structure (i.e. bcc composed or 2 of less nodes) and the cycles (i.e. bcc composed of more than 2 nodes), if any.

Table 8.1 – Boundaries for the amplitude parameter depending on the type of articulation considered.

	Articulation joint:			
	Spine	Hip	Knee	Ankle
Minimum value	0	0	0	0
Maximum value	$\frac{2\pi}{3}$	$\frac{\pi}{2}$	$\frac{\pi}{6}$	$\frac{\pi}{6}$

2. **Identification of the cyclic parts of the body**. Cyclic parts of the body correspond to groups of modules that are fully linked together, meaning that at least two paths exist between any pair of the group. For each cyclic part, we check the connectivity to detect if they are part of the body (d). After removal of the cyclic part, if the remaining graph is disconnected then the cycle is part of the body.

3. **Identification of the linear parts of the body**. For this step (e) we consider the bcc2 nodes (bi-connected components with two nodes) found at step 1. We select only nodes that satisfy the following two rules:

   i **Clustering power**: if the removal of the node leads to a number of components for the remaining graph strictly greater than 2 then the node is a linear part of the body.

   ii **Articulation**: the node must be an articulation of the graph.

   After that, we calculate the shortest path between the selected nodes and we include it in the linear part of the body (minus the intersection with the nodes found at step 2).

The limbs are the disconnected components remaining after the removal of the previously found body.

Articulation Rules

The following are empirical rules that we defined for the selection of bio-inspired articulation joints.

- **Spine**: The spine joints are defined with the following:
  - Spine joints can only be defined in linear parts of the body.
  - Every joint inside the linear part of the body is part of the spine, except for hip joints.
  - Only the middlemost joint is actuated. The other ones are locked.

  - In case of multiple options, the tie is solved at random.

  We chose to consider the cyclic parts of the body as non-actuated.

- **Hip**: There is only one hip per limb, and it must satisfy the following rules:
  - A hip is a joint at the frontier between a limb and the body. It must therefore have at least one neighboring joint being part of the body.
  - A hip joint can be physically placed either in a limb node or in a body node.
  - In case of multiple options, the most proximal joint is chosen as hip.
- **Knee**: There can be only one knee per limb (or none), satisfying the following:
  - A knee joint must be part of a limb.
  - A knee joint is at the center of the limb, at equal distance between the hip and the most distal joint.
  - The limb segment between hip and most distal joint must have at least two joints in order to classify one of them as knee. Otherwise no knee joint can be defined for the limb.
  - In case of multiple options, the more proximal joint is chosen as knee.
- **Ankle**: The ankle joint follows exactly the same rules defined for the knee, but with the knee joint in place of the hip joint.

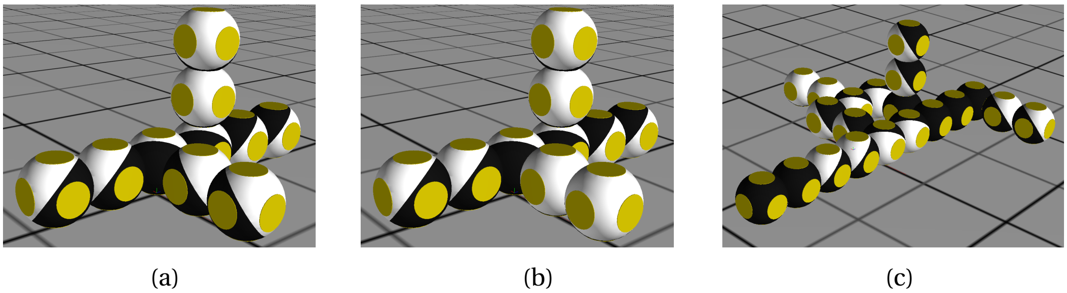
The unclassified degrees of freedom are considered as locked. Figure 8.1 shows an example of joint classification on a structure made of 9 Roombots modules.

Generation of Reduced CPG Networks

After the detection of bio-inspired sub-parts and joints, we let the algorithm generate the reduced CPG network. We assign one oscillator for each detected articulation joint. All the other degrees of freedom are locked. The parameter boundaries for the amplitude of each oscillator depend on the type of articulation. Boundaries used in this work are reported in Table 8.1.

Couplings between oscillators are inspired by the typical bone structure in many vertebrates and are defined by the following rules:

- The spine oscillators are fully coupled together.
- The hip oscillators are fully coupled together. They are further coupled to the closest spine oscillator in the structure.
- Each knee oscillator is coupled to the hip oscillator of the same limb.



(a) (b) (c)

Figure 8.3 – Test structures used to validate the algorithm: (a) *quad5-sym*, a quadrupedal structure composed of 5 Roombots modules with symmetric limbs. (b) *quad5-unsym*, similar to *quad5-sym*, but one of the limbs is not symmetric since it is connected to the body with a different orientation. (c) *struct10*, a pseudo random structure of 10 modules with no symmetric limbs. Image adapted from [17].

- Each ankle oscillator is coupled to the knee oscillator (if it exists) or the hip oscillator of the same limb.

In [17] we demonstrate that for any structure the number of parameters in the reduced CPG network is strictly less than in a “fully open” network (i.e. the general CPG network applied to the original topology).

8.2.3 Distance-based Symmetry

In addition to the BLF algorithm, we also tested a parameter reduction method based on geometrical symmetries between the limbs. The underlying idea is that similar limbs can share the same optimized trajectory (same amplitude and offset for the different oscillators). They will differ in phase shift, which is the parameter that can define different gait patterns.

We defined as “similar” the limbs in which the distribution and orientation of both the degrees of freedom and the mechanical connections between modules and with the body are identical. The complete mathematical description of our distance-based Symmetry Finder algorithm is provided in [17].

8.3 Experimental Results

We evaluated our method on three simulated structures made of Roombots modules, shown in Figure 8.3. The first, named *quad5-sym*, is a quadruped-like structure made of 5 modules in which all four limbs are symmetrical. The second strucutre, called *quad5-unsym*, is similar to the first one, but with a limb connected to the body with a different orientation. In this case, only 3 limbs are symmetric. The last structure, *struct10*, is a pseudo random asymmetric structure made of 10 Roombots modules. The first two structures were chosen to represent bio-inspired structures with symmetric and partially symmetric limbs. *Struct10* was selected to

Table 8.2 – Number of optimized network parameters for the three case structures in the four different conditions. Numbers in brackets correspond to previously defined networks; these cases are optimized only once.

		Structure:		
		quad5-sym	quad5-unsym	struct10
Condition:	FO	44	44	90
	BLF	21	21	26
	BLF-SYM	15	17	(26)
	SYM	26	30	(90)

test our method on a larger structure in which no intuitive gait could be engineered. Moreover the simulation of this larger structure required a longer computation time; this is exactly the type of situation in which we would benefit from a quick learning of new locomotion parameters in just a few optimization iterations.

In order to evaluate the effectiveness of the BLF algorithm and of the Symmetry Finder method, we compared results for four conditions:

1. Fully Open optimization (**FO**): all the parameters of the network are considered open. We used one oscillator per DoF and nearest-neighbor coupling. For each oscillator the amplitude is only constrained to $[0, \pi]$.
2. Body/Limb Finder reduced network (**BLF**): we use the BLF technique to generate a reduced network for the structure. The amplitude parameter is constrained depending on the type of articulation, as in Table 8.1.
3. BLF network and Symmetry Finder (**BLF-SYM**): we apply the Symmetry Finder method to the reduced network generated by the BLF to further decrease the number of parameters to optimize. The amplitude parameter is constrained depending on the type of articulation considered.
4. Symmetry Finder (**SYM**): we applied distance-based symmetries between the limbs to reduce the number of parameters in the Fully Open network. Contrary to the BLF-SYM, this method uses the BLF algorithm just to detect body and limbs, without detecting bio-inspired articulation joints.

Table 8.2 summarizes the number of parameters to be optimized for each structure in the different cases. The parameters used for the PSO optimization for each case are listed in

Table 8.3 – Particle Swarm Optimization settings for the different structures.

Parameters	**quad5-sym**	**quad5-unsym**	**struct10**
Number of particles	80	80	160
Number of iterations		800	
Trial time (t_{total})		30 s	
Maximum velocity		0.6	
Social factor		2.05	
Cognitive factor		2.05	
Constriction factor		0.729	

Table 8.3. The CPG networks for the three structures generated using the four conditions listed above are shown in Table 8.4.

We ran the PSO optimization twenty times with different initial random populations for the three structures *quad5-sym*, *quad5-unsym*, and *struct10*. For the latter, only the FO and BLF networks were tested, since this structure has no geometrical symmetries.

The fitness function f chosen for the optimization process takes into account the displacement of the structure and penalizes collisions between modules:

$$f = \frac{d}{t_{total}} \times c \tag{8.1}$$

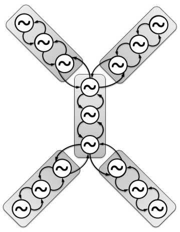
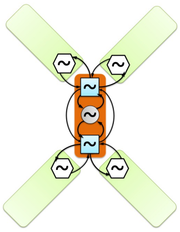
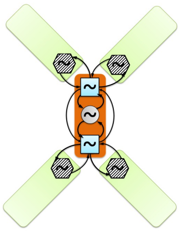
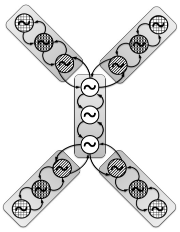
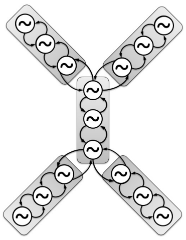
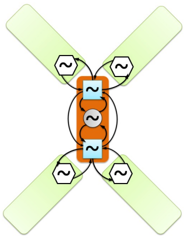
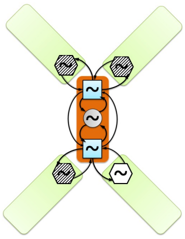
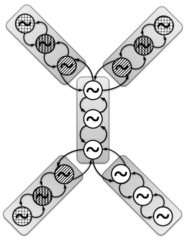
where d corresponds to the displacement of the robot during the total experiment time t_{total} and c is a penalization factor used in case of self-collision equal to 0.001 if there is a collision and 1 otherwise.

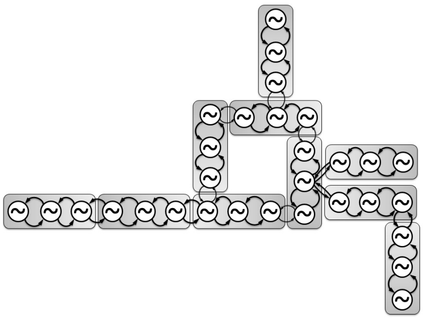
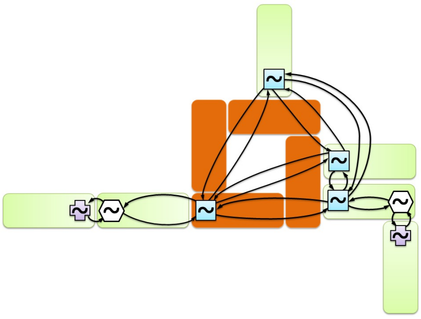
Figure 8.4 shows the results of these twenty optimization runs for each structure. Each point in the plots is the average between the best particle at that specific iteration for each of the twenty optimizations.

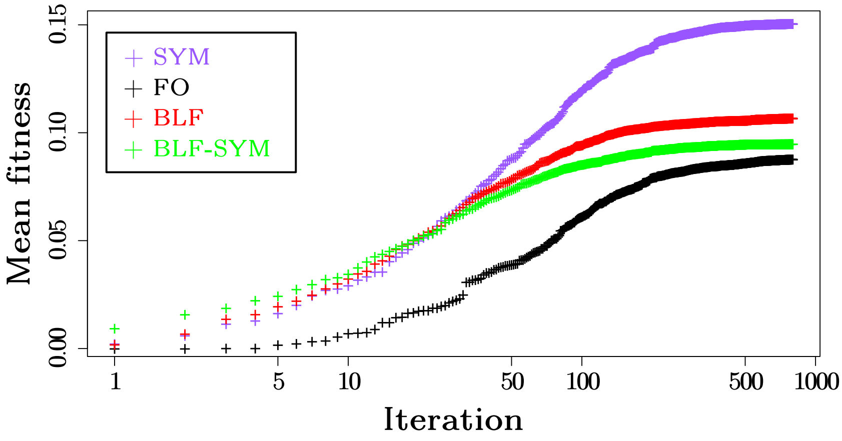
8.4 Discussion

We based this work on two hypotheses. The first one regards the possibility to quickly learn new locomotion parameters in the first few optimization iterations. We hypothesized that

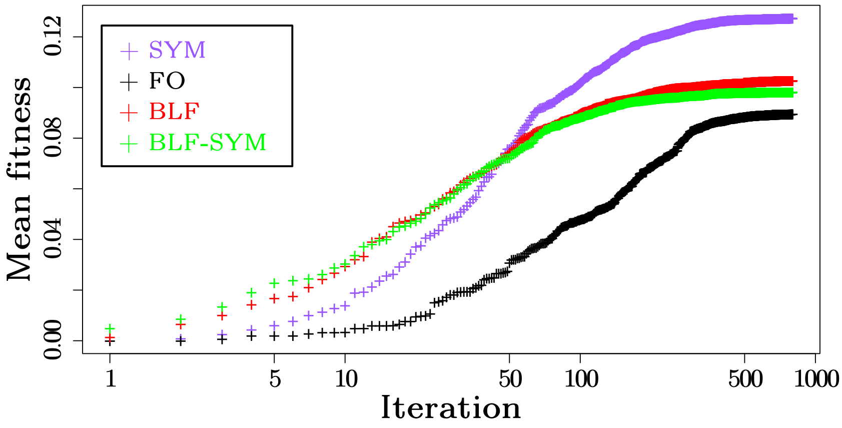
Table 8.4 – CPG networks for the three case structures and different conditions. In the FO case, circles represent generic oscillators. For the BLF and BLF-SYM cases, the limbs are represented in green, the body in orange, and the shape coding is as follows: spine oscillator are circles, hip oscillators are squares, knee oscillators are hexagons, and ankle oscillators are crosses. For the BLF-SYM and SYM cases, symmetric oscillators have similar stripe pattern.

		Condition:			
		FO	BLF	BLF-SYM	SYM
Structure:	quad5-sym				
	quad5-unsym				

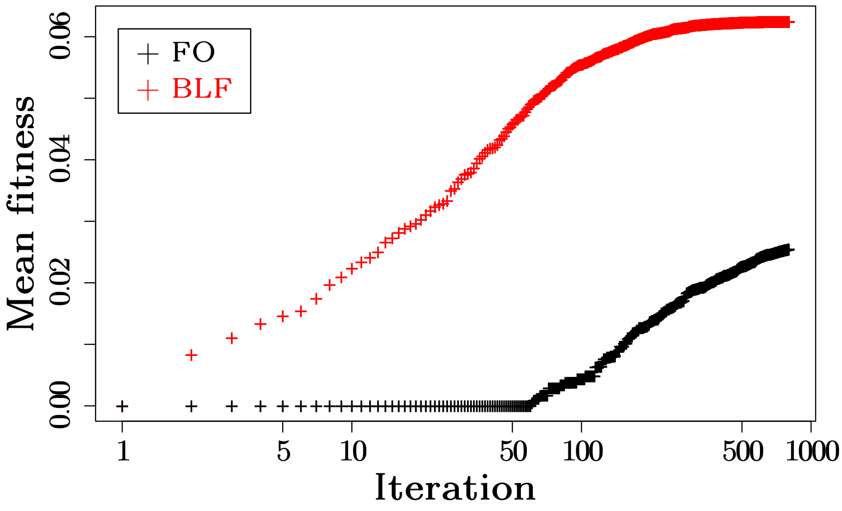
		Condition:	
		FO	BLF
Structure:	struct10		



(a) quad5-sym



(b) quad5-unsym



(c) struct10

Figure 8.4 – The mean value of the fitness function over twenty optimizations for *quad5-sym* (a), *quad5-unsym* (b), and *struct10* (c). Results are displayed in semi-log scale. Images adapted from [17].

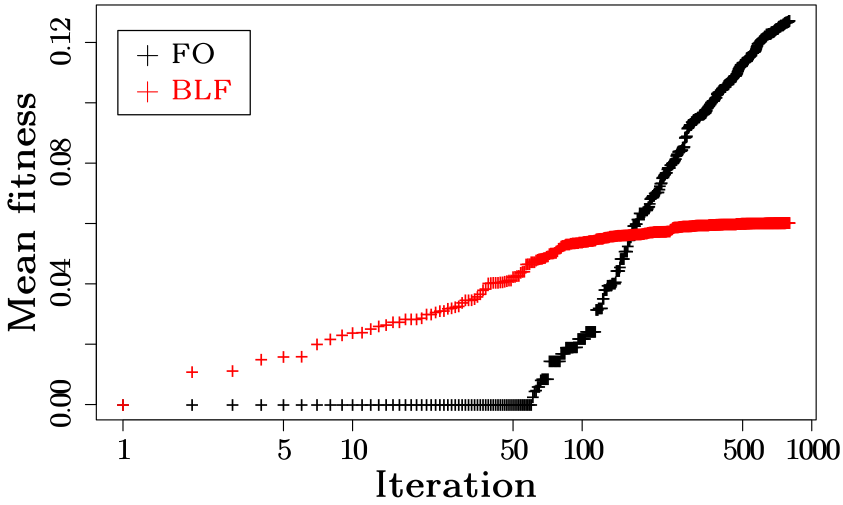


Figure 8.5 – In black, mean value of the fitness function for the four best FO optimization runs of the *struct10* structure. In red, mean fitness value of four randomly chosen optimization solutions in the BLF case. Image adapted from [17].

using a reduced CPG network generated using bio-inspired rules would significantly reduce the number of iterations needed to obtain an acceptable solution. ANOVA tests ran on the results (detailed information available in [17]) show that the reduced CPG network dominated the fully open population at least until the 30^{th} iteration for the *quad5-sym* and *quad5-unsym* structures.

For the *struct10* structure, restricting the search space by introducing automatically generated prior knowledge and boundaries to the parameters created a clear advantage compared to the FO network. Fitness values are significantly better up to the 200^{th} iterations and the convergence is significantly faster. This was mostly caused by the difficulty to find self-collision free solutions in the large search space of the Fully Open network. Out of twenty runs of the *struct10* FO cases, only in four cases we converged to solutions that did not induce self-collisions. Our first hypothesis holds even if we consider only these four solutions with no self-collisions. Figure 8.5 shows these four FO best solutions compared to four randomly chosen BLF solutions. The results from the reduced network are significantly better at the beginning of the optimization process (until iteration 71) before being dominated by the FO solutions.

The second hypothesis concerned the quality of solutions at convergence. We can see from our results that in all cases no significant differences can be found between our three proposed methods and the standard FO case.

One typical test situation for our method would be some hardware failure of a SRMR during a time critical mission: the robot is forced to reconfigure into a new shape and to re-learn how to move. It can, for example, connect to a cloud service to ask for new possible gaits but it cannot wait until the full convergence (meaning hours of computation for large structures).

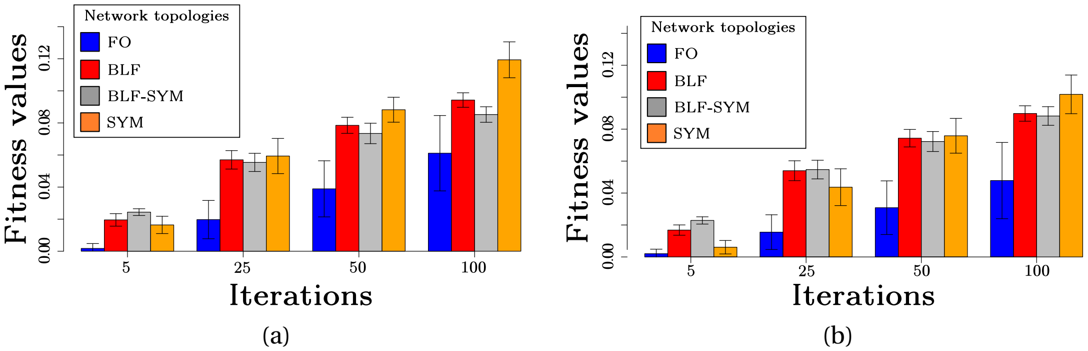


Figure 8.6 – The mean fitness values and standard deviation at iteration 5, 25, 50, and 100 for the different network topologies applied to *quad5-sym* (a) and *quad5-unsym* (b). Images from [17].

Our approach could be used to characterize the new configuration of the robot and propose corresponding reduced CPG networks to speed-up the optimization of the gait.

With our proposed techniques, after only five iterations (around one minute of optimization on average on our computer cluster) we manage to provide gaits for both *quad5-sym* and *quad5-unsym* with a fitness on average 3 times higher than the FO case, as shown in Figure 8.6. Similar trends are can be observed at iterations 25, 50 and 100 (5, 10 and 20 minutes of computation, respectively).

We can also observe on Figure 8.4 that the reduced networks generated using the distance-based Symmetry Finder method (SYM) obtain better results relatively to the two other reduced networks (BLF and BLF-SYM). This can be explained by the fact that the amplitude for the oscillators has larger boundaries than in the two other cases. A qualitative analysis of the resulting gait showed that in the SYM case, as well as on the FO case, the structure tends to rely much on almost rolling movement of some joints to increase its momentum. On the contrary, in the BLF and BLF-SYM cases, the structure tends to have a smaller amplitude of oscillation and favor animal-like displacement of the limb, making the obtained gaits more hardware friendly.

8.5 Conclusions

In this chapter we presented two parameter-reduction methods for Self-Reconfigurable Modular Robots that allow for quick locomotion-parameter optimization in the first minutes after a change in morphology.

The Body/Limb Finder (BLF) algorithm detects bio-inspired sub-structures and articulation joints in any arbitrary modular structure and applies parameter reduction using a set of pre-defined rules that restricts the number of active degrees of freedom.

The Symmetry Finder (SYM) method identifies geometrical symmetries, allowing sub-parts of the structure to share the same optimized parameters.

Both methods and their combination (BLF-SYM) outperformed the “fully open” optimization at least until the 30^{th} iteration of the optimization process (approximately 5 minutes of optimization time on our computer cluster). This makes them ideal for learning “good enough” gaits with just a few optimization iterations for a SRMR engaged in a time-critical mission.

The results presented in this chapter have been obtained using simulation experiments. When transferring optimized locomotion-control parameters to the hardware robot, there could be a mismatch in the expected performance, known as reality gap. To ensure portability of the solutions, in the next chapter we present a hybrid optimization method that allows the simulator to produce results that match well hardware experiments.

9 Hybrid and Meta-Optimization

As the number of degrees of freedom in a Self-Reconfigurable Modular Robot increases, locomotion parameter optimization algorithms require more iterations, and therefore more time, to properly explore the larger search space. In Section 9.1 we describe benefits and drawbacks of online and offline optimization techniques. Online optimization leads to reliable results but is extremely time consuming and could damage the hardware robot. On the other hand, offline optimization can be highly parallelized but results can suffer from the reality gap (i.e. a mismatch between the behavior in simulation and in hardware).

In Section 9.2 we present a hybrid optimization technique that combines advantages of offline and online optimization by providing simulation results that are well matched with hardware experiments. The process is a cycle composed of three steps: (1) an offline optimization to find efficient locomotion controller parameters; (2) a hardware validation of a selected set of solutions; and, (3) a meta-optimization process which improves simulator and model parameters to reduce the reality gap. The hybrid optimization cycle can be repeated multiple times until a good matching is achieved.

In Section 9.3 we show the results of one hybrid optimization cycle applied to the problem of learning locomotion for a Roombots metamodule. Similarity ratio between simulation and hardware experiments is increased by 65%, leading to a visually almost perfect matching.

We conclude in Section 9.4 with an analysis of the benefits of hybrid optimization compared to offline and online optimization. We estimated at least a 2.5-fold reduction in the required optimization time compared to online optimization, while generating results that can be almost seamlessly transferred to the hardware robot.

Some parts of this chapter have been adapted from:

[130] R. Moeckel, Y.N. Perov, A.T. Nguyen, M. Vespignani, S. Bonardi, S. Pouya, A. Spröwitz, J. van den Kieboom, F. Wilhelm, and A.J. Ijspeert. “Gait optimization for Roombots modular robots - Matching simulation and reality". In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Nov. 2013, pp. 3265–3272.

My original contribution: Conceptual contribution, Experimental setup.

9.1 Introduction

In Chapters 7 and 8 we already discussed the challenges in finding suitable controller parameters to generate efficient locomotion in a Self-Reconfigurable Modular Robot. As the number of robotic modules and degrees of freedom increases, the (automatic or manual) design of efficient locomotion gaits becomes onerous and time consuming. Typically, optimization techniques can be applied either *online* or *offline*.

In an *online optimization* the optimization process is performed directly on the hardware robot. Candidate solutions are tested by making the robot move for a few seconds with their corresponding sets of locomotion parameters. While the quality and realism of the solutions is really high, the process is extremely time consuming. The process requires experiments running in real time and usually cannot be parallelized unless many copies of the robot are used at the same time. During online optimization the robotic hardware needs to be used extensively, increasing the risk of failures. Additionally, if not properly detected, some candidate solutions could produce self-collisions, excessive torques, or lead to high impacts with the ground which can damage the robot.

On the other hand, *offline optimization* takes place in a virtual environment, running either on a microprocessor inside the robot or externally on a standard personal computer or on a cluster on the cloud. Simulations can often run faster than real time and in parallel over multiple processors. Repetitive time consuming tasks such as resetting the robot after each experiment or recharging the batteries are eliminated. Locomotion parameters can be tested safely without any risk of damaging the robotic hardware. These are some of the reasons for the widespread use of simulation. The major drawback of offline optimization is that results cannot always be directly transferred to the hardware robot. Simplifications in the model and in the physics engine can often produce behaviors that are only partially reproducible, or even unreproducible, on the robot. Quoting an old article from Rodney Brooks [26]:

> *"There is a real danger (in fact, a near certainty) that programs which work well on simulated robots will completely fail on real robots because of the differences in real world sensing and actuation."*

This problem is known as the *reality gap*. In robotics, crossing the reality gap would mean that any behavior evaluation performed in simulation gives almost exactly the same outcome when performed in reality [258].

9.1.1 Related Work

Several researchers have been targeting the problem of reducing the reality gap using different approaches [108, 67, 41, 56]. Jakobi added random noise to the simulated robot's sensors to ensure that evolved controllers are robust [85]. Adams incorporated feedback from the physical robot into a NEAT evolution to improve the use of artificial evolution as a tool for generating

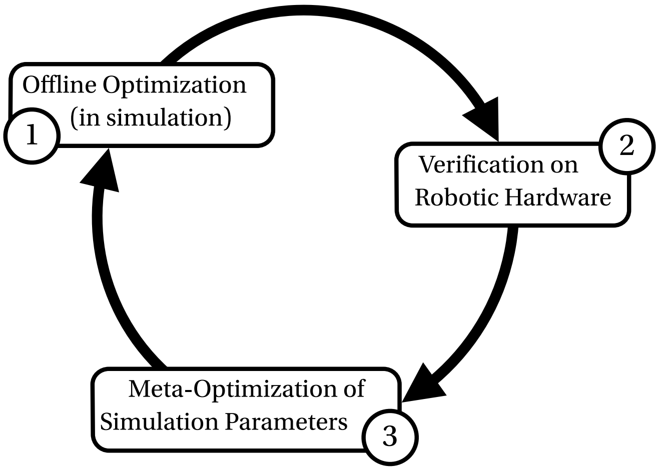


Figure 9.1 – Hybrid optimization is a cyclic process combining (1) offline optimization, (2) exploration of control parameters found in simulation on hardware, and (3) meta-optimization to improve matching of software models and hardware.

controllers for physical robots [1]. Bongard and Lipson used a co-evolutionary approach, called the estimation-exploration algorithm, for automatically improving a robot simulator by changing mass distribution and sensor time lags properties based on how well the controllers evolved in simulation transfer to the hardware robot [20]. Koos et al. developed a reality gap reduction method based on a multi-objective optimization on an 8-DoF quadrupedal walking robot [97] . They use a task-dependent fitness (evaluated in simulation) and a simulation-to-reality disparity measure that estimates the controller's transferability when reproduced on hardware. Klaus et al. published a comparison of different strategies for simulator tuning [96]. A review on different approaches to minimize the impact of reality gap has been recently presented by Eaton [50].

In this work we explored *hybrid optimization* as a solution to combine advantages of online and offline optimization to efficiently find optimal locomotion parameters for arbitrary modular structures. Using a *meta-optimization* cycle, we also optimized simulation parameters allowing for a good match with the hardware.

9.2 Hybrid Optimization

In order to exploit the benefits of offline optimization, we explored the use of hybrid optimization on a Roombots metamodule. Hybrid optimization is a cyclic method that allows to have reliable simulation results by improving control and model parameters while minimizing the use of hardware experiments. Figure 9.1 shows the three steps of this cycle. In the first step, offline optimization is performed using standard simulation and model parameters. A number of virtual sensors record the behavior and the performance of the simulated robot. A selection of optimized gaits is then transferred to the hardware and tested for a certain time.

Hardware sensors analogous to the virtual ones also monitor the behavior of the hardware robot. The third step is the meta-optimization process in which behavior of simulation model and robotic hardware for each selected gait are compared in order to optimize simulation and model parameters. After this step the cycle repeats with a offline optimization run. We will now describe in detail the work done for each step of the hybrid optimization cycle.

9.2.1 Offline Optimization

For the offline optimization step we simulated a Roombots metamodule in PER configuration [205] using Webots [229]. We used a CPG-based locomotion controller similar to the one described in Chapter 7. We used nearest neighbor coupling and a fixed oscillation frequency of 2 Hz. The open parameters that need to be optimized are six amplitudes of oscillation (A_i), six offsets (X_i), and five coupling phases (ψ_{ij}).

To optimize the CPG control parameters we used Particle Swarm Optimization with 300 particles and 200 iterations running on our computer cluster. Each particle corresponded to 30 seconds of physical simulation and was repeated three times to favor robust solutions. For the fitness evaluation we averaged the speed of the robot over the three repetition. Particles that generated self-collisions were heavily penalized. Given the complex geometry of Roombots shells, the simulation ran only slightly faster than real time and required 22 hours on our cluster.

From the 300 optimized particles we chose 17 fast gaits with no self-collisions and that seemed safe for the hardware robot in terms of impacts with the ground. Solutions were not only chosen based on their speed but also on the emerging locomotion strategy, in order to increase the variability of the gaits used for the meta-optimization process.

We added a virtual Microsoft Kinect camera in the simulation and replayed the 17 selected parameter sets while recording the outline of the robot over time. This virtual sensor was added only at this point to avoid slowing down even more the optimization.

9.2.2 Hardware Experiments

For the validation on the robotic hardware we used the test arena shown in Figure 9.2, composed of:

- A 2×2 m rubber flooring used to alleviate the impacts of the robots with the ground.
- A Microsoft Kinect (mounted overhead) using its depth sensor to track the robot and extract its center of mass and its contour.
- A control computer that logs the data from the Kinect and sends CPG control parameters via Bluetooth to the robot.

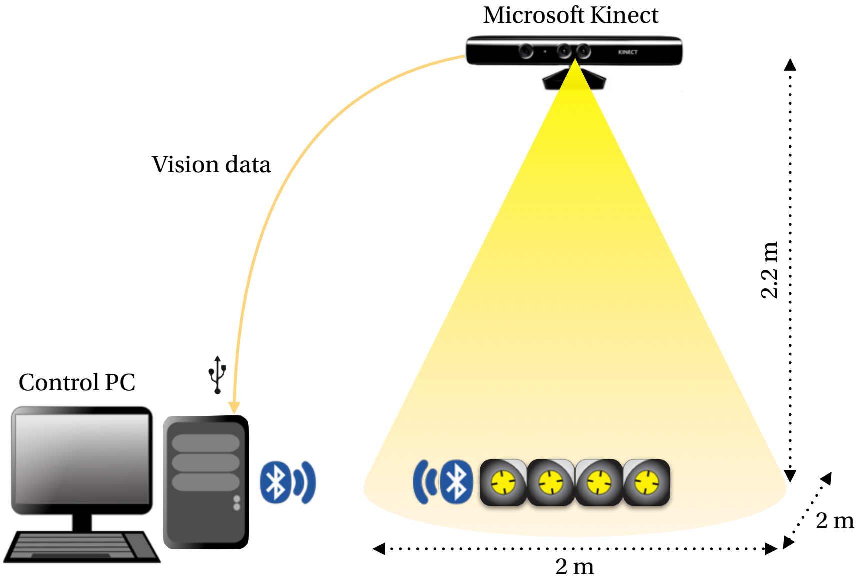


Figure 9.2 – Test arena used to run the hardware experiments. A 2×2m-wide area is in the field of view of a Microsoft Kinect mounted overhead. A control computer sends sets of CPG parameters to the robot and tracks the Roombots's projection over time. Image adapted from [130].

Similarly to the Webots simulation, each of the 17 sets of parameters was tested for a trial duration of 30 seconds while the Kinect recorded the outline of the robot. Six of the selected gaits were used in the next step of the hybrid optimization cycle for optimizing simulation parameters during the meta-optimization. The remaining 11 gaits were used later to validate the optimized simulation and model parameters. This was done to test for overfitting in the meta-optimization.

9.2.3 Meta-Optimization

For the last step of the hybrid optimization cycle we used meta-optimization to reduce the reality gap. Meta-optimization involves a second offline PSO run that does not optimize CPG locomotion controller parameters but instead simulation parameters. We decided to optimize 22 parameters (reported in Table 9.1) that characterize the Webots environment, the Roombots metamodule model, and the virtual Kinect camera. We used 100 particles and 100 iterations.

The fitness function used for meta-optimization is a quantification of the matching between the robot behavior in simulation and in hardware. In our experiments we considered the overlapping over time between the outlines of the Roombots metamodule recorded by the virtual Kinect (V) and by the hardware Kinect (H). We defined the *similarity ratio* R as the overlapped area divided by the total area of the robot projections. If there is a perfect match, R is equal to 1. Figure 9.3 shows an example frame with the virtual projection V (in blue), the

Table 9.1 – Simulation and model parameters subject to meta-optimization. Units follow the Webots standards.

#	Parameter description	Search range	Hand-tuned value	Value after meta-optimization
	Parameters of simulation environment			
1	Constraint force mixing (CFM) in Webots	[0.0, 0.001]	0.00001	0.00014892
2	Error reduction parameter (ERP) in Webots	[0.1, 0.8]	0.2	0.3652
3	Coulomb friction between Roombots's body and ground surface	[0.1, 5.0]	1	0.8018
4	Bounce between Roombots's body and ground surface	[0.0, 1.0]	0.5	0.8040
	Parameters of Roombots model			
5	Mass of metamodule's half-sphere 1	[0.280, 0.370]	0.325	0.3038
6	Mass of metamodule's half-sphere 2	[0.280, 0.370]	0.325	0.3142
7	Mass of metamodule's half-sphere 3	[0.280, 0.370]	0.325	0.2939
8	Mass of metamodule's half-sphere 4	[0.280, 0.370]	0.325	0.3308
9	Mass of metamodule's half-sphere 5	[0.280, 0.370]	0.325	0.3079
10	Mass of metamodule's half-sphere 6	[0.280, 0.370]	0.325	0.3316
11	Mass of metamodule's half-sphere 7	[0.280, 0.370]	0.325	0.3438
12	Mass of metamodule's half-sphere 8	[0.280, 0.370]	0.325	0.3017
13	Diametrical DoF Maximum velocity	[1.0, 15.0]	2.5	6.8721
14	Diametrical DoF Maximum torque	[2.5, 15.0]	4.8	9.0643
15	Diametrical DoF Control parameter P	[5.0, 15.0]	10	6.9873
16	Diametrical DoF Damping constant	[0.0, 1.0]	0	0.1172
17	Central DoF Maximum velocity	[1.0, 15.0]	1.5	7.5193
18	Central DoF Maximum torque	[2.5, 15.0]	3.2	12.2080
19	Central DoF Control parameter P	[5.0, 15.0]	10	12.8652
20	Central DoF Damping constant	[0.0, 1.0]	0	0.5622
	Camera parameters			
21	Height of the virtual Kinect depth camera [cm]	[210.0, 240.0]	260	227.8160
22	Time shift between hardware and virtual camera [ms]	[0.0, 1200.0]	900	424.1056

hardware projection H (in red), and their overlap (in green).

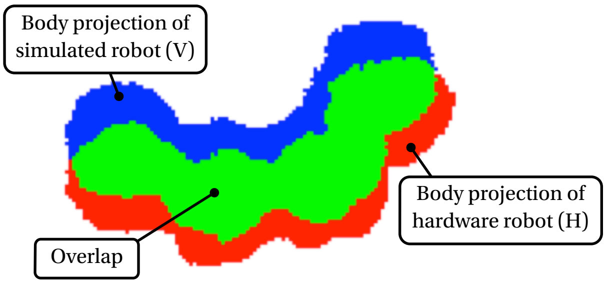

Body projection of simulated robot (V)
Body projection of hardware robot (H)
Overlap

Figure 9.3 – Similarity measured as overlap between the robots' projections onto the ground as observed by a virtual and a hardware Kinect. Image adapted from [130].

Since meta-optimization has to maximize the similarity ratio over time (for each frame), simultaneously for all selected gaits, we defined the following fitness function:

$$F_{meta} = \frac{\sum_{gaits}\left[W_{gait} \cdot \frac{1}{\sum_{frames}} \cdot \sum_{frames} R\left[H(t), V(t+s)\right]\right]}{\sum_{gaits} W_{gait}} \tag{9.1}$$

where $H(t)$ is the movie frame at time t from hardware experiment and $V(t+s)$ is the movie frame from the virtual Kinect at time $t+s$, where s is a time shift variable defined to synchronize hardware and simulation trials. For each gait the similarity rate R is modified by a gait-dependent weight W_{gait} that is influenced by the speed and repeatability in hardware (more details in [130]).

9.3 Experimental Results and Discussion

We followed the steps described in Section 9.2 to perform one full cycle of hybrid optimization for the locomotion of a Roombots metamodule on flat ground. From the 17 selected optimized gaits, we used 6 (training set) to run the meta-optimization algorithm. Table 9.2 shows the values of similarity ratio R before and after the optimization of simulation parameters. We recorded a 65% increase in similarity ratio for the 6 gaits in the training set.

Figure 9.4 shows snapshots of the fastest gait found with the first cycle of hybrid optimization. The robot achieved a speed of 6 cm/s in simulation and 5.8 cm/s in hardware. Visually the matching seems really good even though a similarity ratio of 0.4 is far from the ideal value of $R = 1$. We concluded that this is due to the following reasons:

- The measure of similarity that we defined might not be optimal. We used a measure based on the visual similarity over time of the robots' outlines as seen by external Kinect cameras. This might not capture the full dynamics of the system, limiting the

Table 9.2 – Improvement of similarity ratio R between simulated and hardware robot after meta-optimization. The training set corresponds to the 6 gaits used for meta-optimization. The remaining 11 selected gaits are part of the test set.

	R of hand-tuned simulation	**R of meta-optimized simulation**	**Improvement**
Training set	0.250317	0.414293	65.51 %
Test set	0.325153	0.402104	23.67 %
Training + Test set	0.281152	0.409271	45.57 %

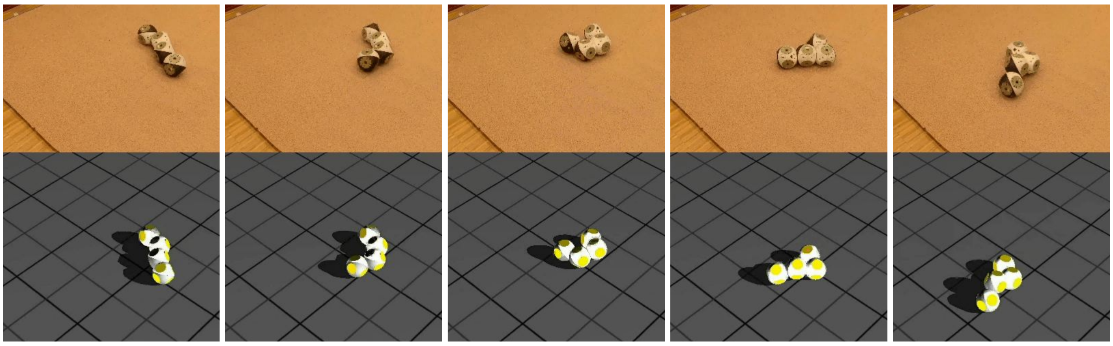


Figure 9.4 – Snapshots of a Roombots metamodule gait in hardware (top) and simulation (bottom) after a cycle of hybrid optimization. Similarity ratio R is about 0.4. Time between snapshots is 12 seconds. Images from [130].

effectiveness of the meta-optimization step.

- The evaluation of F_{meta} required perfect synchronization between the images recorded by the virtual Kinect and the hardware one. The use of the variable s in $R[H(t), V(t+s)]$ might not be sufficient to also adjust for frame rate differences.
- The simulation and model parameters that we optimized with meta-optimization (listed in Table 9.1), and more generally the physics generated by ODE[1], might not be sufficient to simulate complex realistic dynamics.
- Noise in the hardware experiments always produces small differences between different repetitions of the same locomotion pattern.

To verify how well the meta-optimization generalizes to other gaits we compared hardware and “meta-optimized” simulation runs with the 11 gaits in the test set. Table 9.2 shows that the similarity ratio reached by the gaits in this test set in comparable to the R of the training set. Thus, results of meta-optimization can be generalized, at least for different locomotion patterns with the same robot. The optimized simulation and model parameters can be used to run additional offline locomotion parameter optimizations to find new suitable gaits. At any point, hardware validation can be used to evaluate the reality gap and, if necessary, trigger a new hybrid optimization cycle. Meta-optimization results might not generalize to gaits performed by different robots, or in case of self-reconfiguration for SRMRs. In these cases, hardware validation should be performed early on.

9.4 Conclusions

We presented a hybrid optimization technique that combines advantages of offline and online optimization by providing simulation results that are well matched with hardware experiments. The process starts with a run of offline optimization to find efficient and safe locomotion controller parameters. A set of selected solutions are evaluated in hardware while the performance of the robot is recorded. The last step, the meta-optimization, aims at improving simulator and model parameters through a second optimization process. The fitness function of this second optimization is related to the percentage of matching between simulation and hardware experiments. The hybrid optimization cycle can be repeated multiple times until a good matching is achieved.

In contrast to offline optimization, the extra steps required by hybrid optimization allow results to be almost seamlessly transferred to the hardware robot. Compared to online optimization, we estimated at least a 2.5-fold reduction in the required optimization time (a detailed analysis is reported in [130]). The benefits of hybrid optimization increase for more extensive optimizations with a larger number of particles and iterations. Additionally, most of the time required for hybrid optimization is computation time which does not require the physical

---

[1] Open Dynamics Engine, the physics engine used in Webots [144].

presence of an operator. Lastly, unsafe parameter sets that could lead to self-collisions, high actuator torque, or excessive impact forces with the ground can be tested in simulation and discarded before being used on the hardware robot.

The hybrid optimization technique can be used in combination with our BLF and SYM methods to create a locomotion optimization framework that allows for fast locomotion-parameter learning after a morphological change. This was used for example to optimize the quadrupedal gait of the Roombots structure presented in Appendix C.

This concludes our contribution on methods that allow arbitrary modular structures to learn to locomote in efficient ways. In the next chapter we will explore another contribution to the locomotion challenge which is the use of compliance in the structure to improve locomotion speed and energy efficiency.

10 Locomotion with Compliant Elements

In Part II, Chapter 4, we discussed how scalability affects the design choices for the actuation system of a homogeneous Reconfigurable Modular Robot. Moreover, in Section 6.3 we described how compliance could be used to push performance and gave details on how we included passive compliant elements in Roombots structures and in the Reconfigurable Modular Robot Lola-OP™. In this chapter we show the results of three studies in which we studied the effect of compliance on locomotion.

In Section 10.1 we use two types of in-series compliant elements (rotational and omnidirectional), with five different stiffness values for each of them, on a structure made of two Roombots modules. We run dedicated online locomotion-parameter optimization for six different configurations of the hardware robot moving on flat ground, evaluating their performance in terms of locomotion speed. For both types of compliant elements we report an increase of locomotion speed when using softer elements.

In Section 10.2 we create a simulated model of an 8-DoF Lola-OP™ Modular Snake Robot with in-series omnidirectional compliant elements to explore the effect of element stiffness for different gaits, types of terrains, and locomotion frequency. Compliance is simulated with a compliant ball joint with eight different levels of stiffness. Two snake locomotion gaits (rolling and sidewinding) are tested over flat ground and three different types of rough terrains. We perform grid search and Particle Swarm Optimization to identify the locomotion parameters leading to fast locomotion, and analyze the best candidates in terms of locomotion speed and energy efficiency (cost of transport). Compared to the previous results obtained with Roombots modules, presented in Section 10.1, in this study we do not observe the same clear trend in favor of using compliant structures but a more flat response over different levels of compliance. Compliant and stiff elements lead to comparable performances.

In Section 10.3 we report part of the experiments done in order to extend the work of Section 10.2 using a hardware implementation of the compliant 8-DoF Lola-OP™ Modular Snake Robot with four different levels of compliance. We perform grid search experiments with rolling gaits on horizontal pipes with several geometrical characteristics such as different diameter, surfaces with presence of obstacles, and considerable changes in diameter. Performance across different compliance values is evaluated in terms of speed of locomotion and power consumption. While also in this scenario locomotion speed is similar at different levels of stiffness, the compliant robots prove to better adapt to different terrains, especially in terms of power consumption. Motor current measurement show that the stiff robot uses more than double the power compared to softer one, which often results in actuators heating up and engaging their safety shut-down feature.

Some parts of this chapter have been adapted from:

[224] M. Vespignani, E. Senft, S. Bonardi, R. Moeckel and A.J. Ijspeert. “An experimental study on the role of compliant elements on the locomotion of the self-reconfigurable modular robots Roombots". In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Nov. 2013, pp. 4308-4313.

> **My original contribution**: Conceptual contribution, Experimental setup, Hardware experiments, Data analysis, Writing.

[226] M. Vespignani, K. Melo, S. Bonardi, and A.J. Ijspeert. “Role of Compliance on the Locomotion of a Reconfigurable Modular Snake Robot". In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sept. 2015.

> **My original contribution**: Conceptual contribution, Software development, Simulation experiments, Hardware experiments, Data analysis, Writing.

[225] M. Vespignani, K. Melo, M. Mutlu, and A.J. Ijspeert. “Compliant snake robot locomotion on horizontal pipes". In *Safety, Security, and Rescue Robotics (SSRR), 2015 IEEE International Symposium on.* Oct. 2015.

> **My original contribution**: Conceptual contribution, Experimental setup, Hardware experiments, Data analysis, Writing.

10.1 Roombots Locomotion with In-Series Compliant Elements

One of the challenges in the field of (Self-) Reconfigurable Modular Robotics is to design modules strong enough to be able to lift several times their own weight. For this reason, many robots in literature are designed with large torque capability (e.g. with high reduction ratio gearboxes), resulting in rather slow speeds and non-dynamic gaits [182].

This research study aims at investigating whether compliance can be beneficial for the locomotion of SRMRs, pushing their physical performance boundaries (e.g. getting more dynamical and energy efficient gaits by storing and releasing elastic energy) and reducing the complexity of their control system (e.g. to passively adapt to the shape of the terrain [248]), with the ultimate goal of contributing to the hardware scalability challenge [249]. To pursue this objective, we investigated the effect on flat-ground locomotion of added in-series compliance in the inter-connection between two Roombots modules (Figure 10.1).

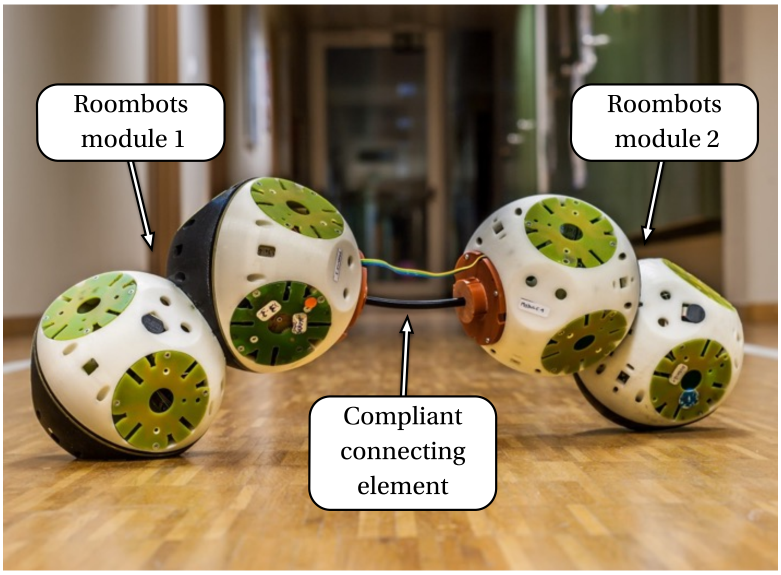


Figure 10.1 – Two Roombots modules interconnected with a compliant rod. Image adapted from [224].

This work is guided by the following three main hypotheses:

1. Compliance can improve the locomotion performances of a Self-Reconfigurable Modular Robot. Among the several factors that can define the performance of locomotion, we considered the speed (displacement divided by elapsed time) of the gait and its repeatability.
2. Different types of compliant elements will produce a significantly different behavior. For this study we analyzed the effect of torsional and omnidirectional springs.
3. When testing a set of compliant elements with different values of stiffness, the loco-

motion control parameters optimized for an intermediate value of the set will perform better when tested on any other member than the control parameters optimized for the stiffest or the softest element in the set.

10.1.1 Related Work

While the role of compliance in a monolithic robotic structure has long been suggested and studied, for instance in the field of articulated locomotion [164, 80], to the best of our knowledge this is still a scarcely explored topic in the field of Reconfigurable Modular Robot.

Previous work has been done by Aoi et al. [5], who investigated the role of joint compliance in achieving high maneuverability for the locomotion of a simulated multilegged modular robot. Yu et al. [257] presented Morpho, a self-deformable modular robot where active and passive links work together to shape the structure into different geometries.

The work that reflects more closely our objectives of exploring and exploiting the effect of compliance in a (Self-) Reconfigurable Modular Robot was done by Sastra et al. [181, 184]. In their research, they used the reconfigurable modular robot CKBot as a tool to quickly build robotic structures with different morphologies and explore a novel biologically-inspired legged style of locomotion. They designed a set of purely passive compliant legs for CKBot in order to increase its dynamics and show a bouncing gait that runs like a Lateral Leg Spring (LLS) model.

10.1.2 Materials and Methods

Compliant Elements

In this work we studied the effect of multiple types of compliance in several structures using two Roombots modules. The modules were assembled in a configuration similar to a Roombots metamodule in PER configuration [205]. However the modules were not directly connected to each other but separated using passive compliant elements between them, as described in detail in Section 6.3.

We used two types of compliant elements: *Compliant Rods* (CR), that are cylindrical structures that can bend in any direction tangential to their longer axis and twist around it, and *Torsional Springs* (TS), that only allow rotation. For each of the two types of elements we chose five stiffness levels, as reported in Table 6.1.

To ensure mechanical stability and remove any external source of uncertainty, we mechanically fixed the compliant elements to the modules (using the same threaded insert usually used to hold the connector plates) instead of using ACMs. To simplify the communication between central host and robots and ensure perfect synchronization we used only one wireless communication board and a wired communication channel between modules (the intra-module

Table 10.1 – Open CPG parameters used for the on-line PSO optimization of two Roombots modules with in-series compliant elements.

Variable	**Range**	**Units**
Amplitude	[0.8, 0.75π]	[rad]
Offset X_3	[-1.2, 1.2]	[rad]
Offset X_4	[-1.2, 1.2]	[rad]
Phase lag ψ_{23}=ψ_{32}	[$-\pi$, π]	[rad]
Phase lag ψ_{34}=ψ_{43}	[$-\pi$, π]	[rad]
Phase lag ψ_{45}=ψ_{54}	[$-\pi$, π]	[rad]

communication cable can be seen in Figure 10.1).

Control and Optimization Framework

The two Roombots modules were controlled using a network of six coupled non-linear oscillators representing a CPG. We designed an oscillator network topology which matched the hardware morphology, with one oscillator per degree of freedom (Figure 10.2), with rules according to those described in Chapter 7.

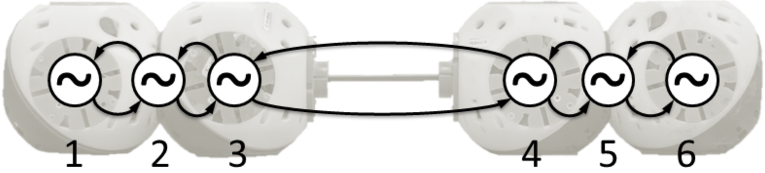


Figure 10.2 – CPG topology used to control two Roombots modules with in-series compliant elements. The topology is similar to the one used to control a Roombots metamodule: each joint is represented by one oscillator and all oscillators have nearest-neighbor coupling. Image from [224].

To reduce the number of open parameters, we used only one amplitude $A_i = A$, set the external oscillators (1 and 6) to a locked state, and set the offset for the oscillators 2 and 5 to zero (we assumed that the external geometry of the modules could be simplified with spheres). We did not however induce any artificial symmetry, i.e. we did not apply any mirroring of parameter sets along our network, to avoid restricting the possible variety of parameters. Table 10.1 summarizes the open CPG parameters and their range of values.

In order to find the fastest gait for each type of compliant element, we let a population-based algorithm based on Particle Swarm Optimization (PSO) provide an automatic design of the CPG control input parameters. For this work we chose to run all the experiments with the hardware robot, avoiding simulated (off-line) gait optimization experiments. This choice was based on the following reasons:

- Compliant elements are in general complex to model and can often induce numerical instability. At the time this work was performed, Webots [229] (Version 6.4.4) did not include *compliant BallJoint nodes* (feature that we used in the work presented in Section 10.2). Instead, compliant elements could be modeled using a *physics plugin* (slow, computationally expensive), or a combination of compliant hinge joints (unstable).
- Approximations in the model and in the physics engine result in a gap between simulation and reality. This makes it difficult to transfer results from simulation to the hardware robot, even though hybrid optimization techniques (as explained in Chapter 9) could be used.
- The complex shape of Roombots modules, combined with the possible need for a physics plugin or multiple joints, slows down simulation time to one or more orders of magnitude below real-time. Simulations can anyway run in parallel and without the constant supervision of a human operator, but we considered the time required to develop the simulated model, combined with the time needed to transfer results to the hardware robot, was not worth the advantages that this slow simulation could bring.

Therefore, instead of using simulation, all the parameters used herein were evolved using on-line optimization, with each particle of the PSO tested directly on the hardware modules. The typical optimization for one set of parameters consisted in 9 particles and 20 iterations (180 particles to be tested).

The test arena used for on-line optimization is similar to the one used for hybrid optimization (Figure 9.2). For each trial, the computer generates the PSO particle containing the new CPG parameters to be tested. These are sent via Bluetooth to the Roombots structure, which checks whether to accept the set of parameters using an internal collision detection routine. If the particle is discarded, a fitness value of zero is returned. Otherwise the robot uses the parameters in its on-board CPG controller, for a trial duration of 30 seconds. The first five seconds of the trials are not evaluated, in order to wait until the CPG reaches a stable state. At the end of the trial, the computer measures the robot's displacement using the position data collected with the Kinect and calculates the fitness value as displacement divided by elapsed time. We did not restrict the evaluation of the speed to any specific direction. We also did not use any strategy in order to minimize the sideways displacement compared to the main direction of motion. The position information provided by the Kinect was used solely for the fitness evaluation. The CPG controller runs in open-loop with no sensory feedback components.

Comparison Methodology

Defining a good quantitative experimental protocol to evaluate and compare the effect of compliance in a robotic structure can be a quite challenging task since several factors are affected. In order to evaluate Hypothesis 1 (improvement of performance), we compared the maximum speed that we were able to achieve for each configuration and stiffness value. In the data analysis, we also considered repeatability, a measure of the variability of the speed when repeating the same experiment multiple times. Since the CR configurations have a bigger size (5 Roombots grid units) compared to the TS (~4 Roombots grid units), we also considered the body-lengths per second as a simplistic method to compare different types of compliant elements.

Since the speed of locomotion depends both on the gait and on the type of compliant element used, and considering that a good gait for one element might not be well suited for another one, we developed the following experimental protocol to test Hypothesis 3:

- We defined three test cases (*soft structure*, *medium structure*, and *stiff structure*) corresponding to the first, third, and fifth stiffness value for each type of compliant element.
- We ran a full on-line optimization (using speed of locomotion to evaluate the fitness value) for each of these three cases (a total of six optimizations) to find good sets of control parameters (*soft pattern*, *medium pattern*, *stiff pattern*), fit for the value of stiffness of each test case.
- We tested these patterns on structures with the same type of compliant element but different stiffness values and evaluated how well they performed.

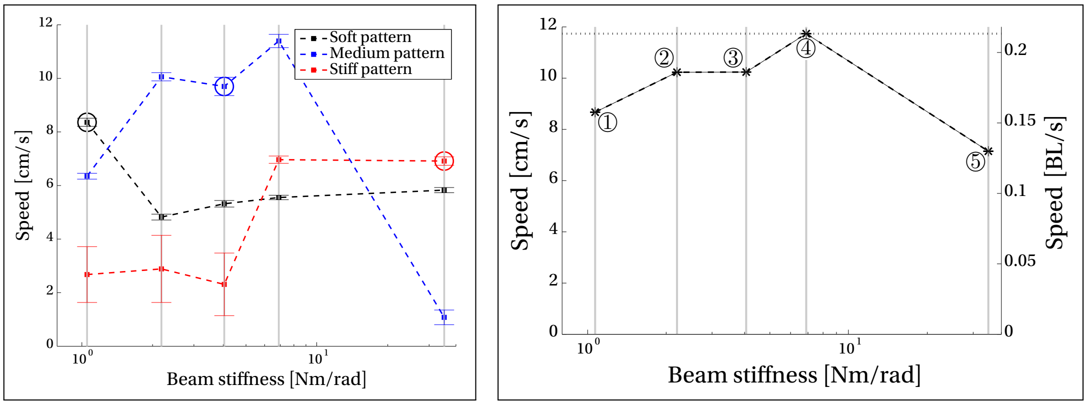
Hypothesis 2 was evaluated qualitatively from video recordings of the experiments.

10.1.3 Experimental Results and Discussion

Following the methods described in the previous section, we ran six different on-line parameter optimizations, namely for CR1, CR3, CR5, TS1, TS3, and TS5, and obtained six sets of CPG parameters (*patterns*). We then tested each compliant element with all three sets of parameters obtained for that type of compliant element. We repeated each experiment 10 times (30 seconds per run) to test the repeatability of the gait. Using the Kinect tracking, we evaluated the displacement and thus the speed.

Figures 10.3.a and 10.4.a show the mean speed value for each CR and TS experiment. The error bars represent the standard deviation of the 10 repetitions. The x-axes are represented in a logarithmic scale in order to properly fit the stiffness values used herein. The circles represent the data points that have been optimized for that particular structure.

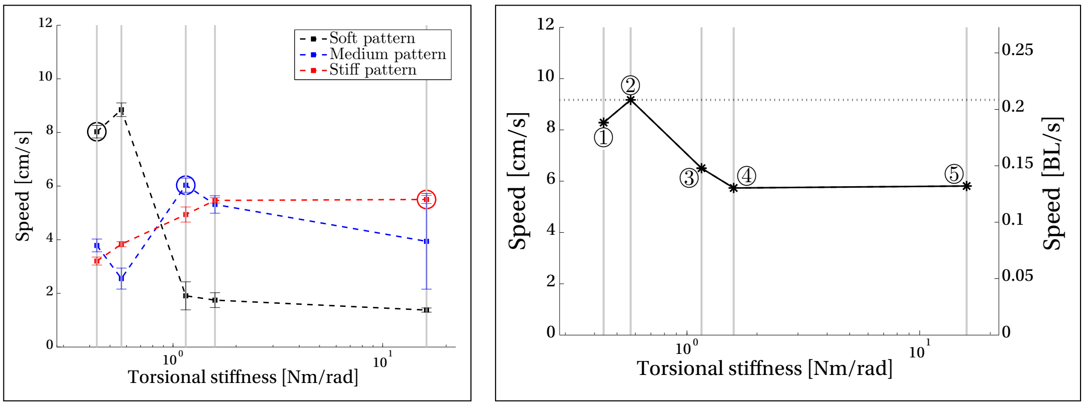
For the Compliant Rod (Figure 10.3.a), the *medium pattern* can be used to achieve very good



(a)

(b)

Figure 10.3 – Results of the experiments with the Compliant Rod (CR) elements: (a) Mean robot speed for each type of pattern elements. The error bars represent the standard deviation for 10 repetitions. The circled data points are the direct result of on-line optimization. (b) Maximum robot speed for each level of compliance. The speed in body lengths per second was calculated using a length of five Roombots units ($0.55m$). Points two and four are not the result of a dedicated optimization but obtained by running the *medium pattern* CPG parameters. Images from [224].



(a)

(b)

Figure 10.4 – Results of the experiments with the Torsional Spring (TS) elements: (a) Mean robot speed for each type of pattern elements. The error bars represent the standard deviation for 10 repetitions. The circled data points are the direct result of on-line optimization. (b) Maximum robot speed for each level of compliance. The speed in body lengths per second was calculated using a length of four Roombots units ($0.44m$). Points two and four are not the result of a dedicated optimization. Point two was obtained by running the *soft pattern*; point four was obtained by running the *medium pattern* CPG parameters. Images from[224].

locomotion gaits while changing stiffness value (Hypothesis 3), except for the *stiff structure*. For this configuration, the presence of compliance deeply affected the locomotion pattern. The *stiff pattern* produced a movement perpendicular to the main axis of the robot, with the modules generating some momentum in order to roll the structure.This strategy failed when tested with more compliant elements because the structures were not able to flip over. On the contrary, the *soft* and *medium patterns* generated a movement normal to the main axis, in which one module is always ahead and pulls the other module forward. The *medium pattern* performed badly with CR5 because after a few locomotion cycles the structured became unbalanced and flipped over.

For the Torsional Spring (Figure 10.4.a), the *soft pattern* generates a sideways crab-like motion that uses high amplitude and exploits the high deformability of the compliant element. It however fails at increased stiffness because the high amplitude makes the structure flip. The *stiff pattern* uses almost the same control strategy, except for the amplitude that is much smaller and keeps the structure stable. On the other hand, the *medium pattern* evolved using a completely different strategy that was not well suited for other stiffness values. Overall, the behavior of the Torsional Spring was quite different from that of the Compliant Rod (Hypothesis 2).

Figures 10.3.b and 10.4.b show the maximum robot speed that we achieved with the Compliant Rod and the Torsional Spring elements. The curves represented in the plots are the envelopes of the maximum values of Figures 10.3.a and 10.4.a.

For each type of compliant element, there seems to be an optimal level of compliance (Hypothesis 1). For the Compliant Rod, we have a peak of performance for a medium level of compliance, with an increase of speed of almost 65% compared to the stiff case. For the Torsional Spring, instead, there is a peak for low stiffness values (57% more speed than the stiff case). This is because the modules are using the soft spring as an additional (almost with no resistance) degree of freedom.

10.1.4 Conclusions

In this section we presented a study on compliance using structures made of two Roombots modules connected with two types of in-series compliant elements with five possible stiffness levels each. This research was driven by the following three hypotheses:

1. Compliance can improve locomotion performance.
2. Different types of compliance will result in diverse locomotion behaviors.
3. Control parameters optimized for a medium level of compliance will perform better for other values of compliance than parameters optimized for extremal compliance.

We ran dedicated on-line CPG parameter optimizations for six different configurations and evaluated the performance using the speed and the repeatability of the gait. Different types and values of compliance produced quite different gaits (Hypothesis 2) and it was hard to predict how the gait would be affected by them. Results show a clear increase in performance (up to

65%) after the introduction of in-series compliance (Hypothesis 1), thus making compliance a possible way to partially tackle the hardware scalability challenge. Hypothesis 3 remains a partially open question, depending on the type of compliance that is added to the SRMR. For omnidirectional compliance, it is possible to transfer the optimized gait parameters with quite good results.

This work could be extended by studying different Roombots structures with a more diversified set of compliant elements, including more in-series elements and also external compliant extensions. Other factors could also be considered to better evaluate the effect of compliance on the locomotion of a SRMR, such as power consumption (as done in the work presented in Section 10.3), impact forces, and measurement of the deformation of the compliant element. In the next sections we will present followup work done using a Reconfiguring Modular Snake Robot.

10.2 Lola-OP™ Locomotion with in-series Compliant Elements

Snake robots can be seen as an example of chain-type Reconfigurable Modular Robots [249]. The connectivity of the modules forming the structure can be easily rearranged manually by an external operator (no self-reconfiguration capabilities), for instance to change the number of elements that create the snake robot. Exactly like SRMRs, also snake robots suffer actuator scalability and limited power as the number of modules increases.

In this work we extended our previous study on the effect of added in-series compliance to a reconfigurable modular snake robot. Compliance in the snake robot structure could bring many advantages such as faster locomotion speed, higher energy efficiency (e.g. by using the compliant elements to store and release elastic energy), reduced shock loads on the robot and its actuators, and improved terrain handling capabilities (e.g. for crossing features that offer fewer points of contact, such as moving along cylindrical pipes), compared to having completely rigid body and actuators.

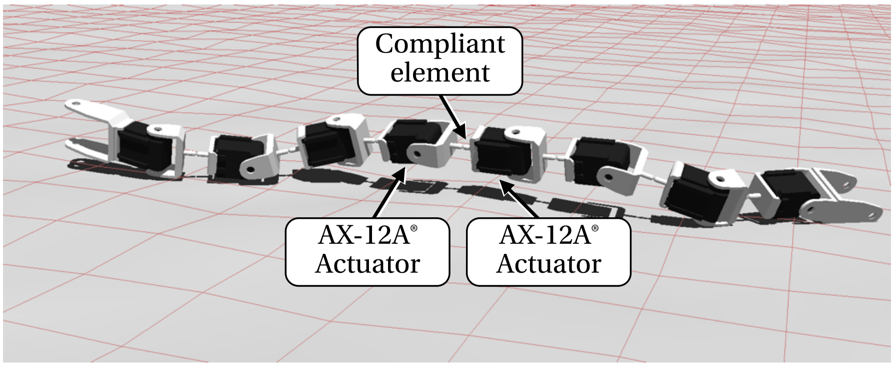


Figure 10.5 – Simulated model of the 8-DoF Lola-OP™ robot with in-series compliant elements. Image from [226].

We developed a simulated model based on a compliant Lola-OP™ modular snake robot (Figure 10.5) and we analyzed its behavior as a function of the stiffness value of the compliant elements for two types of gaits (i.e. rolling and sidewinding) on four different terrains.

10.2.1 Related Work

In our previous work [224], described in Section 10.1, we explored the effect of added in-series compliance using structures made of two Roombots modules. The outcome showed an optimal level of compliance for which the locomotion speed was increased up to 65% compared to the stiff configuration. Related work on the use of compliant elements or compliant behaviors in snake robots is limited to few contributions. In [242], the design of a robotic modular snake robot for real-world applications is presented. The authors introduce a novel force/torque sensor based on a compliant element built in the actuator module. Although the overall behavior of the robot is not intrinsically compliant, the design allows such modules to use some limited impedance control strategies. However, performance of the compliant behavior is not reported. Rollinson & Choset [169] presented an interesting work on an active compliant strategy for snake robots. This contribution aims to exploit current-based torque control to simulate a compliant behavior useful for adapting gaits to some specific environments (i.e. poles and pipes). Lately, Rollinson presented as a part of his PhD dissertation [172] the design of a new snake robot specifically designed to present series elastic actuators [170]. In this contribution, the addition of a compliant element attached to the axis of actuation of each module that composes the snake robot is explained in detail. The resulting series elastic module enables the exploration of some interesting torque controlled behaviors [171], as well as the exploitation of the robot's compliance to search for a comprehensive framework for terrain adaptation and obstacle negotiation schemes. Sato et al. [186] developed HAUBOT, a 18-segment snake robot with an elastic material in-series with each actuator. The same authors presented HAUBOT-II, a snake robot with variable joint stiffness that can adapt in real-time to environmental changes [185]. A similar preliminary work was reported in [121], in which a snake robot with variable stiffness actuators was constructed. This robot uses VSA-CubeBots as modules to create the snake body. Some experiments and demonstrations of this robot negotiating cluttered environments while executing simple gaits thanks to the wise choice of the compliant/stiffness preset value of the actuators are available, though not reported.

10.2.2 Materials and Methods

We conducted the experiments presented in this work using a simulated model of the robot developed in Webots [229]. The shape of the robot has been directly imported from CAD models provided on the Bioloid website [12] after minor simplifications to reduce unnecessary features (e.g. screw-holes) that could slow down simulation time. Physical properties of the hardware robot, such as weight of the different components, and the motors' maximum speed and stall torque have been included in the simulated robot. One advantage of using simulation

Table 10.2 – Characteristics of the terrains used for the snake robot locomotion with in-series compliant elements. The maximum slope is computed as $tan^{-1}(\frac{h_{max}-h_{min}}{spacing})$.

Name	**h$_{min}$ [m]**	**h$_{max}$ [m]**	**Max slope [deg]**
FLAT	0	0	0
RT1	0	0.01	5.7
RT2	0	0.05	26.6
RT3	0	0.1	45.0

is the possibility of seamlessly adding sensors that could not be easily included in the physical Lola-OP™ robot; the current model has absolute position sensors inside each module to localize them in 3D and position and torque sensors for each of the motors.

Terrains

Working in a simulated environment allows one to quickly evaluate the performance on different types of simulated terrains. We tested the robot on flat ground (FLAT) and on three types of rough terrains (identified as RT1, RT2, and RT3) with increasing peak and slope sizes. We generated the rough terrains using an elevation grid defined by a 100x100 matrix of random values between h_{min} and h_{max}. The distance between vertices in the elevation grid is set to *spacing* =0.1 m. The details of these terrains are reported in Table 10.2.

In order to ensure randomness in the roughness encountered by the robot, we can choose to initialize the robot in a different position and/or orientation on the terrain. For the optimization framework used in this work we initialized the robot in a random orientation for each

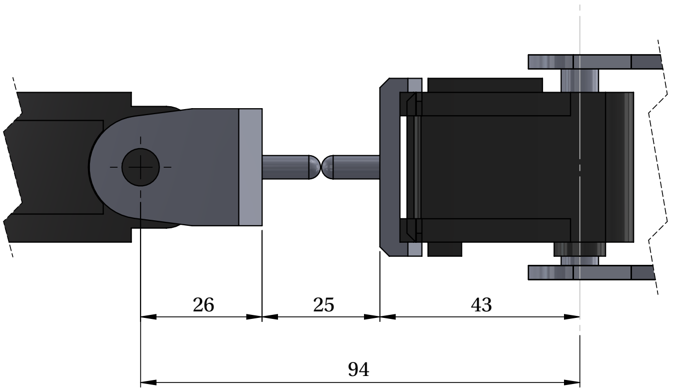


Figure 10.6 – Detail of the CAD model of the simulated Lola-OP™ robot with in-series compliant elements. Only two modules and one compliant element are shown. Dimensions are expressed in mm. Image from [226].

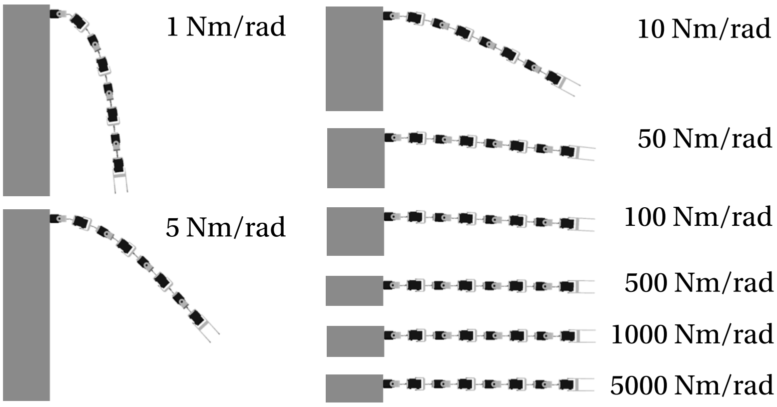


Figure 10.7 – Simulation of compliant 8-DoF Lola-OP™ robots with eight different levels of stiffness, under the effect of gravity. All servos are blocked at 0 position to show solely the bending of the compliant elements. The leftmost module is rigidly fixed to a frame. Image adapted from [226].

trial.

Compliant Elements

For this work we analyzed the effect of added in-series compliance in the interconnection between each module of an 8-DoF Lola-OP™ robot. We considered what we called omnidirectional compliance [224], which represents a beam that can bend in any direction tangential to its longer axis and has also axial torsion. Details about the hardware implementation of these compliant elements are given in Section 6.3.2.

For the simulated version of the robot we modeled these elements using a passive ball joint with a user-defined stiffness. One downside of this implementation (using the latest version of Webots available at the time: 8.0.5) is the impossibility of defining different stiffness values for the three different directions of motion of the ball joint, but it is sufficient as a first approximation since flexural and torsional stiffness of the elements are in the same order of magnitude.

For the simulated robot, we fixed the length of the deformable element to $\ell = 25$ mm (Figure 10.6) and represented it with a compliant ball joint placed at the center of this beam. We explored a large range of stiffness values and we chose the following eight values according to a reasonable range:

$$\begin{bmatrix} 1 & 5 & 10 & 50 & 100 & 500 & 1000 & 5000 \end{bmatrix} \frac{Nm}{rad}$$

The smallest value, 1 Nm rad^{-1}, corresponds to a soft element that deforms under the nominal forces applied by the motors. We chose a stiffness value of 5000 Nm rad^{-1}, which appears completely stiff in simulation, to be an upper bound. To get an intuition of the chosen range of values, we show the effect of gravity on cantilevered 8-DoF Lola-OP™ robots (Figure 10.7).

Control Framework

The specific orthogonal distribution of degrees of freedom in the Lola-OP™ robot (i.e. two bending planes) allows to use an adaptation of Hirose's serpenoid curve [75] for our gait control scheme, as in [123]. Starting from the head of the robot, the actuators are divided in odd-numbered and even-numbered, corresponding to the two orthogonal axial directions. A robot's configuration is determined by the set of joint angles $\theta(n,t)$ in Equation 10.1.

$$\theta(n,t) = \begin{cases} O_o + A_o \sin\left(\dfrac{n}{\lambda_o} + \omega_o t\right), n\ odd \\ O_e + A_e \sin\left(\dfrac{n}{\lambda_e} + \omega_e t + \delta\right), n\ even \end{cases} \tag{10.1}$$

In this function, n represents the module number and t is the time step. There are nine real-valued parameters, including offsets ($O_{o,e}$) and amplitudes ($A_{o,e}$) for both odd and even modules. A phase shift, δ, is included to adjust timing between the odd and even parts of the function. The sin function argument represents the frequency of the motion. There are two components of this frequency: spatial frequencies $1/\lambda_{o,e}$ (one for each module parity), that are multiplied by the module number variable n and whose inverse can be seen as the number of wavelengths along the robot's body length, and temporal frequencies $\omega_{o,e}$ that multiply the time variable t and determine the speed of the gait cycle.

Hirose's serpenoid curve is used for its simplicity and similarity with wave functions. By modifying these nine parameters, it is possible to recreate most of the snake gait patterns. For this work we decided to convert these equations into a network of coupled non-linear phase oscillators. This will give out system nice properties such as smooth transitions and disturbance rejection, as described in Chapter 7.

The simulated robots were therefore controlled using an open-loop CPG network [82], with one oscillator per actuated degree of freedom (motor). The coupled phase oscillators are implemented in the form of the following coupled differential equations:

$$\dot{\phi}_i = 2\pi \cdot f + \sum_j w_{ij} \cdot r_j \cdot \sin\left(\phi_j - \phi_i - \psi_{ij}\right) \tag{10.2}$$

$$\dot{r}_i = a_i \left(R_i - r_i\right) \tag{10.3}$$

$$\theta_i = r_i \cdot \sin\left(\phi_i\right) + X_i \tag{10.4}$$

i and j are the indexes of the oscillators. The output, θ_i, directly controls the position set

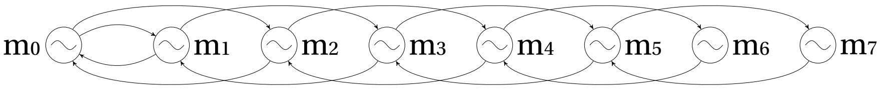


Figure 10.8 – Topology of the CPG network implemented for the Lola-OP™ robot. Even numbered DoFs are coupled together. Same goes for odd numbered DoF. A bidirectional coupling between modules m_0 and m_1 allows to adjust the phase shift between the two orthogonal directions of motion. Image from [226].

point of the motor actuating the i^{th} degree of freedom. Once in steady state (typically after 5 seconds [205]), θ_i follows a sinusoidal oscillation with an amplitude, r_i, a phase, ϕ_i, and an offset, X_i. We use one common frequency, f, for all oscillators. Bi-directional couplings follow the rule ψ_{ij} = -ψ_{ji}. All coupling weights, w_{ij}, are set to 0.5. The constant, $a_i = 2$, controls how quickly oscillators reach a modified amplitude.

For the coupling phase ψ_{ij}, considering the proposed equations of motion for the Lola-OP™ robot (Equation 10.1), we derived a CPG network in which the odd-numbered modules are only coupled to other odd-numbered modules (with nearest-neighbor coupling); the same was done for the even-numbered modules. In addition to this, the first two modules (m0 and m1) are coupled to define the difference of phase between odd and even chains (Figure 10.8). After convergence, the CPG behavior is equivalent to Equation 10.1.

In the most general case, the CPG network of a Lola-OP™ snake robot made of n modules has $3n-1$ open parameters to be tuned (n amplitudes, n offsets, and $n-1$ phase couplings). However, depending on the gait that we would like the robot to perform, this number can be dramatically reduced.

Locomotion Gaits

In this work we explored the effect of added in-series compliant elements on locomotion for two snake robot gaits: rolling and sidewinding.

> ***Rolling gait:*** The rolling gait is a highly efficient non-biologically inspired gait in which the robot takes the shape of an arc and then rolls sideways [217]. By applying the parameter reduction proposed in [123], the number of open parameters for a robot of any length n is reduced to 1, namely the amplitude, which is equal for all the modules. According to [123], the offsets are set to zero, ψ_{01}, between odd and even modules, fixed to $\pi/2 + \psi_{ij}/2$, and the phase differences between the other modules are set to zero. We considered a frequency $f = 0.5$ Hz for all modules.
>
> Given the low dimensionality of the search space, for this gait we optimized the amplitude for each type of compliant element with a grid search, testing nine amplitude values between 10 deg ($\frac{\pi}{18}$ rad) and 90 deg ($\frac{\pi}{2}$ rad) in steps of 10 deg. For each value, we ran 30 seconds of physical simulation in Webots while logging the speed of the robot and an

Table 10.3 – Open CPG parameters for the two gaits considered in this work. Rolling parameters have been tested using a grid search; we used Particle Swarm Optimization for sidewinding. All values are expressed in radians.

	Variable	**Range**	**Units**
Rolling gait	Amplitude A	$[\frac{\pi}{18}; \frac{\pi}{2}]$	[rad]
Sidewinding gait	Amplitude A_{odd}	$[0; \frac{\pi}{2}]$	[rad]
	Amplitude A_{even}	$[0; \frac{\pi}{2}]$	[rad]
	Phase lag ψ_{ij}	[1; 2]	[rad]

estimate of the energy consumption for the different motors. The energy consumption was estimated by multiplying the instantaneous rotation speed by the instantaneous generated torque. It should be noted that this is a rough estimation that can be affected by numerical errors. We repeated each experiment five times (each time the robot was initialized in a random initial orientation) and considered the average. A complete grid search was performed for the four terrains (FLAT, RT1, RT2, and RT3).

We performed a total of 1440 experiments (4 terrains, 8 stiffness levels, 9 amplitudes, and 5 repetitions for each configuration).

Sidewinding gait: Sidewinding is a biologically inspired gait used by some real snakes to locomote on sand. The gait is created by the combination of a horizontal sinusoidal wave with a vertical one, which creates a motion in which at any time the snake has a least two static contact points with the ground. It is highly energy efficient and fit for rough ground locomotion [217]. Applying the parameter reduction rules in [123] to this gait, we end up with three open parameters, independent in number from the number of modules n in the snake robot: a common amplitude for the odd numbered modules A_{odd}, a common amplitude for the even numbered modules A_{even}, and a constant shared phase difference ψ_{ij} (see Table 10.3). ψ_{01} is fixed to $\pi/4 + \psi_{ij}/2$ and again the offsets are set to zero. For sidewinding, we explored gaits at f = 0.5, 1, and 2 Hz.

To search through this parameter space we used PSO to find the best CPG parameters that lead to fast locomotion speed. The ranges for these three open CPG parameters have been chosen according to [123]. For the PSO, we used 10 particles and 150 iterations tested for 30 seconds of physical simulation in Webots. Each particle was tested three times (each time with a different random initial orientation) and we only considered the fitness value (average locomotion speed calculated as displacement divided by elapsed time) of the worst performing one (the other two were completely discarded). This was done assuming that there could be an “easier” direction in the rough terrain, or that

> the robot could “learn” how to overcome the roughness in a very specific direction; instead, we tried to favor more robust particles that perform well enough for any three random initial orientations. Given the four terrains and eight levels of stiffness, our PSO framework performed a total of 432000 experiments (4 terrains, 3 frequencies, 8 stiffness levels, 10 particles, 3 repetitions, 150 iterations).

Table 10.3 summarizes the open CPG parameters used in this work.

10.2.3 Experimental Results and Discussion

Rolling Gait

Following the experimental protocol described in the previous section, for the rolling gait we performed a grid search for each terrain. Figure 10.9 shows the results in terms of locomotion speed and cost of transport (CoT). Cost of transport is defined as $CoT = \frac{power}{(mass \cdot g \cdot speed)}$, where power and speed are evaluated in the simulation, the mass of the robot is set to 0.7 kg (corresponding to the weight of the robot without onboard battery and control board), and g is the standard gravity. We repeated each simulation five times and calculated the average.

For the rolling gait, it can be observed that there is a general trend of best performance for both speed and CoT for stiffer robots controlled with small amplitudes (between $\pi/18$ and $3\pi/18$ rad). This can be explained by the fact that the rolling gait produces a motion that can be almost approximated as wheel-like. The continuous (instead of intermittent) ground contact does not allow the compliant elements to store and release energy.

Sidewinding Gait

We performed 96 optimizations using PSO (one optimization for each of the eight stiffness values, repeated for three frequency values and four terrains). We considered the particle with the highest fitness value (highest locomotion speed) and replayed it 20 times to test the robustness over different initial orientations. Figures 10.10 and 10.11 show the trend of speed and cost of transport with stiffness for three locomotion frequencies (0.5, 1, and 2 Hz).

Contrary to the results for rolling gait, we notice that more compliant structures perform similarly to the stiff ones, sometimes outperforming them both in terms of speed and energy efficiency, e.g. for a stiffness of 50 $\mathrm{N\,m^{-1}}$ at 2 Hz on FLAT and RT1. By analyzing the behavior of the robot in simulation, it could be seen that stiffer robots are sometimes able to use the concave bumps in the rough terrain as fulcrum (i.e. the support point for the lever) to create large lever arms and push themselves forward.

It can be observed that the standard deviations of speed and CoT increase with the roughness in the terrain. One reason for this is the fact that the locomotion controller is running in open loop, and therefore there is no feedback to keep the trajectory of the robot on a straight line. As

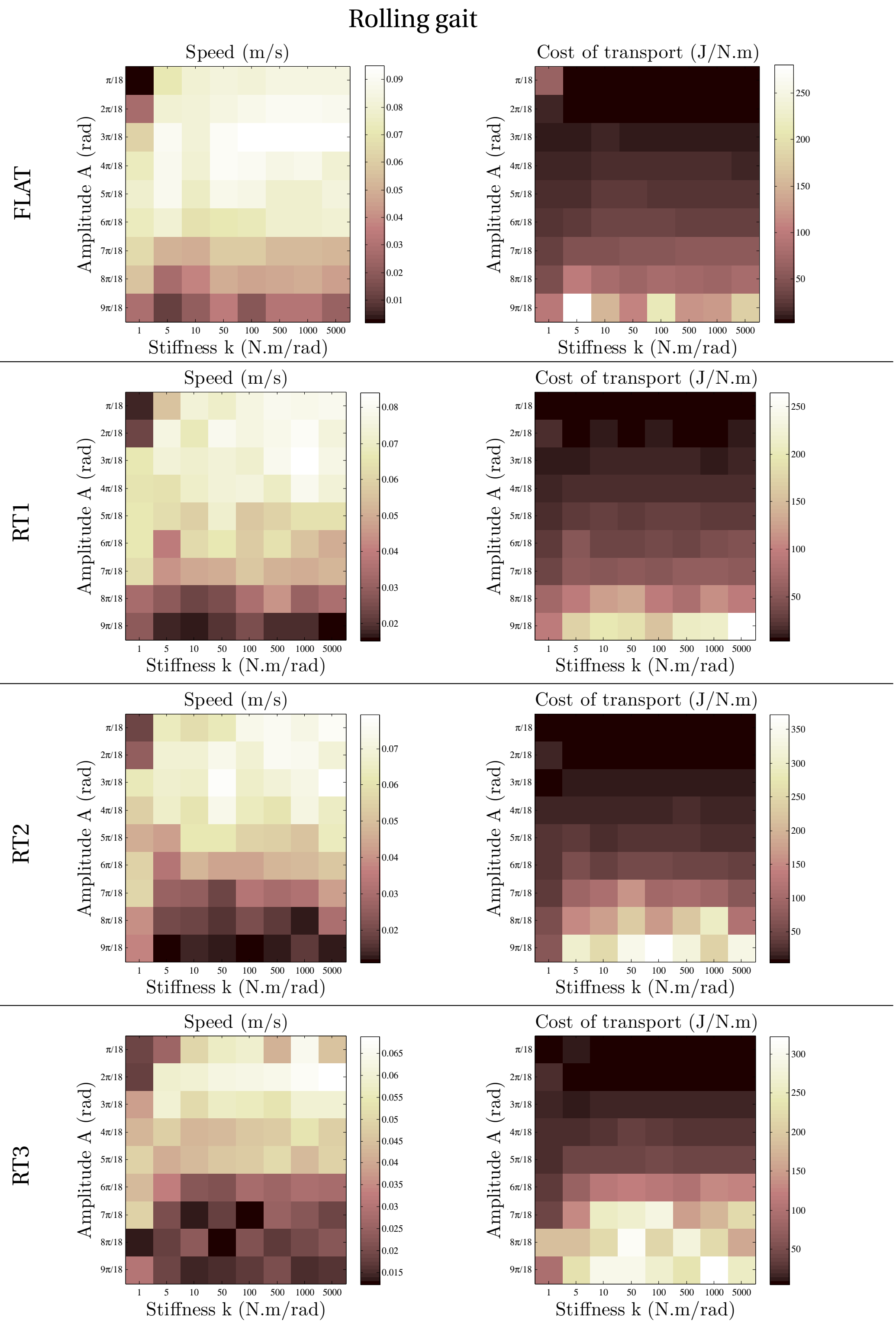
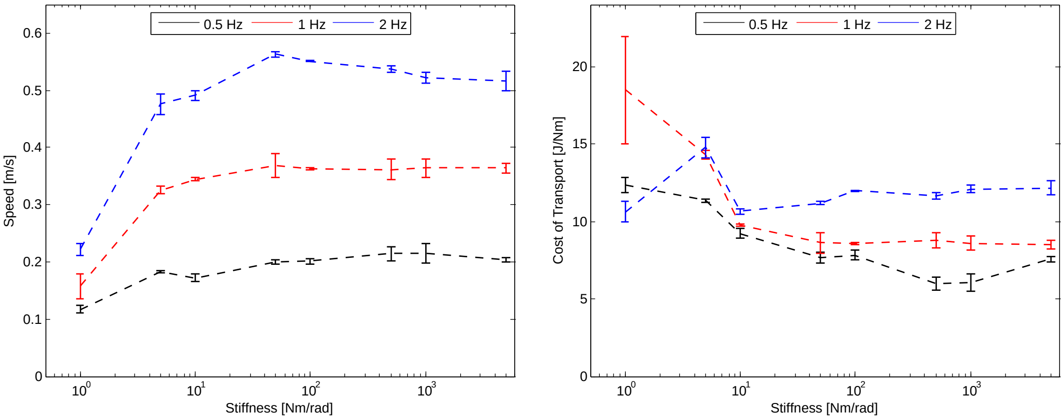


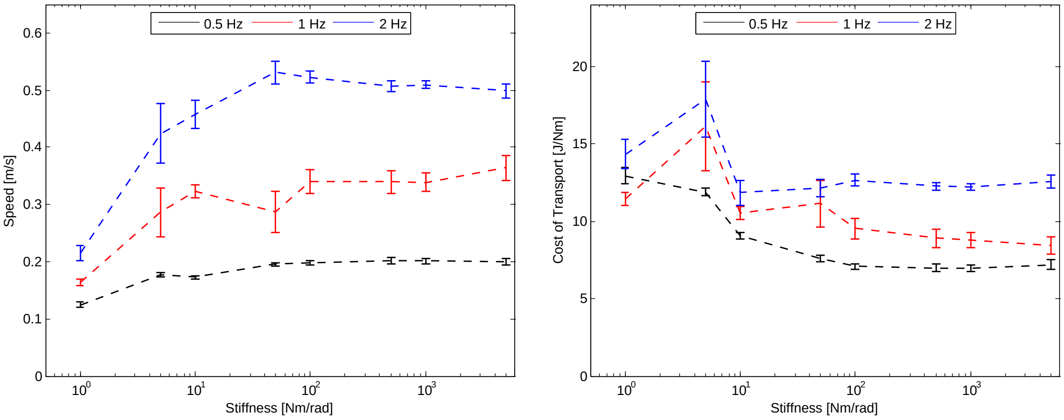
Figure 10.9 – Grid search for the rolling gait on the four terrains: (left) Speed. (right) Cost of Transport. Each value is the average of five repetitions of the same experiment (with random initial rotation). Images adapted from [226].

Sidewinding gait



(a) Speed on FLAT

(b) CoT on FLAT

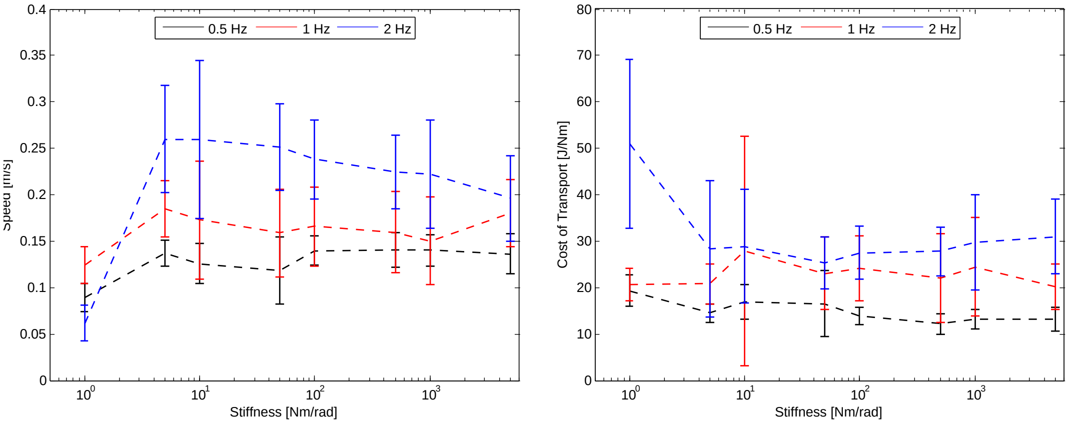


(c) Speed on RT1

(d) CoT on RT1

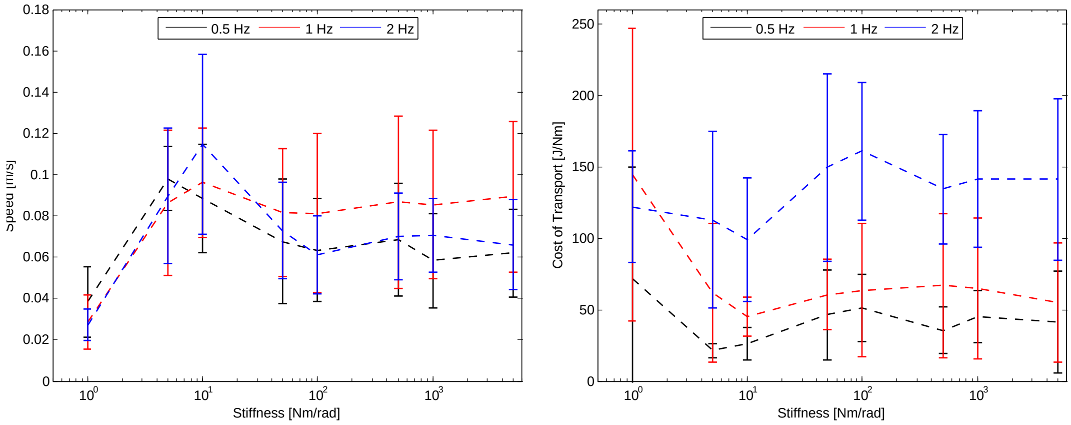
Figure 10.10 – Speed and CoT results for sidewinding gait on FLAT and RT1. Each point is the average of 20 repetitions of the best particle from each PSO optimization. Images adapted from [226].

Sidewinding gait



(a) Speed on RT2

(b) CoT on RT2



(c) Speed on RT3

(d) CoT on RT3

Figure 10.11 – Speed and CoT results for sidewinding gait on RT2 and RT3. Each point is the average of 20 repetitions of the best particle from each PSO optimization. Images adapted from [226].

Table 10.4 – Hardware runs of the Lola-OP™ robot on flat ground. Each experiment was repeated 5 times.

	Frequency $[Hz]$	**Stiffness** $[Nmrad^{-1}]$	**Mean speed** $[m/s]$	**Std. deviation** [ ]
Sidewinding gait	0.5	3	0.1363	0.0010
	0.5	70	0.1309	0.0105
	1.0	3	0.2286	0.0080
	1.0	70	0.3338	0.0109
Rolling gait	0.5	3	0.0783	0.0018
	0.5	70	0.0835	0.0011

a result, the unevenness in the terrain sometimes causes the robot to move in a circle, resulting in a small displacement and thus a low locomotion speed.

Hardware Experiments with 8-DoF Lola-OP™ Robot

We validated the simulated experiments with a series of hardware runs on flat ground. For this work we restricted the hardware experiments to two levels of compliance: *"soft"*, with an estimated stiffness of ~ 3 Nm rad^{-1}, and *"stiff"*, with a stiffness of ~ 70 Nm rad^{-1}. For each gait, frequency and level of stiffness, we used the best parameter set obtained from simulation. We repeated each run 5 times and calculated the speed as displacement (measured with a tape measure) over time (20 seconds for 0.5 Hz gaits, 10 seconds for 1 Hz gaits). Results for sidewinding and rolling gaits are reported in Table 10.4. We omitted the results for 2 Hz as this frequency exceeded the real capabilities of the hardware robot and generally resulted in inefficient gaits. As can be seen in Figure 10.12 for sidewinding, and comparing Table 10.4 with the results for FLAT in Figure 10.9 for rolling, there is a reality gap between simulation and hardware. Nevertheless, the trend of this limited dataset seems to match simulation. Snapshots from representative runs are shown in Figures 10.13 and 10.14.

10.2.4 Discussion

In this section we analyzed in simulation the effect of changing the stiffness value of a compliant modular snake robot while performing rolling and sidewinding gaits over different types of terrains and frequencies. Compared to the results of Section 10.1, in this study we did not find a clear advantage using a compliant modular robot instead of a stiff one.

For the rolling gait, the apparent reason is that the robot is creating a sort of wheel-like locomotion that does not allow any storage and release of energy in the compliant elements.

On the other hand, for sidewinding gait we have a plateau, with minor variation of speed and CoT for a large range of stiffness values and some performance peaks at higher frequency and intermediate stiffness. Although these results do not clearly show an optimal “general purpose” level of compliance (that could be chosen e.g. around 50-100 Nm rad^{-1}), it is still interesting to note that relatively soft Lola-OP™ robots can locomote with a speed and CoT comparable to stiffer ones.

Frequency does not seem to have a major effect on the overall behavior. We reported some performance peaks at 2 Hz, although this frequency is above the real capabilities of the hardware robot.

Perhaps the major benefits of compliance are not achieving faster locomotion speed, but in helping to cross features that offer fewer points of contact (e.g. moving along a cylinder or terrains with big sine-like waves), where intuitively compliance would help provide a better match to the terrain. In the next section we will present the results of a follow work done on horizontal pipes with the hardware robot.

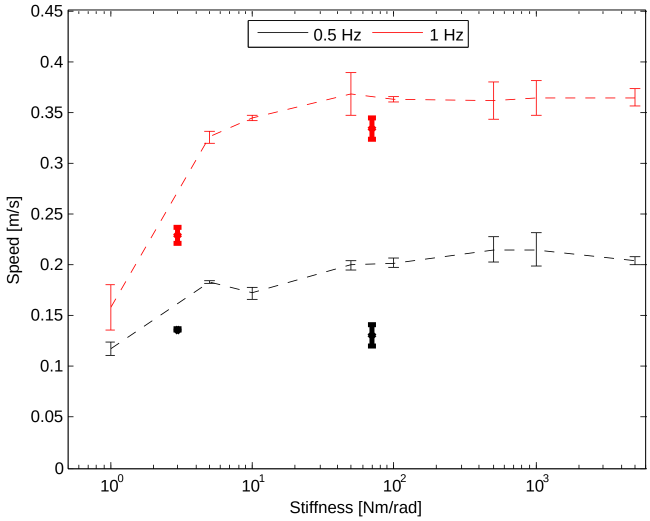


Figure 10.12 – Validation of optimization results: speed of the hardware robot sidewinding gaits on flat ground (in bold) compared to simulation runs on FLAT. Image adapted from [226].

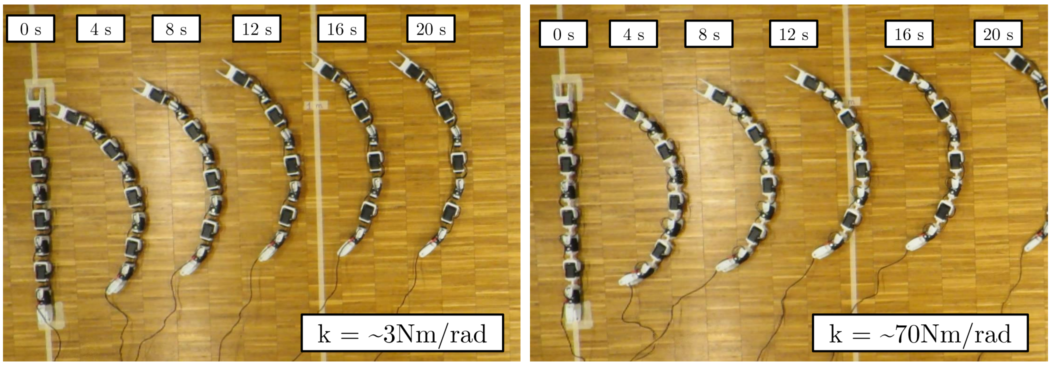


Figure 10.13 – Snapshots from hardware experiments. Rolling gait with *“soft”* elements (left) and *“stiff”* elements (right). Locomotion frequency is 0.5 Hz. Vertical lines are 1 m apart. Images from [226].

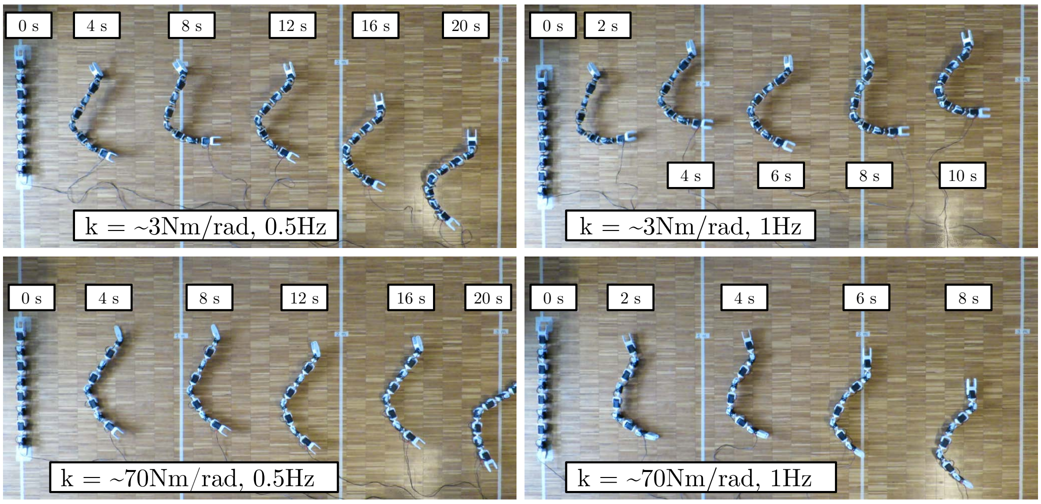


Figure 10.14 – Snapshots from hardware experiments. Sidewinding gait with *“soft”* (top) and *“stiff”* elements (bottom) at 0.5 Hz (left) and 1 Hz (right). Vertical lines are 1 m apart. Images from [226].

10.3 Lola-OP™ Locomotion on Horizontal Pipes

An interesting application for snake robots are exploration tasks in disaster scenarios [139, 119]. These robots are capable of producing motion in terrains in which the unevenness is comparable to their own size and, given their small cross section, they are ideal for carrying and bringing sensors into narrow gaps in the rubble. We believe that exploring snake robot locomotion on horizontal cylindrical surfaces is a good starting point for understanding how compliance can help to cross complex features with few contact points. Results and locomotion strategies can then be extended to more complex terrains, including pipes of different sizes, tree branches, bushes [120], geological formations, and in general uneven surfaces or cylindrical/conical shapes that can be found in disaster zones.

For these reasons we extended the studies on the effect of added in-series compliance, presented in Section 10.2, to the particular case of locomotion on horizontal cylindrical terrains (Figure 10.15). We built four versions of the 8-DoF Lola-OP™ robot with different levels of stiffness in their joints (Figure 10.16). We analyzed their behavior in terms of speed and power consumption while performing rolling motion on three types of horizontal pipe environments.

This work is guided by the following hypotheses:

1. A robot with compliant elements can better adapt (in terms of increased locomotion speed, and reduced motor load and power consumption) to complex features compared to a stiff one.

2. The use of compliant elements can help relax the importance of well-tuned control parameters.

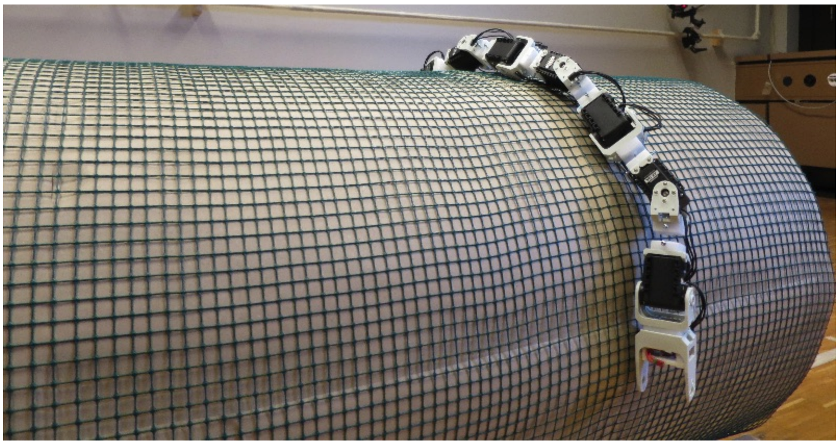


Figure 10.15 – Compliant 8-DoF Lola-OP™ robot locomotion on horizontal pipe environment.

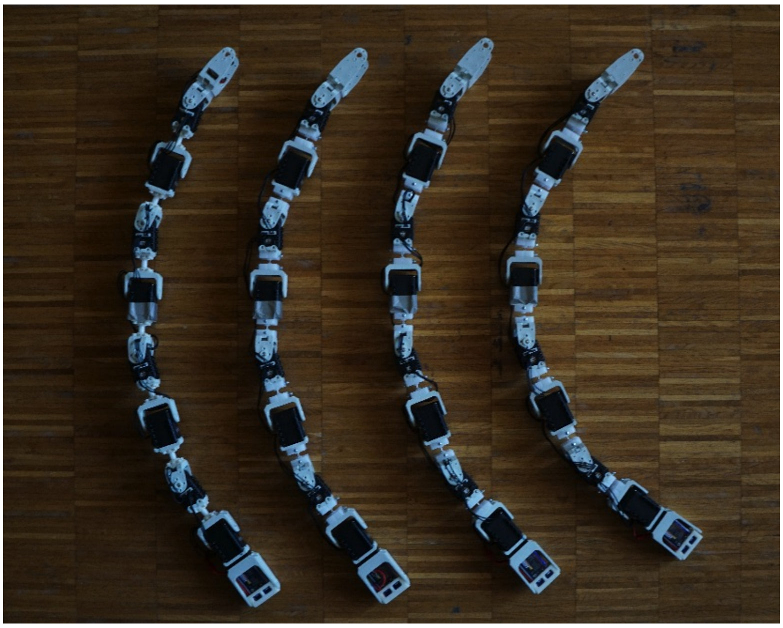


Figure 10.16 – Hardware implementation of the Lola-OP™ robot with eight degrees of freedom and in-series compliant elements. Robots are shown from stiff (left) to compliant (right). Image from [225].

10.3.1 Related Work

Achieving sustained locomotion with modular snake robots in complex terrains terrains is not as simple as observed in flat ground [217, 123]. One gait realization that stands out other possible gaits due to its simplicity is the *rolling* motion. This gait has been widely studied by several authors using different approaches that include robot's motion estimation [168], capture of robot's overall heading [173], and also modeling of the gait shape using simple and intuitive composition of shape projections [262]. In particular, these arc-shaped gaits (easily extended to helix shapes) result convenient for locomotion on cylindrical surfaces as reported in the literature for horizontal motions [150, 119, 51], as well as in vertical/obliquous directions (i.e. climbing a pole [7, 171]).

Previous experiments on modular snake robot gaits on horizontal pipes [119] show that there is an important motion constraint in this setting: since the rolling arc shape is controlled by a single value of amplitude for all joints, the shape tends to remain invariant during the whole execution of the gait [118]. This can make the robot fail when facing sudden variations in diameter or changes in the round smoothness of the pipe (e.g. bumps in tree branches). For this reason we believe that a robot with in-series compliant elements can provide an underactuated method for adapting the robot's shape to the environment while using just a simple open-loop controller.

Terrain adaptability of snake robots using compliant elements has been fairly studied during the last years. Liljeback *et al.* [106] proposed a compliant control of the shape of snake robots while moving in cluttered terrains with obstacle aided locomotion using shape control points as motion planner. The mentioned compliant control replaces rigid robot-environment contacts with mass-spring-damper dynamics. Our approach differs significantly from this work because (i) we use a specific gait (i.e. rolling) on cylindrical surfaces, and (ii) our robots feature built-in compliant dynamics, instead of compliant contact dynamics.

Additionally, Choset *et al.* at CMU explored a rigid body robot controlled by a gait-based compliant control which acts on the overall shape of the robot instead of each joint [169]. However, the gait-based compliant control is based on an expensive computation of the robot state using an extended Kalman filer (EKF). The controller compares current state (affected by the interaction with the environment) with the commanded one (gait control parameters updated once they are fitted) in two fashions: (i) curvature compliance (i.e. controlling gait's amplitude), and (ii) position compliance (i.e. controlling the gait's temporal position). With this controller the snake robot is able to climb poles with changing diameters. The controller effectively drives the robot by changing gait parameters to compliantly adapt its shape according to the environment. However it is computationally expensive and requires a robot capable of sensing its full state each time step.

On the other hand, adding series elastic actuators to the structure of the snake robot [170] allows for more elaborated and reliable torque control strategies [171]. In this last work, the authors directly control actuators torques based on three different strategies. One of these strategies ("compliant roll-in-shape") is directly related to the problem of moving snake robots with rolling gaits on uneven surfaces. It operates by commanding the joint torques based only on a linear mapping from the local curvature (which is based on the joint angles). This strategy can be easily extended to other types of compliant snake robots, including one based on variable stiffness actuators (VSA) [121], in which position and stiffness value of each joint can be separately controlled.

10.3.2 Materials and Methods

For this work we used the hardware implementation of the Lola-OP™ robot presented in Section 2.3. The Dynamixel actuators are controlled by an Arduino Pro Mini board located in one of the plastic structural frames at the end of the robot (that we identified as head), using a free communication library developed by Savage Electronics [6]. An HC-05 Bluetooth module provides a two-directional communication channel for starting/stopping the robot, and for changing the locomotion parameters. A battery located in the tail module (i.e. the other end of the robot) provides power to the actuators and control unit for approximately 30 minutes of continuous operation. We built four copies of the robot using the four types of in-series compliant elements presented in Section 6.3.2.

Pipes

We chose cylindrical horizontal pipes as locomotion terrain as they represent features that can be easily found in industrial settings (e.g. oil pipes, chimneys, which are a niche for robot inspection tasks), or that can be generalized to other possible locomotion structures that can serve as surveillance posts (e.g. tree branches).

We tested the on-pipe locomotion on three 1m-long straight pipes (P1, P2, and P3). The diameter of each pipe was chosen in relationship with the robot length (0.86 m), in order to have the following conditions, similarly to [119]:

- P1 (small pipe), with a circumference approximately equal to half the robot's length. The robot theoretically[1] wraps twice around the pipe.
- P2 (medium pipe), with a circumference approximately equal to the robot's length. The robot wraps around the pipe, with head and tail almost touching.

---

[1] In practice, the reduced number of degrees of freedom in the 8-DoF Lola-OP™ robot (only 4 DoF for each motion plane) causes an under-sampling behavior in the reconstruction of the round shape for a small pipe. The robot effectively wraps the small pipe slightly more than one time. This phenomenon is explained in [150].

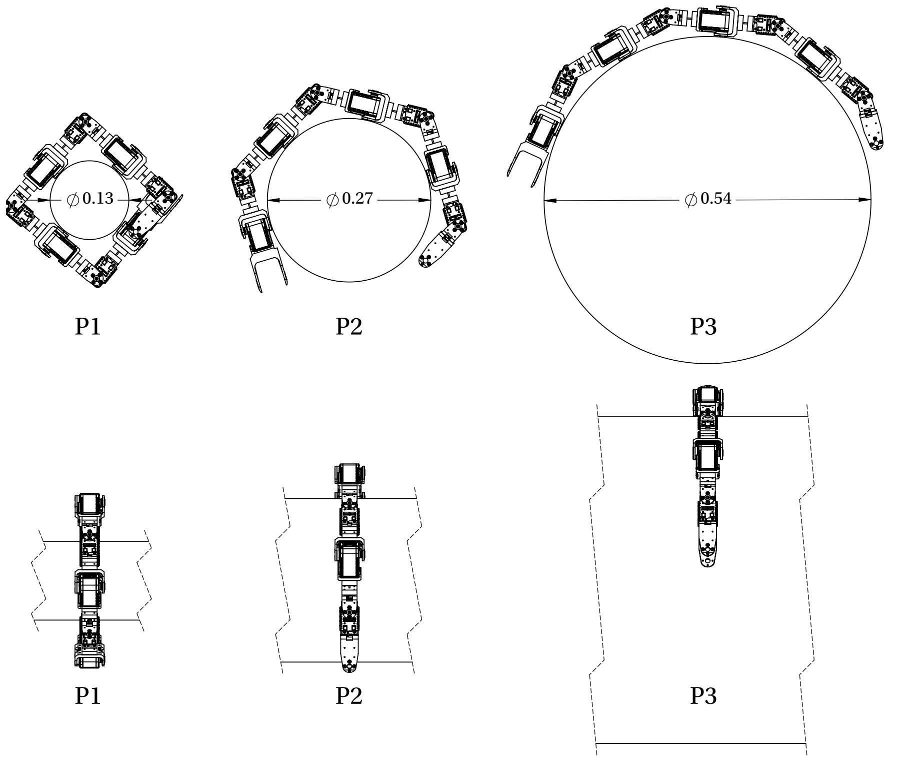


Figure 10.17 – Front and lateral views of the three pipes used to study the effect of in-series compliance on snake robot locomotion. 8-DoF Lola-OP™ robots shown for size comparisons. Images from [225].

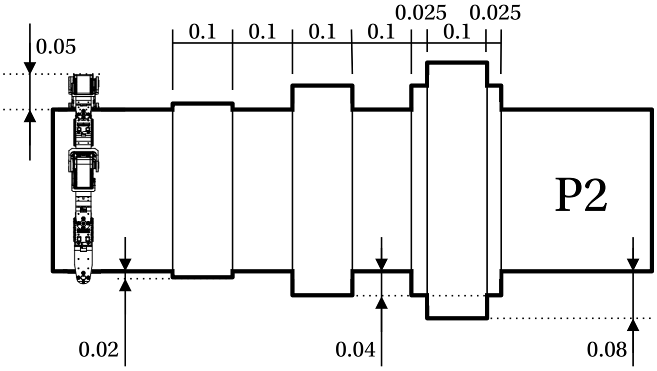


Figure 10.18 – Diametrical rough terrain, obstacle, and continuous obstacle (as defined in [143]) added to the medium pipe (P2). Dimensions are in meters. Image from [225].

- P3 (large pipe), with a circumference approximately equal to double the robot's length. The pipe can almost be seen as a large bump. The robot spans over the top hemicircumference.

Figure 10.17 gives an illustration of these three cases. Considering that the robot lies on a circumference that has a diameter equal to the sum of pipe diameter and the robot's cross section (h_{robot} = 0.05 m), we chose the following pipe diameters to fulfill the aforementioned desired conditions: D_{P1} = 0.13 m, D_{P2} = 0.27 m, and D_{P3} = 0.54 m.

The three pipes were manufactured using a combination of plastic pipes (for P1), cardboard, and plastic fencing meshes. Externally, we wrapped the pipes with kraft paper and then added a thin layer of plastic netting (with a square mesh size of 12 mm) to provide more friction.

To study the effect of compliance when crossing perturbations we created diametrical bumps on the medium pipe (P2). We chose the height (increase in pipe diameter) of these bumps according to the definitions of *rough terrain*, *obstacle*, and *continuous obstacle* proposed in [143]. Figure 10.18 shows the dimensions of these bumps.

Finally, in order to demonstrate how the same control parameters can adapt to large changes in diameter, we connected together the three pipes as shown in Figure 10.19. This terrain has

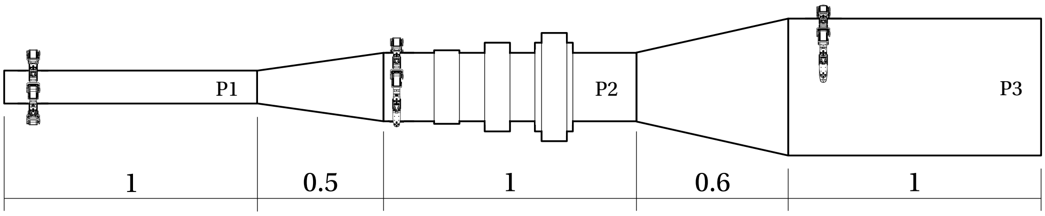


Figure 10.19 – CAD dimensions of the complete terrain used to show the adaptability of the robots to changes in pipe diameter. Dimensions are in meters. Image from [225].

the following features:

- A fourfold increase in diameter
- “Rough terrain", “obstacle", and “continuous obstacle" features
- “Slope" features (as also defined in [143]) while transitioning between the different pipes

Control Framework

For the locomotion over horizontal pipes we used the equations of motion of the Lola-OP™ robot, as defined in Equation 10.1, to produce rolling gait patterns.

By applying the parameter reduction proposed in [123], we fixed eight of the nine open parameters of Equation 10.1: the offsets ($O_{o,e}$) are set to zero; δ, the phase difference between odd and even modules, is fixed to $\pi/2$; the wavelengths $\lambda_{o,e}$ are set to infinity (thus the terms $1/\lambda_{o,e}$ are set to zero); we used a common frequency $f = 0.5$ Hz for both odd and even modules (with $\omega_{o,e} = 2\pi f$).

The only open variables are the amplitudes, that are however set to the same value ($A_o = A_e = A$), thus reducing the number of open parameters to just one, gaining a considerable reduction in the controlling space of the robot.

The rolling gait adapts very well to the problem of pipe locomotion [119]: the robot takes the shape of an arc whose radius is inversely proportional to the amplitude value used in the locomotion controller. Let's consider that at time $t = 0$, $\theta(odd, 0) = 0$ and $\theta(even, 0) = A$; as a good estimation, the robot arc spans over $4 \cdot A$ degrees (because there are 4 even modules; in this work we always measured A in degrees) and the radius is given by the robot length ($L_{robot} = 0.86$ m) divided by $4 \cdot A \cdot \frac{\pi}{180}$. It is important to note that this radius corresponds to the arc passing through the center line of the robot.

In order to calculate the amplitude for the snake robot to wrap following the curvature of a pipe of known diameter D, we also need to consider its cross section ($h_{robot} = 0.05$ m):

$$A = \frac{1}{4} \frac{L_{robot}}{\frac{(D + h_{robot})}{2}} \frac{180}{\pi} = \frac{0.86}{(D + 0.05)} \frac{90}{\pi} \tag{10.5}$$

Experimental Protocol

Using Equation 10.5 it is possible to calculate the locomotion controller amplitude for any given pipe diameter. However, there is no guarantee that this amplitude value is the optimal to allow fast locomotion (for instance, a robot that holds tighter on the pipe surface might generate just the right amount of friction to avoid any slipping). Moreover we are interested in checking the robustness of the different compliant robots to changes in pipe diameter (and thus indirectly to changes in locomotion amplitude). Finally, Equation 10.5 is an approximation that suffers from the limited number of degrees of freedom in the system.

For these reasons, in the first part of this work we optimized the amplitude for each type of compliant element and terrain with a grid search, testing several amplitude values at intervals of 5 degrees around the theoretical amplitude calculated with Equation 10.5 for each pipe size.

For each value, we let the physical robot run over each pipe and, after an initialization time, recorded 3 seconds of motion capture data (MoCap). We wrapped a 25mm-wide reflective tape around the approximate center of mass (CoM) of the robot, and tracked this point in space (3 DoF) using 14 Optitrack s250e cameras. Data were recorded at 240 frames per second using the Optitrack Motive software and streamed to a custom program that automatically calculated the speed of locomotion as displacement of the CoM divided by trial time (3s). We repeated each run 5 times to evaluate repeatability.

For the second part of this work we evaluated speed and power consumption for the locomotion over bumps on P2. For the evaluation of the speed we used the aforementioned setup, extending the trial time to the necessary duration to allow overcoming all the bumps. To measure power consumption, we tethered the robot to a Voltcraft HPS-13015 power supply (set to 12 V) and used a current probe clamped around the power cable (complete experimental setup shown in Figure 10.20). The Tektronix TM502A amplifier output was sampled with a LeCroy 6100 digital oscilloscope and low-pass filtered in Matlab.

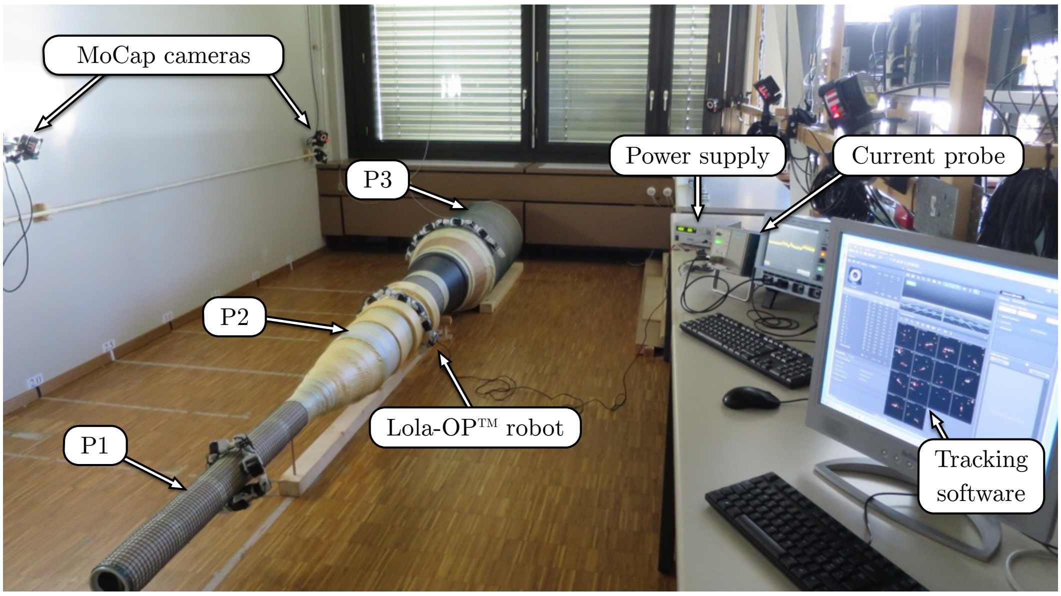


Figure 10.20 – Experimental setup equipped with 14 infrared motion capture cameras (MoCap) and a current probe and digital oscilloscope for measuring the robot's power consumption.

For this set of experiments we fixed the amplitude to $A = 77$, calculated using the P2 diameter (0.27 m) in Equation 10.5.

Lastly, for the final demonstration we recorded videos of the experimental trials to show qualitative results. In this case we used a fixed amplitude $A = 70$, which seemed a good tradeoff for locomotion on small, medium, and large pipes.

10.3.3 Experimental Results

Locomotion Speed on Straight Pipes

For each of the three pipes and each of the four compliance levels, we performed a grid search to evaluate how the amplitude A affects locomotion speed. For P1 the theoretical value calculated with Equation 10.5 (137 deg) goes above the possible range of motion of the robot's actuated DoF (105 deg). Therefore we tested 7 values, from 75 degrees to the maximum range (Figure 10.21).

The speed of the robots increases as the amplitude value approaches the theoretical one (the robot holds tighter around the pipe). Interestingly different compliance levels, with the exception of C1, produce comparable speeds. The softest elements prove to be inefficient for P1 because they deform too much and therefore the robot does not have enough contact

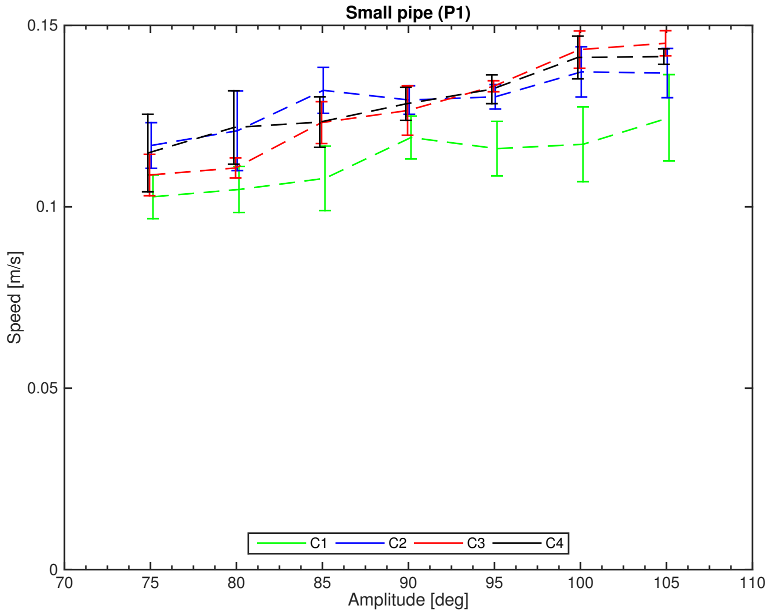


Figure 10.21 – Locomotion speed for 7 values of amplitude, for each level of compliance on the small pipe (P1). 105 degrees is the maximum range allowed by the actuators. Each parameter was tested 5 times. Image from [225].

Table 10.5 – Specifications of the experimental setup used for the locomotion experiments on horizontal pipes.

Compliant Lola-OP™ robot	
Overall dimensions	0.86 × 0.05 × 0.05 m
Weight	790 g (incl. battery)
Actuated DoF	8: Dynamixel AX-12A®
Number of compliant elements	7; four different levels of compliance
Levels of compliance [Nm/rad]	$k_{C1,F} = 0.46$ $k_{C1,T} = 0.35$
(F = Flexural stiffness)	$k_{C2,F} = 2.3$ $k_{C2,T} = 1.75$
(T = Torsional stiffness)	$k_{C3,F} = 7.3$ $k_{C3,T} = 5.54$
	$k_{C4,F} = 70.4$ $k_{C4,T} = 52.3$
Control board	Arduino Pro Mini, ATmega 328, 5V
Communication	HC-05 Bluetooth board
Energy source	3-cell LiPo Battery, 450 mAh (Autonomy ~30 minutes) or tethered 12V Voltcraft HPS-13015 power supply
Terrains	
Number and type	3 Horizontal pipes
Type of surface	Rugged plastic netting (mesh size 12 mm)
Diameter of pipes [m]	$P_1 = 0.13$, $P_2 = 0.27$, $P_3 = 0.54$
Control parameters	
Gait	Rolling
Frequency of locomotion	0.5 Hz
Open parameter	Amplitude of oscillation
Fitness evaluation	
Parameters	Speed = displacement / trial time Power consumption = voltage·current

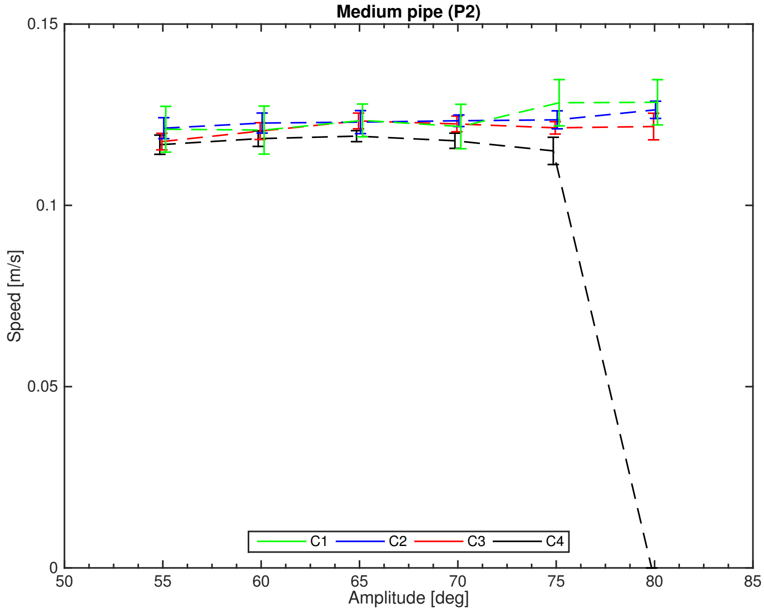


Figure 10.22 – Locomotion speed for 6 values of amplitude, for each level of compliance on the medium pipe (P2). The stiff robot is not able to run with amplitude values greater than 75 degrees. Each parameter was tested 5 times. Image from [225].

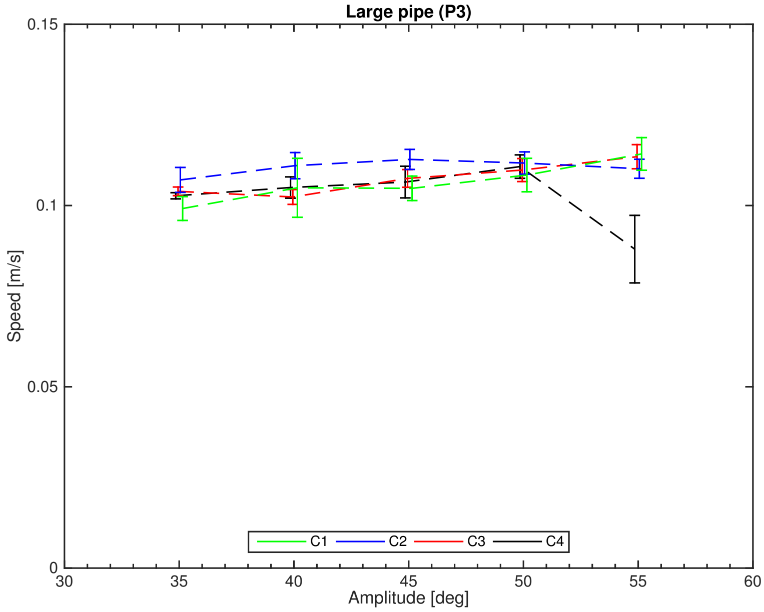


Figure 10.23 – Locomotion speed for 5 values of amplitude, for each level of compliance on the large pipe (P3). Each parameter was tested 5 times. Image from [225].

points with the pipe.

Figure 10.22 shows the results for the medium pipe (P2). In this case, the compliant robots could effortlessly run with amplitude values above the theoretical one (77 degrees). Instead, for the stiff configuration (C4), actuators were already heating up above normal levels for $A = 75$, and shutting down (temperature/torque protection) for $A = 80$.

We reported similar behavior also for the large pipe (Figure 10.23): over a certain amplitude value the stiff robot cannot passively adapt its arc to the pipe size, the motor torque increases, and the safety temperature/torque protection of the Dynamixel AX-12A® gets activated.

These results show that, when the robot is loosely holding to the different pipes (amplitudes not much above the theoretical ones), there is not a significant difference in speed between compliant and stiff configurations (with the exception of C1 on P1). However, as soon as the robots start holding tighter, the adaptability of the compliant elements becomes clear. These results suggest that we should also consider required torque/power consumption in the gait evaluation metric.

Power Consumption over Obstacles on Medium Pipe

For the second set of experiments we recorded speed of locomotion and power consumption of the robot. We only ran experiments on the medium pipe because it is the one that allowed the most stable gaits during the straight pipe tests. Diametrical bumps were added to P2, as described in Subsection 10.3.2.

For the control parameters, we used the theoretical amplitude ($A = 77$) calculated with Equation 10.5.

As seen in Figure 10.24, while the robots have comparable speed over time, the energy consumption of the stiff one is more than double compared to C2 and C3. This confirms our intuition that stiffer robots cannot effortlessly adapt to different terrains but instead will generate higher torques.

On the other hand, C1 is too compliant and did not manage to successfully complete the task. As stated in Section 6.3.2, C1 is a lower bound where compliance is too high and cannot be beneficial.

Demonstrator

We tested the robot on the pipe terrain shown in Figure 10.19. While the softest robot (C1) could not overcome the slopes and the bumps, all the other robots managed to successfully cross the whole structure. The same control parameters adapted to fourfold changes in pipe diameter (at times, we paused the robot and manually realigned it on top of the pipe). As reported also in the other experiments, the stiff robot clearly had an higher current consumption and one of

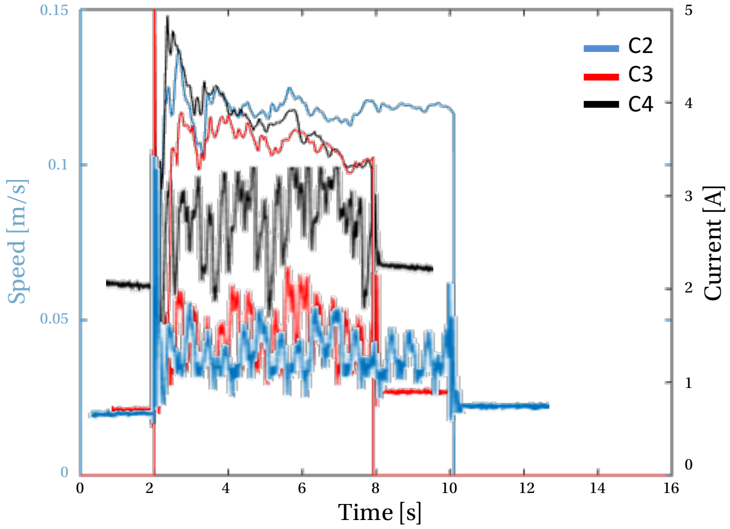


Figure 10.24 – Speed of locomotion (thin lines) and current (tick lines) over time for three compliant robots while crossing the pipe P2 with bumps. C1 is not reported because the robot could not cross the continuous obstacle. Image from [225].

the motors went in secure mode just at the end of the run. Figure 10.25 shows snapshots of a successful run with C3 compliant elements.

10.3.4 Discussion

We presented a study on the effect of passive in-series compliant elements on modular 8-DoF snake robots rolling on horizontal pipes. Our method differs from the state of the art because we are trying to ease the importance of well-tuned control parameters by letting the mechanical system do part of the work. Obviously, extra control layers such as feedback based on current and torque measurement can be added on top of this (not included in this work), but we are interested in a system that is inherently passively adapting to changes in the environment.

Rollinson & Choset [169] use an expensive computational algorithm that estimates the current state of the robot and modifies the equations of motion in order to provide adaptability of the robot to the terrain. This allows their robot to successfully climb up poles with changing diameter. We voluntarily avoided using a complex algorithm to control our robots. Moreover, our current robotic framework has limited internal sensing capabilities and cannot ensure a constant torque (e.g. to hold in place). However with further analysis we could study and implement a controller of whole-body shape compliance by modeling the dynamics of the in-series compliant elements. With our approach we cannot yet climb up vertical poles, but we can successfully locomote on horizontal pipes of changing diameter without changing control parameters.

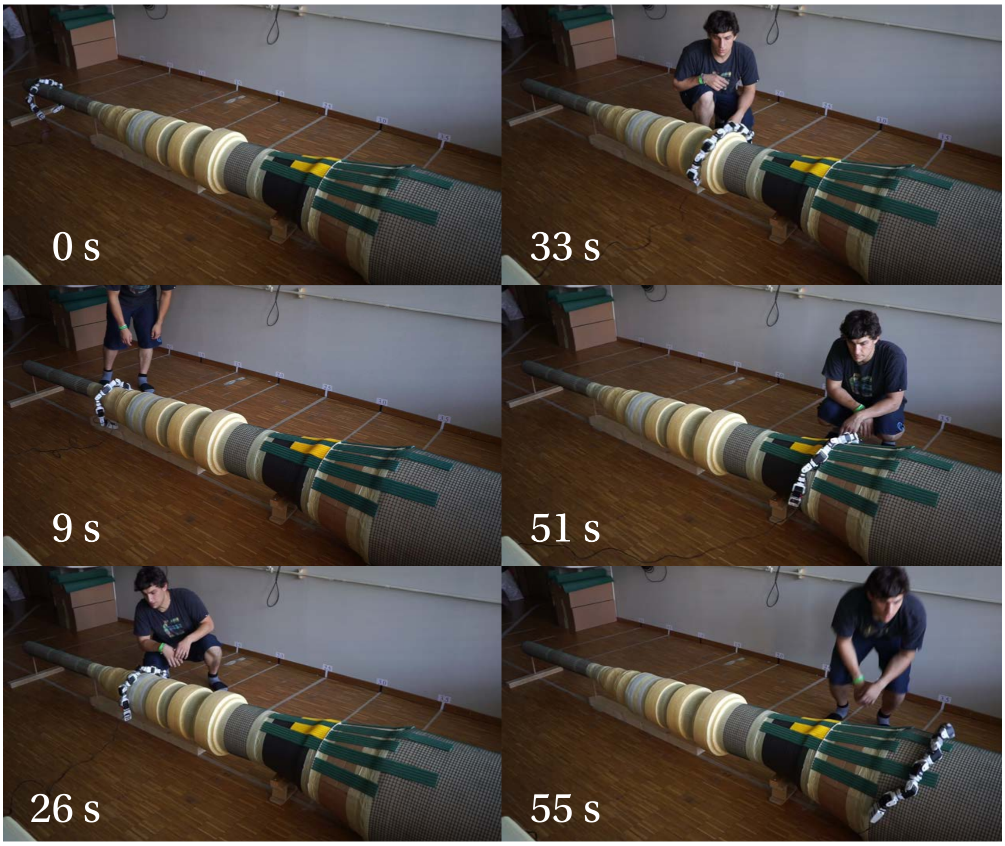


Figure 10.25 – Snapshots of a C3 compliant robot successfully crossing all the features of our pipe terrain. Images from [225].

Rollinson *et al.* [171] created a SEA Snake with compliance in the joints. This allows them to do torque control with a linear mapping from joint to torque. In our approach, we changed the compliance of the links between modules (shape compliance). The disadvantage of our system is that it is under-actuated (the dynamics explode). However our controller remains simple, the robot is cheap and lightweight, giving us the possibility (in terms of cost and payload capabilities) of adding sensors and devices that are useful for the safety and security mission.

We showed that compliant elements can increase the adaptability of a snake robot to changes in diameter of horizontal pipes (Hypothesis 1). While locomotion speed does not vary significantly, we have shown that power consumption for the stiff elements (C4) is more than doubled compared to softer elements (C2 and C3). This happens when the pipe diameter is larger than the robot's arc shape defined by the amplitude. In softer robots, the compliant elements can deform and the snake robot can passively adapt to the pipe size. Instead in the stiff robot the motors use more power and heat up to the point of triggering the embedded safety protection (Hypothesis 2).

10.4 Conclusions

In this chapter we analyzed the effects of in-series passive compliant elements on locomotion. To achieve this goal we conducted three different studies.

In the first study we used two types of in-series compliant elements (rotational and omnidirectional) connected between two Roombots modules to explore: (i) whether compliance could improve locomotion performance in terms of speed; and, (ii) if locomotion parameters optimized for an intermediate value of compliance could also be used to efficiently control stiffer or softer structures. We tested five values of stiffness for each of the two elements and ran online locomotion-parameter optimization for six different configurations. Results show a clear increase in performance (up to 65%) after the introduction of in-series compliance. The optimization process generated quite different gaits depending on the stiffness value of the structure. Sometimes these gaits are not robust to changes in stiffness.

In the second study we extended the previous approach and hypotheses to the locomotion of compliant 8-DoF Lola-OP™ Modular Snake Robots. We carried out this study in simulation, using offline optimization, and explored the effect of compliance for rolling and sidewinding gaits. We recreated a model of the robot and simulated in-series omnidirectional compliant elements with virtual compliant ball joints. Compared to the results that we obtained with the two Roombots modules, in this study we did not find a clear advantage of using a compliant configuration instead of a stiff one but instead we observed a rather similar behavior over different values of stiffness.

For the third study we considered again the compliant 8-DoF Lola-OP™ Modular Snake Robot but this time we performed our experiments using the hardware robot and exploring the specific case of horizontal pipe locomotion. The hypotheses for this work were: (i) that the major benefits of compliance are not for achieving fast ground locomotion but to help cross complex features that offer fewer points of contact; and, (ii) that compliance can ease the need for well-tuned control parameters by letting the mechanical system do part of the work. While also in this scenario locomotion speed is similar at different levels of stiffness, the compliant robots adapt better to different terrains, especially in terms of power consumption. Stiff robots require more than double the power compared to softer ones, which often caused the robots to heat up and engage their safety shut-down feature.

This concludes our contribution to the locomotion challenges for SRMRs. In the next part we will propose a new application of Roombots as building blocks for furniture and will explore intuitive ways to interact with modular robots.

Applications Part IV

11 Towards Self-Reconfigurable Robotic Furniture

Despite the promised advantages of Self-Reconfigurable Modular Robots compared to conventional robots there is still the challenge of finding the optimal application that can exploit these advantages. We propose the futuristic idea of using Self-Reconfigurable Modular Robots as building blocks for Self-Reconfigurable Robotic Furniture that can move, morph, and create new features for household activities in order to improve the quality of life.

As a first step towards the realization of this visionary scenario in Section 11.1 we propose the use of SRMRs' modules as Plug-n-Play Robotic Elements. Robotic modules can attach to passive connectors embedded in standard pieces of furniture or in the domestic environment to be used as legs, wheels, or manipulators.

In Section 11.2 we report examples of the prospective use of Plug-n-Play Robotic Elements. In these preliminary demonstrations we use Roombots modules to provide passive objects with locomotion capabilities and to manipulate dedicated attachments.

11.1 Introduction

As discussed in Chapter 1, one of the main challenges facing SRMRs is that of determining possible application scenarios. We propose to merge ideas from the fields of self-reconfigurable modular robotics, assistive robotics, and domotics in order to create adaptive *Self-Reconfigurable Modular Furniture* (SRMF) that moves, self-assembles, and creates new functionalities in a household in order to improve the quality of life. The ultimate vision for SRMF is on-demand furniture, in which groups of robotic modules autonomously connect to each other and to passive parts to form different types of furniture, e.g. stools, chairs, sofas, and tables. This furniture can change shape over time, move to different locations, and adapt to the user's needs, as presented in Figure 2.4 in Chapter 2.

As a first step toward this long-term vision in this chapter we propose the use of SRMRs' modules as *Plug-n-Play Robotic Elements* that can be used to augment existing furniture and create new smart features at disposal of the household. Figure 11.1 shows an illustration of such scenario in which the robotic modules interact with all the members of a family. Standard pieces of furniture such as wooden tables or sofas can be equipped with passive connection ports to which Plug-n-Play Robotic Elements can connect. These active robotic modules can be used to provide standard pieces of furniture with locomotion capabilities. In this scenario we imagine passive connection ports also distributed in the environment, embedded in the walls and ceiling. Plug-n-Play Robotic Elements can be manually attached or use lattice locomotion to connect to these environmental connectors. Once attached, they can

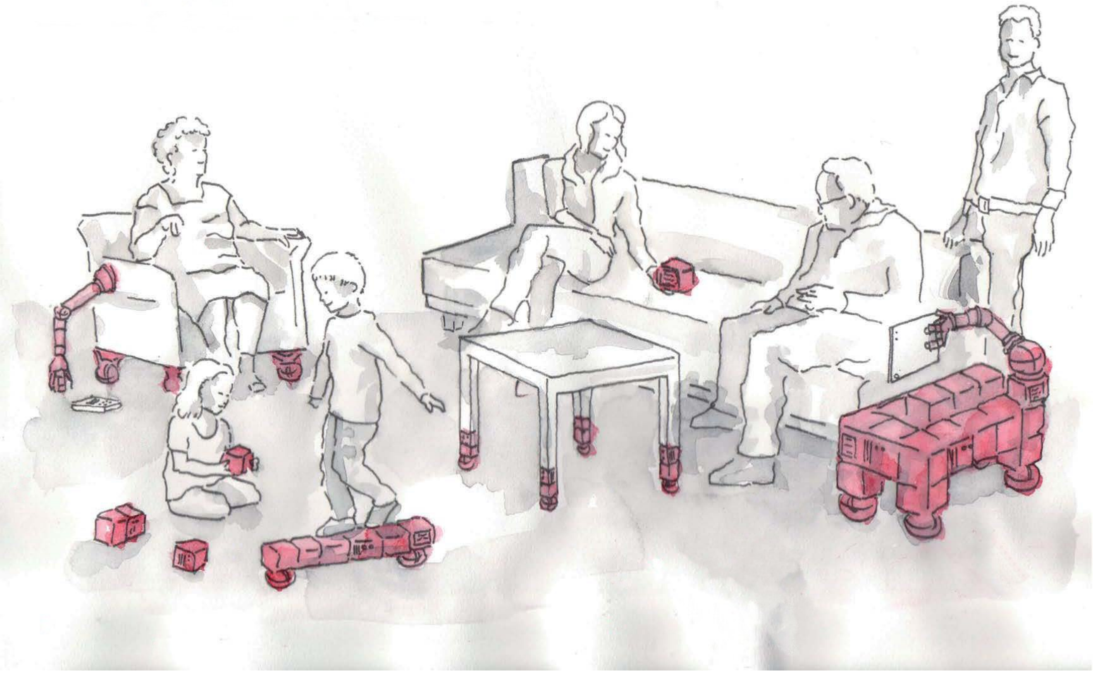


Figure 11.1 – Artist illustration of the Plug-n-Play Robotic Elements being used in a household to increase quality of life. Image courtesy of the Biorobotics Laboratory, EPFL.

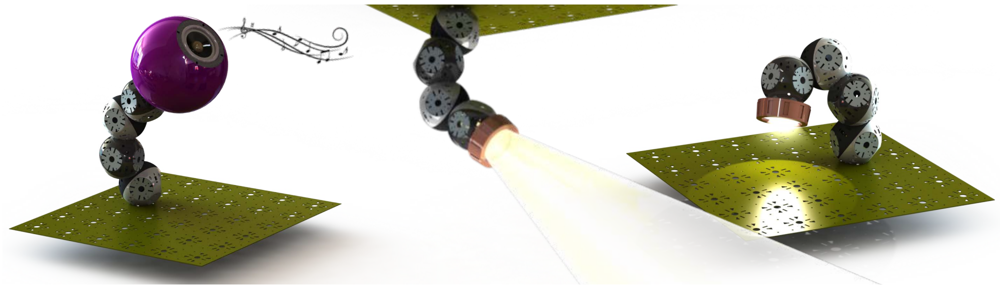


Figure 11.2 – Examples of external attachments being manipulated by Plug-n-Play Robotic Elements: (left) Re-orientable speaker. (center and right) Smart lighting system.

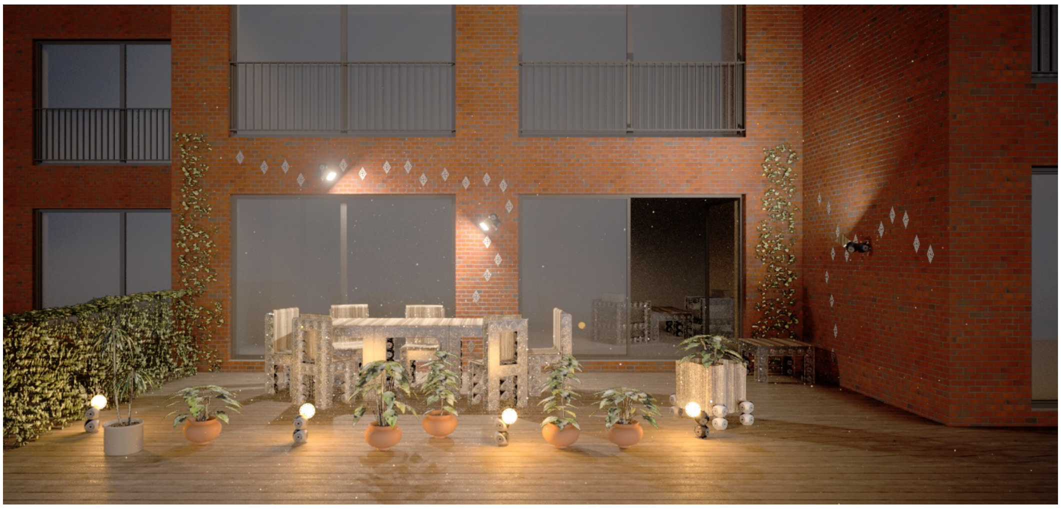


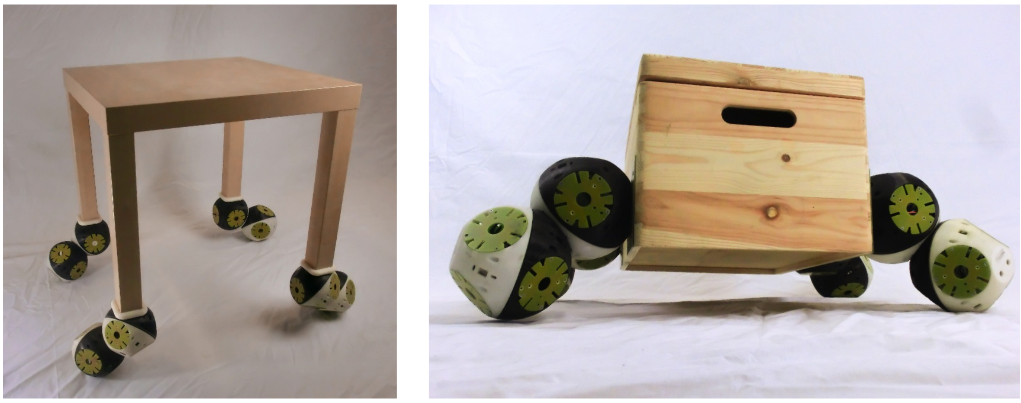
Figure 11.3 – Concept of a smart patio with plug-n-play robotic lighting elements and reconfigurable robotic furniture. Rendering created by Augustin Mercier.

act as manipulators and interact with the environment using dedicated external attachments equipped with different types of devices such as end-effectors, environmental sensors and transducers similar to those listed in Chapter 6.

Some concepts of devices that can manipulated, reoriented, and relocated using Plug-n-Play Robotic Elements are shown in Figure 11.2. Using intuitive interfaces, end-users can control the behavior of these modular robots. Figure 11.3, created by Augustin Mercier, one of our students, shows a concept of application scenario where an outdoor environment is enriched with modular robotic elements to create dynamic lighting points and moving structures.

11.2 Examples of Plug-n-Play Robotic Elements

As a demonstration of potential applications for the Plug-n-Play Robotic Elements we created a few preliminary demonstrations. Figure 11.4 shows two examples of standard existing pieces of furniture augmented with Roombots modules. The design of the pieces of furniture has



(a)

(b)

Figure 11.4 – Examples of standard pieces of furniture augmented with Plug-n-Play Robotic Elements: IKEA LACK table (a) and wooden box (b) equipped with four passive connection ports and four active Roombots modules connected to them.

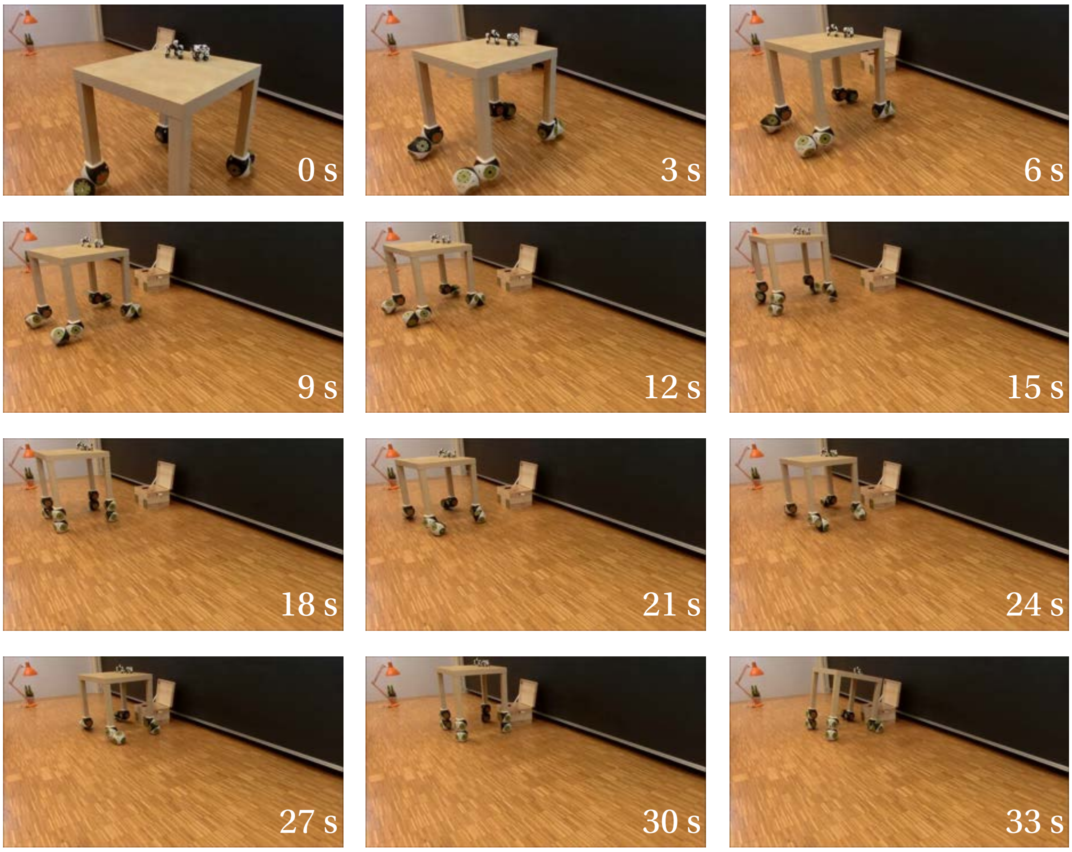


Figure 11.5 – Snapshots of an IKEA LACK table augmented using four Roombots modules. The robots are using their continuous-rotation capability to act as wheels. The augmented table is manually operated and able to move across the room and lean over to drop carried objects inside a box.

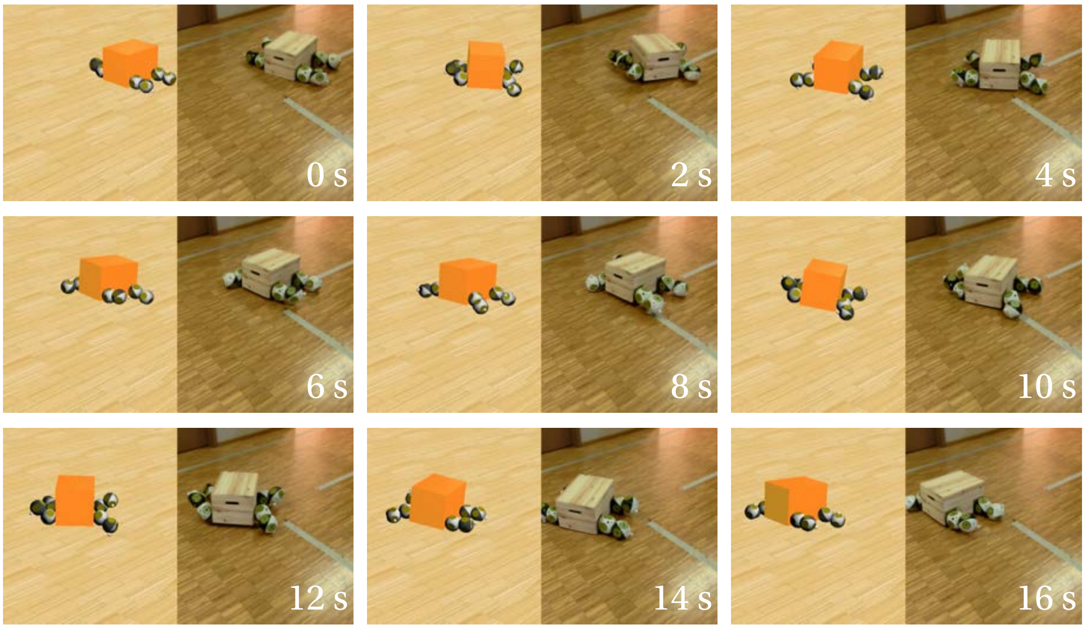


Figure 11.6 – Snapshots of an augmented wooden box performing articulated locomotion. The gait was first optimized in simulation (left side of the snapshots) and then transferred to the robots (right side). The wooden box is equipped with four Roombots modules used as legs. Each Roombots module is also connected to passive rubber feet extensions used to increase ground clearance.

been slightly modified to include passive connection ports. The assembly of the structures was done manually.

We demonstrate two different uses of the Roombots modules as Plug-n-Play Robotic Elements. In the first case we exploit their continuous rotation capability to use them as wheels. Figure 11.5 shows a locomotion sequence in which an augmented IKEA table rolls across the room, approaches an open wooden box, and leans over to unload some objects that were being carried. The sequence was controlled manually by an expert operator via low level motor position commands.

In the second example we used oscillatory motion to generate a gait that moves an augmented wooden box (Figure 11.6). Given the increased complexity of this task, we used our optimization framework to find CPG locomotion-controller parameters that lead to fast locomotion.

As an additional example of the prospective use of Plug-n-Play Robotic Elements, we developed a “smart lighting system” similar to the concept shown in Figure 11.2. A spotlight external attachment is manipulated and reoriented by a Roombots metamodule attached to a ceiling passive port (Figure 12.1). This smart lighting system provides on-demand localized illumination based on the user’s needs. The user interface used to control this element is explained in the next chapter.

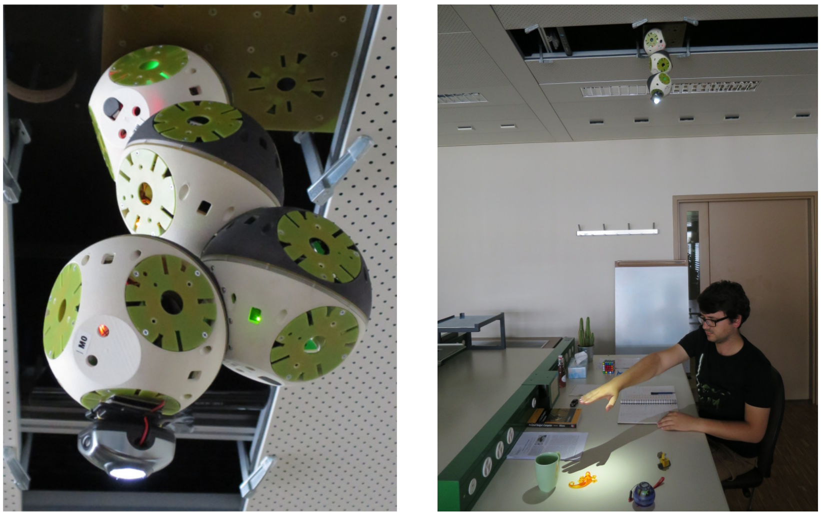

(a) (b)

Figure 11.7 – Demonstration of the smart lighting system: (a) Spotlight external attachment being manipulated by a Roombots metamodule connected to a ceiling passive port. (b) Example scenario in which the user is able to control the orientation and intensity of the spotlight. Images from [140].

11.3 Conclusions

In this chapter we described our long term futuristic dream of using Self-Reconfigurable Modular Robots as building blocks for Self-Reconfigurable Robotic Furniture. As a first step towards this scenario we use Roombots modules as Plug-n-Play Robotic Elements. These elements use their ACMs to connect to passive ports embedded in existing furniture, walls, or ceiling with the goal of creating new automated features in a household. We presented three examples of prospective uses: a table capable of rolling across a room, a wooden box that can walk with robotic modular legs, and a spotlight which is reoriented using a Roombots metamodule attached to a ceiling passive port.

The design of additional dedicated attachments is by itself quite simple. The desired device (e.g. spot light, screen, camera, gripper) needs to be embedded in a part which includes a Roombots passive port. Using one of its ACMs, a Roombots structure can connect and manipulate the attachment. The number of devices that can be included as attachments depends mostly on the imagination and creativity of the designer. The challenging tasks are the control, the lattice locomotion, sensing the environment, understanding the user's needs, and providing them with a proper human-robot interface. Many of these problems are still open research questions being addressed by many researchers in different fields. In the next chapter we will present two examples of user interfaces that allow controlling Roombots structures with natural gestures.

12 Gesture-Based User Interfaces for Self-Reconfigurable Modular Robots

One of the challenges in the field of Self-Reconfigurable Modular Robotics is to find a well suited interface that allows the user to control and direct movements, self-reconfiguration, and overall behavior of the modular structure.

In Section 12.1 we briefly describe several Roombots interfaces that were developed for expert users, which provide low-level control of the modules (e.g. direct motor-position control). We then proceed to present two user interfaces that allow non-expert users to control Roombots modules at different levels of abstraction.

In Section 12.2 we describe a vision-based lighting control interface used to control a Plug-n-Play Robotic Element composed of a Roombots metamodule with a spotlight attachment. The user can use hand gestures to regulate directionality and brightness of the light.

In Section 12.3 we present a Natural Roombots User Interface that translates pointing gestures into robot commands. The interface is used to direct on-grid locomotion sequences of two Roombots modules. Both interfaces are intuitive and do not require the user to wear or carry any type of sensor or device.

Some parts of this chapter have been adapted from:

[140] M. Mutlu, S. Bonardi, M. Vespignani, S. Hauser, A.J. Ijspeert, and A. Bernardino. "Natural User Interface for Lighting Control: Case Study on Desktop Lighting Using Modular Robots". *10th ACM International conference on Tangible, Embedded and Embodied Interaction (TEI).* In review.

My original contribution: Conceptual contribution.

[147] A. Özgür, S. Bonardi, M. Vespignani, R. Möckel, and 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.

My original contribution: Conceptual contribution, hardware development.

12.1 Introduction

In Chapter 11 we proposed the use of Plug-n-Play Robotic Elements to add new functionalities in a household to improve the quality of life. If we envision the use of SRMRs in every-day life environments we need to also take into consideration natural ways of interacting with such robots. These human-robot interfaces should be non-invasive and easy to use for lay users.

Several interaction strategies have been developed to control Roombots modules, targeting users with different levels of experience. The low-level control, for development and debugging purposes, allows direct access to the motion command sequences of single modules. A more abstract Graphical User Interface allows the manipulation of entire Roombots structures. In [18], the GUI was transferred from a computer screen to an iPad display. Using Augmented Reality the user can place virtual objects in the environment, displayed on the screen as seen by the iPad back camera.

In this chapter we present two natural interfaces that don't require the user to wear or carry any type of sensor or device.

12.2 Vision-based lighting control

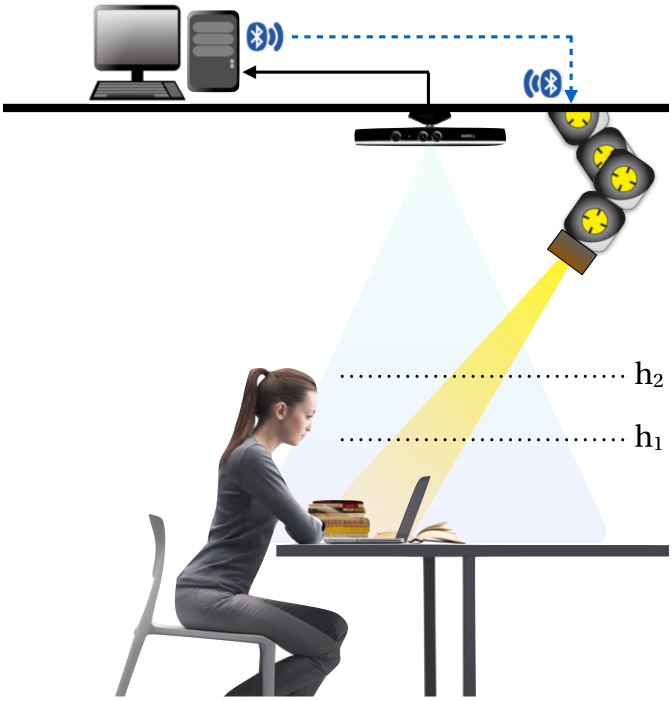
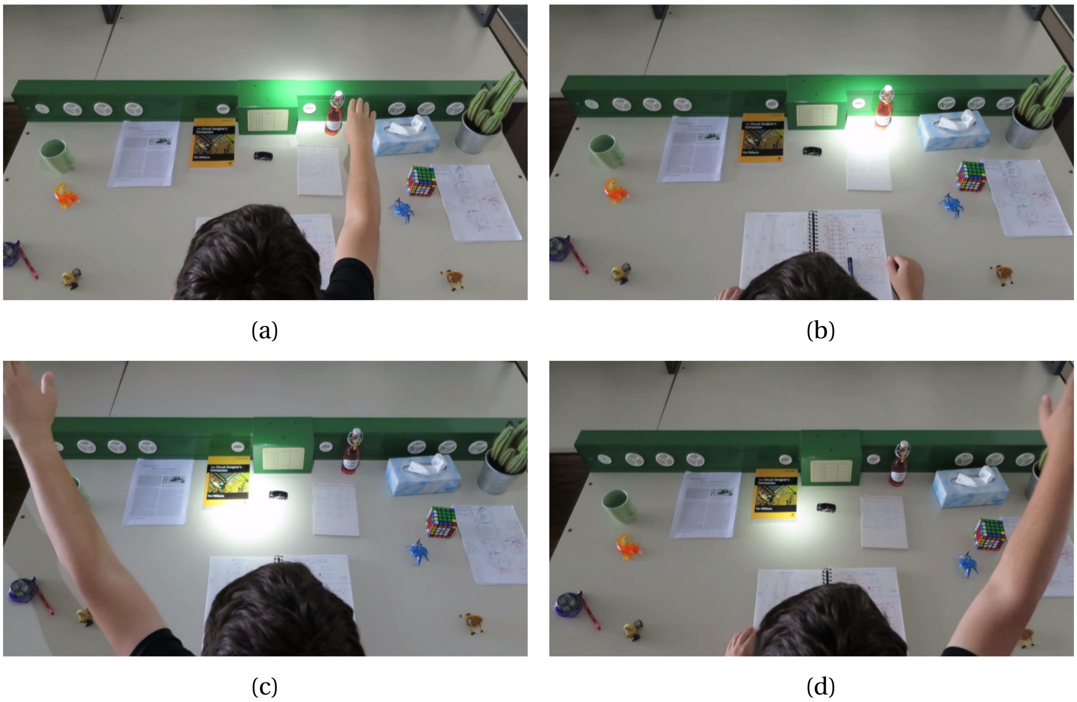


Figure 12.1 – Vision based lighting control: a Kinect mounted overhead detects user's hand gestures which are converted into robot control signals. By placing the arm at a height h_1 the user can redirect the spotlight. By raising the arm at a height h_2, the user can control the light brightness.



(a) (b) (c) (d)

Figure 12.2 – Vision based lighting control: (a and b) Spotlight directionality control. (c and d) Spotlight brightness control. Images from [140].

We proposed a novel tracking method that allows users to interact with a set of modular robots using hand movements. As application scenario, we considered a Roombots plug-n-play robotic lighting element in which directionality and intensity of the light are controlled by the user using simple gestures.

The system, shown in Figure 12.1, uses a spotlight extension mounted on a Roombots meta-module to illuminate the working area of a desk. A Kinect mounted overhead scans the tabletop using its depth sensor, searching for specific hand gestures by the user. A control computer analyzes the sensor data and generates pose and light intensity commands for the Roombots plug-n-play robotic element.

In order to activate the light position control, a user needs to lift their arm at a certain height h_1. The hand is tracked to calculate the desired light location. The DoF configuration necessary to light up the desired spot is calculated using geometrical methods and transmitted to the robot. Brightness is controlled by first moving the hand to the threshold height, h_2, to trigger the brightness adjustment mode. Then the brightness of the spotlight can be modulated by moving the hand left or right.

The proposed interface is designed to have a quick learning curve, in which the user just needs to learn the two heights h_1 and h_2 in order to effectively control the 6-DoF Roombots plug-n-play robotic element.

12.3 Natural Roombots User Interface

We created an intuitive gesture-based interface, called *Natural Roombots User Interface,* that allows the user to control individual Roombots modules, direct their on-grid locomotion, and receive visual feedback without carrying additional devices (Figure 12.3).

The user is able to select individual Roombots modules and move them to target positions by pointing at them. The interface works with a dual Kinect setup. The first Kinect is pointed at the user and recognizes the pointing gestures using its depth sensor. The second Kinect is pointed toward the grid environment (i.e. where Roombots modules and environmental passive ports are located) to detect its current state. The two Kinects are calibrated in order to share the same coordinate frame. We used the head-hand vector to estimate the pointing direction. Once detected, this vector is extended toward the grid environment and the first object that is intersected (if any) is considered to be the object that is targeted by the pointing gesture.

In order to enhance the usability of the Natural Roombots User Interface we included visual feedback to the user by means of RGB LEDs placed in the Roombots modules and on the connector ports in the grid environment, as shown in Figure 12.4. We included LEDs in the

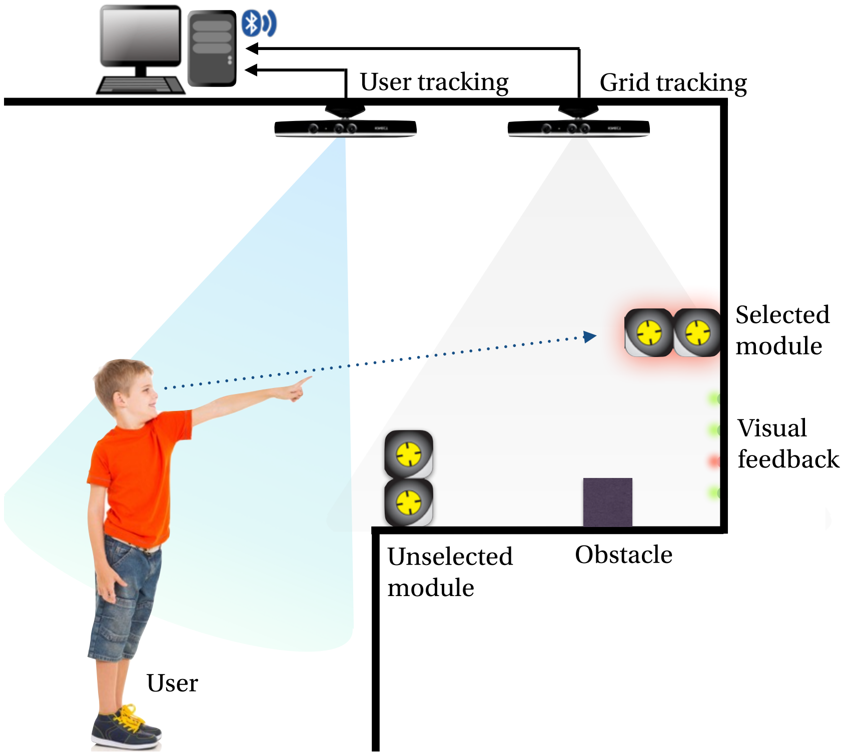
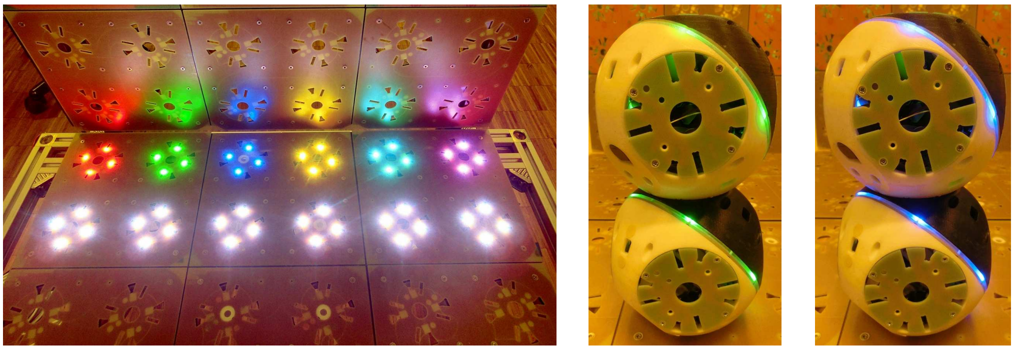


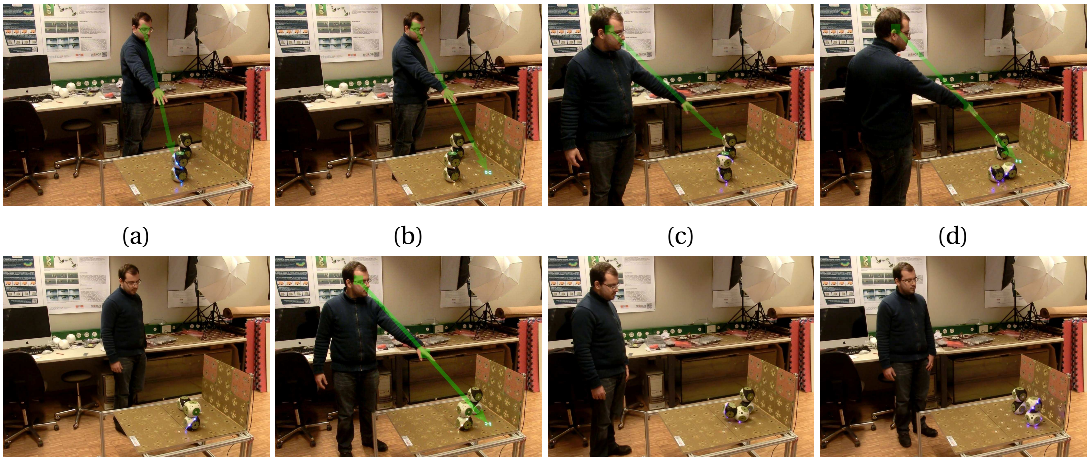
Figure 12.3 – Natural Roombots User Interface setup. Pointing gestures of the user are recognized by one Kinect. A second Kinect tracks the grid state. Visual feedback is given using LEDs in the Roombots modules and in the connector ports.

Roombots modules by placing six of them on each main plate, pointing outwards. Using the plastic ring described in Section 3.6 we were able to create a glow effect. RGB LEDs allow the creation of a multitude of colors by mixing different amounts of the three primary colors. We controlled the LEDs to create a number of visual effects such as *breathe*, in which the intensity is modulated over time, and *turn*, used on the Roombots's DoF to show for instance the direction of motion. While being aesthetically pleasing, these effects also convey additional



(a) (b) (c)

Figure 12.4 – RGB LED illumination: (a) Grid tiles LEDs in full intensity, showing different combinations of colors. (b and c) LEDs on the Roombots's diametrical degrees of freedom. Images from [147].



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 12.5 – Natural Roombots User Interface in action to control two modules (only of of them equipped with LEDs). (a) Roombots module 1 is selected. (b) Target connector port is selected. Module 1 starts an on-grid locomotion sequence to reach the target. (c) Roombots module 2 is selected. (d) Target connector port for module 2 is selected. Meanwhile, module 1 is still moving toward the target. (e) Module 1 stops because module 2 obstructs its path. (f) After module 2 reaches its target, module 1 and its target are selected again. (g) Module 1 resumes its locomotion sequence. (h) Both Roombots modules reach their targets. Images from [147].

information to the user.

Visual feedback is given to show the current target of the pointing gesture and to indicate the system state (e.g. available connector ports, planned path, obstacles). Idle Roombots modules are illuminated with a white-colored breathing effect. Idle connector ports are not illuminated. When the user points at the grid environment, modules or connector ports that are being pointed are lit with a cyan color. We defined a set of predefined commands that the user can execute. For instance, pointing at an object for more than two seconds selects that module as the active module.

Figure 12.5 illustrates an example of use of the Natural Roombots User Interface to control two modules. We used two Roombots modules to test selection and simultaneous path execution with multiple modules. Path planning and on-grid locomotion (not part of this thesis) were implemented using methods developed by Stéphane Bonardi [16].

12.4 Conclusions

In this chapter we presented two interfaces that allow a lay user to control different functions of Plug-n-Play Robotic Elements. In the first example the user can use hand gestures to control directionality and brightness of an automated spotlight. The Natural Roombots User Interface allows a direct control of Roombots modules to guide their lattice locomotion on a grid environment using pointing gestures.

Although these two interfaces have been only tested in a controlled laboratory environment, we believe that these are the first necessary steps toward the seamless integration of Plug-n-Play Robotic Elements and Self-Reconfigurable Robotic Furniture in every-day life environments.

13 Conclusions and Outlook

The goal of this thesis was to contribute in several ways to tackle some of the challenges in the field of Self-Reconfigurable Modular Robotics. To reach this objective we first improved the hardware of Roombots modules by proposing new mechanical solutions to increase their performance and robustness. Second, we proposed new algorithms for locomotion learning for arbitrary modular structures. Third, we explored the effect of in-series passive compliant elements on locomotion. Fourth, we proposed new application scenarios and interaction methods for using Self-Reconfigurable Modular Robots to create an active household environment that improves quality of life.

In this thesis we proposed the following original contributions:

- **Mechanical redesign and implementation of the Roombots actuation**
  In Chapter 4 we explained the redesign process for the actuators used in Roombots modules. We proposed three new gearbox implementations (B, C, and D) designed to reduce backlash, allow for better manufacturability, withstand higher torque, and decrease their decrease the space required to house them. With Gearbox B we reduced the thickness of the gearbox by 23% by using different materials and manufacturing techniques. Gearbox C, the last design built in plastic, has a more robust internal structure, which allowed it to withstand 21% more payload compared to the previous versions. Additionally it has a 12-bit absolute encoder at the output, which is used to actively compensate for backlash. The newest version, Gearbox D, combines all the previous improvements in a metal design that is able to withstand a twofold increase in output load in comparison with the original design.
- **Improvement of the Roombots Active Connection Mechanisms**
  In Chapter 5 we proposed two improvements for the Roombots Active Connection Mechanisms (ACMs). ACMv4 has improved kinematics which create a larger scooping movement of the latches and increases the gripping range by 50% from 3.1 mm to 4.7 mm. This improvement allowed us to complete an autonomous on-grid locomotion sequence with a Roombots module on a vertical grid for the first time. The hybrid

ACM is an extension of the ACMv4 which combines the advantages of physical latching with the use of magnets to improve the self-aligning properties of the connector. This new design allowed us to dramatically increase connection success rate during on-grid locomotion or self-reconfiguration tasks.

- **A bio-inspired approach to speed up optimization after morphological change**
  In Chapter 8 we presented two parameter-reduction methods for SRMRs that allow for quick locomotion-parameter optimization in the first minutes after a change in morphology. Body/Limb Finder detects bio-inspired sub-structures and articulation joints in any arbitrary modular structure and applies parameter reduction using a set of pre-defined rules that restricts the number of active DoF. This can also be used in combination with the Symmetry Finder method, which identifies symmetrical sub-parts of the modular robot that can share the same optimized parameters. The two methods provide quick learning of new locomotion parameters that achieve on average a value of fitness three times higher than an unconstrained optimization.
- **A hybrid optimization method for the improvement of the simulation environment**
  In Chapter 9 we proposed a hybrid optimization technique that combines advantages of offline and online optimization by providing simulation results that are well matched with hardware experiments (with a reduction of the reality gap). After a first offline locomotion-parameter optimization step, a selection of gaits obtained from simulation are evaluated with the hardware robot. A second optimization step takes place to optimize simulation environment variables. The optimized simulator can then be used to find new optimized gaits. In contrast to offline optimization, the extra steps required by hybrid optimization allow results to be almost seamlessly transferred to the hardware robot. Compared to online optimization, we estimated at least a 2.5-fold reduction in the required optimization time.
- **Improvement of locomotion speed and energy efficiency in (Self-) Reconfigurable Modular Robots by using in-series passive compliant elements**
  In Chapter 10 we analyzed the effects of in-series passive compliant elements on the locomotion of Roombots and 8-DoF Lola-OP™ structures. With two Roombots we showed a 65% increase in locomotion speed after the introduction of in-series compliance. We extended the study on a simulated 8-DoF snake robot performing rolling and sidewinding on different rough terrains. In this case softer and stiffer structures produced locomotion runs with similar speed. We further extended this work to hardware 8-DoF Lola-OP™ snake robots and considered the specific task of rolling on a horizontal pipe. We showed that compliant elements can increase the adaptability of a modular snake robot to changes in diameter of horizontal pipes. While showing similar locomotion speed, compliant Lola-OP™ robots are more than 50% more energy efficient compared to their stiff implementation.
- **Plug-n-Play Robotic Elements**
  In Chapter 11 we proposed the use of Plug-n-Play Robotic Elements as a first step

towards realizing the Self-Reconfigurable Modular Furniture scenario. Plug-n-Play Robotic Elements are SRMR's modules that can be used to augment standard pieces of furniture or to manipulate dedicated extensions plugged into passive connector ports distributed in the environment. We presented three examples of prospective scenarios.

- **Two intuitive user interfaces for SRMRs**
  In Chapter 12 we proposed two interfaces that allow non-expert users to control the behavior of SRMRs and Plug-n-Play Robotic Elements using gestures. In the Vision-based lighting control interface the user can intuitively control directionality and brightness of an automated spotlight connected to a Roombots metamodule. The Natural Roombots User Interface allows a direct control of the modules to guide their lattice locomotion on a grid environment using pointing gestures. Both interfaces are intuitive and do not require the user to wear or carry any type of sensor or device.

Discussion and Future Challenges

In Chapter 4 we analyzed and improved the Roombots actuation in order to make it stronger and more robust for use in self-reconfiguration and locomotion tasks. The three motor-gearbox assemblies are the most expensive[1] and one of the most critical subparts in a Roombots module. For these reasons we put extra effort into the redesign process in order to achieve a product that would satisfy all of our requirements. We evaluated several custom-designed and commercial gearboxes and motor-gearbox combinations. Off-the-shelf products could not could fit our purpose. We found that the market is currently divided between small and compact "hobby-like" products and sturdy, robust, heavy, and high-precision professional products. The formers usually have hardware specifications that are almost one order of magnitude below our requirements. As an example, Dynamixel MX-106 motors, the strongest actuators in the Robotis product range, have a size that would fit in the Roombots hemispheres but a *stall* torque of 10 Nm, compared to the *nominal* torque of 8.4 Nm of our Gearbox D with Maxon motor design[2]. On the other hand, high-end professional gearbox systems are usually either too large for our space constraints or too heavy. The requirement of having a hollow gearbox constrains its design, making it much more complex. This design is needed because we chose to have continuous rotation capabilities for the Roombots DoF and electrical slip rings to transfer power and data between hemispheres. Continuous rotation is theoretically advantageous in self-reconfiguration sequences; it is also useful for locomotion as modules can be used as wheels or a combination of rotations and oscillations [158]. In case of future development of a new self-reconfigurable modular system we would however evaluate real advantages and disadvantages of designs in which modules are able to perform continuous rotation. Additionally, we would take into consideration completely different modular designs

---

[1] Currently the cost of three Gearbox D units, three Maxon motors of the correct type, and three absolute encoders (one per DoF in a Roombots module) amounts to roughly 1700 CHF (approximately $1700), compared to about 3000 CHF for a complete module.

[2] Usually the stall torque is between 5 to 10 times higher than the nominal torque. Nominal torque not provided for Dynamixel motors.

and technologies, e.g. taking inspiration from the field of soft robotics using fluid-filled compliant skins such as those developed by Pneubotics [156], that could scale better in terms of number of modules, size, and cost.

In Chapter 5 we analyzed and redesigned the Roombots ACMs in order to improve their tolerance to misalignment. This redesign enabled Roombots to successfully complete single-module locomotion through reconfiguration tasks (explained in Appendix B), which was not possible with the previous ACMv3 connectors. At the moment, even with the ACMv4 or the hybrid ACM, we are not able to complete sequences of on-grid locomotion using Roombots metamodules (i.e. two Roombots modules connected together). The reason is the deflection of the long lever arm of the metamodule, which results in errors of one or more centimeters in the alignment with the target connector. The sources of this deflection are, in order of magnitude, backlash in the Roombots actuators, wiggling in the bonding between the two connectors at the opposite side of the metamodule, and deformation of connector plates and internal structures. We targeted the reduction of backlash during the redesign of the actuation system. In gearbox designs C and D the control system can compensate for backlash using the absolute encoder mounted on the gearbox's output shaft. However, this feature was only tested in isolated gearboxes controlled by an Arduino board because the old version of the Roombots motor control boards does not support communication with the absolute encoder. New electronic boards are currently being developed and will be available soon. We believe that eliminating the effect of backlash could already be sufficient to allow metamodule on-grid locomotion. An interesting future direction could also be the use of a closed-loop control that actively controls the position of the metamodule based on a measurement of the misalignment between controlled and target connectors. This measurement should be done using exteroceptive sensors such as infrared distance sensors or vision systems.

In Chapter 8 we proposed two parameter-reduction methods for SRMRs that are useful for quickly learning new gaits after morphological changes due to self-reconfiguration or manual assembly. We developed these methods with a time-critical scenario in mind in which the robot is forced to change shape and needs to learn a "good-enough" gait[3] to be able to return to its main task. In our work we let all the optimization (FO, BLF, BLF-SYM, and SYM) run for 800 iterations and then compared them to evaluate the quality of the different solutions. In a real application scenario we would need a method that can automatically estimate the number of iterations necessary to have a good-enough solution. Moreover, in our work we defined empirical rules for the detection of bio-inspired articulation joints. While these rules produced optimized gaits that were significantly better than FO gaits, there is no guarantee of their optimality. It would therefore be useful to create an automated system that calculates optimal reduced networks for any arbitrary structure.

In Chapter 9 we developed a hybrid optimization method that allowed us to reduce the

---

[3] If we consider the optimization time t_o needed to learn new parameters and the locomotion time t_l that the robot requires to reach its target position using these new parameters, we look for solutions that minimize the total time $t = t_o + t_l$.

reality gap. The reduction was obtained through an optimization of simulation environment variables based on a similarity ratio between simulation and hardware experiments. We defined similarity ratio as the percentage of overlap between the virtual and hardware robot's body projections over time. As discussed in Section 9.3 this definition might not be optimal and could limit the effectiveness of the method. The hybrid optimization is a cycle which includes a first offline optimization, the evaluation on hardware of selected gaits, and the meta-optimization process. Once the cycle is complete, results from meta optimization can be used for further offline optimization runs which will produce results that are better matched with the hardware. Results from meta-optimization are in different measures dependent on the robot morphology, the type of gait, and the ground properties. An interesting research question is the evaluation of how these results generalize to different conditions and when it is necessary to trigger a new meta-optimization process.

In Chapter 10 we investigated the use of in-series passive compliant elements for locomotion with Roombots modules and 8-DoF Lola-OP™ robots. In the first case we reported a clear increase in locomotion speed when using compliant structures instead of stiff ones. This was not found to be the case for the experiments with the modular snake robot. One reason for this could be the different types of gaits performed by the two robots. The two Roombots modules had more abrupt movements that were smoothed out when the structure was more compliant. For the experiments with the Lola-OP™ robot we used instead traditional snake gaits (sidewinding and rolling) that do not generate large impact forces with the ground. Perhaps there could also be a different effect of compliance on locomotion when we compare "slow and heavy" Roombots to "fast and lightweight" snake robots. For future experiments we would like to find ways to measure the deflection of each compliant element to estimate how much energy is being stored and released. With the horizontal pipe locomotion experiments we reported a clear increase in energy efficiency for more compliant configurations. This was likely due to a passive adaptation to the complex terrain features. Based on these results, it would be interesting to validate the simulation results of Section 10.2 with the hardware robot. For further studies we could increase the repertoire of types of compliant elements and investigate also the effect of external passive extensions, which go in direct contact with the ground. A thorough search could also be done on the hardware robot to further investigate if actuation frequency combined with passive compliance has an effect on locomotion efficiency.

In Chapter 11 we presented examples of augmented furniture and Plug-n-Play Robotic Elements used to create new features in a domestic environment. These first demonstrations showed the feasibility of these types of applications for SRMRs. At the moment the torque generated by the Roombots modules is sufficient to manipulate dedicated extensions and move lightweight objects like an IKEA LACK table but certainly not enough to lift heavier pieces of furniture. We believe that a key element towards the realization of the Self-Reconfigurable Modular Furniture scenario is the creation of dedicated heterogeneous modules designed for specific types of movements. An example of such modules is an omnidirectional wheel module equipped with active connectors. This specialized module would primarily be used for relocating augmented furniture.

In Chapter 12 we proposed two device-free user interfaces for controlling SRMRs using gestures. The two interfaces seemed natural and easy to use. Gesture-based user interfaces have the potential to provide a robot control at different levels of abstraction (e.g. control of the overall behavior, such as in the control of orientation and brightness of the smart lighting system; control of the behavior of each Module, such as in the Natural Roombots User Interface). Future directions could include the integration of speech recognition in the Roombots user interface to provide an even more natural way to interact with the robot.

Appendices Part

A List of modular robots

Table A.1 – List of 111 modular robots, from 1988 until 2015. Data from [204] and [2], integrated with our best knowledge on most recent publications.

Year	No.	Module	Reference
1988			
	1	CEBOT	[57]
	2	CMU RMMS	[187]
1993			
	3	Polypod	[246]
1994			
	4	Fracta	[135]
	5	Metamorphic	[33]
1996			
	6	Ganryu	[76]
	7	Tetrobot	[74]
1998			
	8	3D-Fractum	[100]
	9	Molecule	[99]
	10	Vertical	[78]
1999			
	11	CONRO	[237]
	12	I-Cube	[222]
	13	Micro-robot	[221]
	14	Miniaturized System	[255]
2000			
	15	Crystalline	[227]
	16	Micro-unit	[254]

	17	M-TRAN	[102]
	18	Proteo	[14]
	19	PolyBot	[244]
2001			
	20	ANAT	[92]
2002			
	21	Molecule II	[177]
	22	Pneumatic	[84]
	23	TeleCube	[213]
	24	Uni-Rover	[88]
2003			
	25	CHOBIE	[83]
	26	Gear-Type	[219]
	27	M-TRAN II	[103]
	28	Swarm-Bot	[132]
2004			
	29	ATRON	[146]
	30	Claytronics	[40]
	31	HitMSR	[260]
	32	Neubot	[72]
	33	Random Parts	[69]
	34	Stochastic 2D	[236]
	35	Y1	[68]
2005			
	36	Catoms	[95]
	37	HYDRON	[145]
	38	Molecube	[264]
	39	Microtube	[27]
	40	Programmable parts	[13]
	41	Stochastic 3D	[233]
	42	Superbot	[180]
	43	Slimebot	[196]
2006			
	44	AMAS	[216]
	45	CkBot	[151]
	46	Deformatron	[207]
	47	MEMS	[49]
	48	ORTHO-BOT	[165]
	49	roBlocks	[189]
	50	YaMoR	[131]

2007			
	51	AMOEBA-I	[109]
	52	CHOBIE II	[98]
	53	Miche	[66]
	54	M-TRAN III	[101]
	55	ROBMAT	[52]
	56	Shady 3D	[253]
	57	TETwalker	[29]
	58	Tribolon	[127]
	59	XBot	[234]
2008			
	60	Em-cube	[4]
	61	GZ-I	[259]
	62	Morpho	[257]
	63	Odin	[111]
	64	SYMBRION	[91]
2009			
	65	Raupi	[113]
	66	Roombots	[203]
2010			
	67	Factory Floor	[58]
	68	iMobot	[178]
	69	Kilobot	[176]
	70	Modular-Expanding	[10]
	71	Mod-Leg	[166]
	72	Responsive Truss	[125]
	73	Sambot	[231]
	74	Single-material	[220]
	75	Smart Pebbles	[65]
	76	Thor	[112]
	77	UBot	[214]
	78	Vaccubes	[61]
2011			
	79	Anibot	[251]
	80	Crossball	[124]
	81	Cubelets	[188]
	82	Heterogeneous modules	[53]
	83	HitMSR II	[263]
	84	M-Cubes	[241]
	85	M-Lattice	[70]

	86	HexBot	[179]
	87	SMART	[8]
	88	X-Cell	[77]
2012			
	89	ModRED	[79]
	90	M3Express	[238]
	91	Smart Blocks	[128]
	92	SMORES	[46]
	93	Soft Module[1]	[63]
2013			
	94	CoSMO	[105]
	95	EDHMoR[1]	[54]
	96	Fable	[148]
	97	Hybrid Robots[1]	[104]
	98	M-Block	[174]
	99	Single module	[163]
	100	USS	[240]
2014			
	101	Hinged-Tetro	[89]
	102	MOSS	[129]
	103	ReBis	[218]
	104	Reconfigurable Endoscopic Capsule[1]	[252]
	105	ReCTeR	[28]
	106	Soft Cell[1]	[62]
2015			
	107	Fable 2	[149]
	108	HMA-based RBE[1]	[25]
	109	PMR	[48]
	110	Soldercubes	[142]
	111	T.E.M.P.	[153]

---

[1] Unofficial name.

B On-Grid Locomotion through Self-Reconfiguration

In this chapter we present two hardware experiments that show the capability of Roombots modules to locomote on a 3D grid environment. The robot cannot perceive the environment and joint moves were pre-computed. The tasks were executed autonomously with an open-loop control, without human interference. We manually triggered ACM connections and disconnections to ensure that a successful ACM connection had been properly established before taking the next step.

Some parts of this chapter have been adapted from:

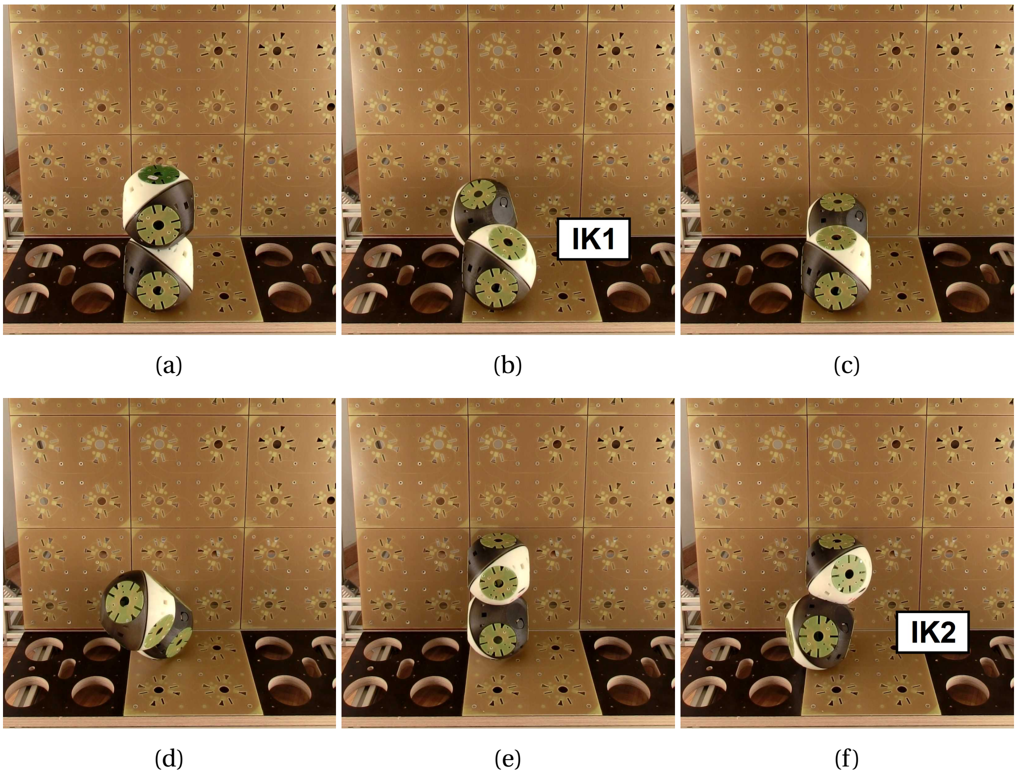
[204] A. Spröwitz, R. Moeckel, M. Vespignani, S. Bonardi, and A.J. Ijspeert. "Roombots: A Hardware Perspective on 3D Self-Reconfiguration and Locomotion with a Homogeneous Modular Robot". In *Robotics and Autonomous Systems*, Volume 62, Issue 7, July 2014, Pages 1016-1033, 2014.

> **My original contribution**: Conceptual contribution, Experimental setup, Hardware experiments, Partial writing.

Single Roombots Module On-Grid Locomotion in 3D

This experiment illustrates the ability of a single Roombots module to autonomously overcome a concave edge and climb a structured wall. The setup (Figure B.1a) is composed of a horizontal and a vertical plate (of 2 × 6 and 8 × 6 grid units), shaping a concave corner. A single Roombots module starts on-grid, one grid unit away from the wall, and climbs the wall in 14 steps (Figure B.2), with four sub-sequences:

1. The approaching move towards the wall is a pre-computed, inverse kinematic (IK) based motion sequence (Figures B.1a-B.1c). To avoid collision with the wall through the module's rotation, the approaching ACM is guided through IK based motion to stay parallel to the approaching wall.



(a) (b) (c)

(d) (e) (f)

Figure B.1 – A single Roombots module overcoming a concave edge. (a) Initial configuration with the module placed in its SRZ configuration, one grid unit away from the wall. (b) The module approaches the wall using inverse kinematics. (c) Connection to the vertical plate (first row) and disconnection from the horizontal one. (d) The module approaches the wall with a simple rotation of a diametrical DoF. (e) Connection to a passive port on the second row and disconnection from the first row. (f) The module escapes the concave corner upwards using inverse kinematics. Images from [204].

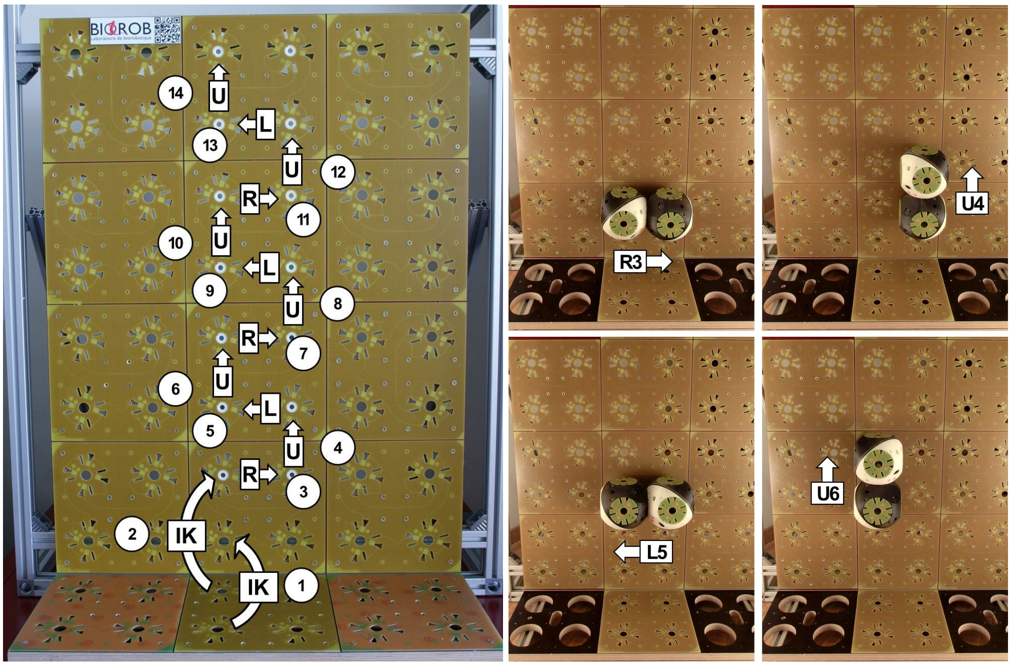
2. The module connects the wall approaching ACM, and disconnects its foot ACM (Figure B.1c).

3. As before IK based motion was applied to avoid ACM-wall collision when rotating the module upwards (Figure B.1f).

4. A cyclic sequence of moves was used to climb the vertical surface. As explained in [19], the Roombots's DoF does not allow for a straight movement, but the module performed a zig-zag motion: right, up, left, and up (Figure B.2). Each move was followed by an ACM connection-disconnection.

In Figures B.1 and B.2 we show one Roombots module overcoming a concave edge and climbing a vertical grid wall.

In all above experiments we proceeded with an identical set of Roombots hardware: two Room-



(a)

(b)

Figure B.2 – Single Roombots module on-grid locomotion in 3D experiment: (a) The sequence of 14 atomic moves used to climb the wall. IK: inverse kinematic based motion, R move to the right, U move up, L move to the left, in the global coordinate system. A connection and disconnection phase took place between each move. (b) Detail of four atomic moves. Images from [204].

bots modules, each equipped with two ACMv4 with improved scooping movement[1]. Roombots modules equipped with the old ACMv3 connectors were able to perform the concave edge sequence, but could not successfully execute step R3 because of the limited misalignment tolerance. Under standard laboratory conditions, such as module preparation, careful preparation of the on-grid plates and their connector ports, and constant module maintenance, the ACMv4 gripping range of 4.7 mm was sufficient to successfully complete the whole sequence. Robustness was however quite low and we had to repeat the experiment several time in order to have a flawless run with no human intervention to help the modules. The addition of the small magnets in the hybrid ACM helped to dramatically reduce the number of unsuccessful connection attempts. The increased robustness allowed us to demonstrate multiple times the whole experiment during an open-day exhibition and several lab tour demos.

Two Roombots Modules Join into One Metamodule - Crossing Convex Edges

This experiment demonstrates the ability of a Roombots module to overcome a convex edge with the help of a second module, and the ability of Roombots modules to join into meta-

---

[1] See Chapter 5 for details about the ACM mechanic design.

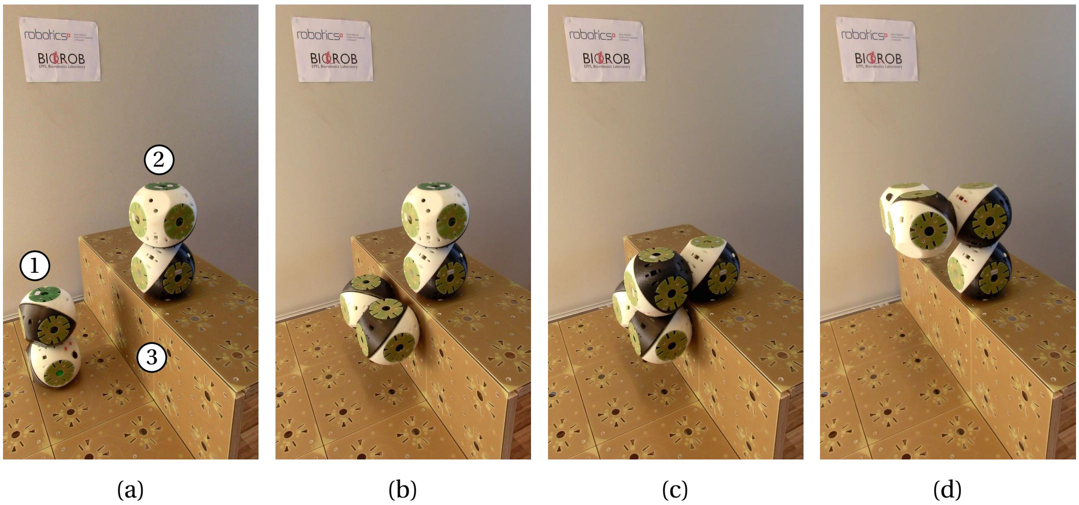


Figure B.3 – A Roombots module overcomes a convex edge by joining with a helper module into a Roombots metamodule. (a) Module 1 (M1) is one grid unit away from the wall. Module 2 (M2) waits on the top surface. (b) M1 climbs the wall for one grid unit. (c) M2 connects its ACM to the approaching M1. (d) M1 is lifted onto the horizontal plane. Images from [204].

modules. We showed earlier that single Roombots modules can overcome concave edges. To overcome a convex edge, a single Roombots modules requires the assistance of a second module. This is a kinematic constraint: the shortest distance from one connection port on the wall, over the convex edge, to the next port on the above plane is three grid units. Single Roombots modules are only two grid units long.

The experimental setup was composed of three grid planes forming a concave/convex edge (#3 Figure B.3a). Two untethered Roombots modules, M1 and M2, were used. M1 was placed on the lower plane, and M2 "waited" on the upper grid plane. To approach and climb the wall, M1 performed the same motion sequence as in the concave-edge experiment (Figure B.1). Once M1 reached M2, both connected (Figure B.3c) into a metamodule. The metamodule lifted itself over the convex edge (Figure B.3d).

At the end of this sequence the two modules can separate and continue their on-grid locomotion through self-reconfiguration task. The Roombots modules could also continue their on-grid locomotion as a metamodule. This configuration would potentially allow them to reach more remote positions in the 3D grid [16]. However hardware validation of this ability could not be demonstrated. Roombots metamodules show too large a deflection[2] which prevents them from connecting to the target ports using just an open-loop control.

---

[2] Roombots metamodule on-grid locomotion was only tested using modules with the Gearbox B (without absolute encoder) and hybrid ACM.

C Roombots Quadruped Locomotion

In this chapter we show snapshots of a locomotion experiment with five Roombots modules assembled in a quadrupedal structure (Figure C.1). The robot was equipped with four external passive rubber feet extensions to increase friction with the ground. Roombots modules use oscillatory movements produced by a network of coupled phase oscillators (CPGs). The structure has 12 actuated DoF, 3 for each leg module. The spine module is not actuated. The CPG control parameters were previously optimized in simulation using a Particle Swarm Optimization. The simulation environment was optimized using meta-optimization.

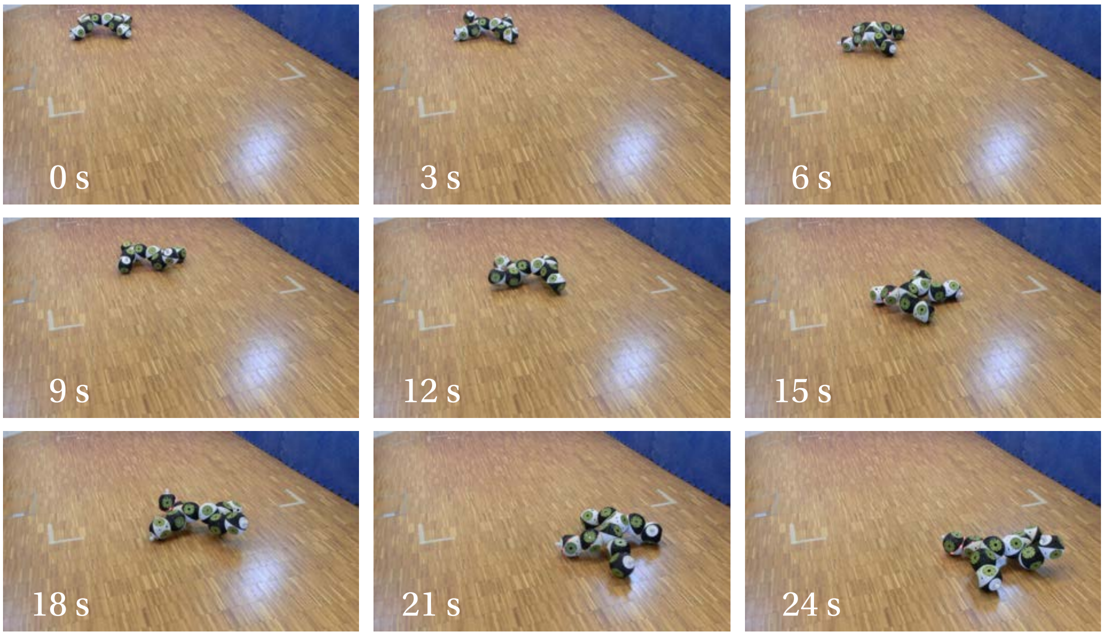


Figure C.1 – Locomotion of a quadrupedal structure made of five Roombots modules.

D Self-Reconfiguration from Tripedal to Snake Morphology

In this chapter we show snapshots of a self-reconfiguration sequence in which a Roombots structure composed of three modules changes morphology (Figure D.1). The structure has 9 actuated DoF. At the beginning the robot moves with a tripedal shape, using oscillatory movements produced by a network of coupled phase oscillators (CPGs). The locomotion-controller parameters were pre-optimized using offline optimization. After $t = 12$ seconds, we manually triggered a self-reconfiguration sequence. This would normally happen automatically when the robot evaluates that the current morphology is not well adapted for the task or the specific environment. However this feature has not yet been implemented in Roombots. The self-reconfiguration sequence executes a pre-determined list of movements, connections, and disconnections in order to change the connectivity of the structure. This process requires 30 seconds. At $t = 42$ seconds, new locomotion-controller parameters for snake locomotion are loaded into the Roombots modules. These new parameters were also pre-optimized. Thanks to the self-aligning properties of the hybrid ACM, we were able to repeat this self-reconfiguration sequence multiple times with high success rate.

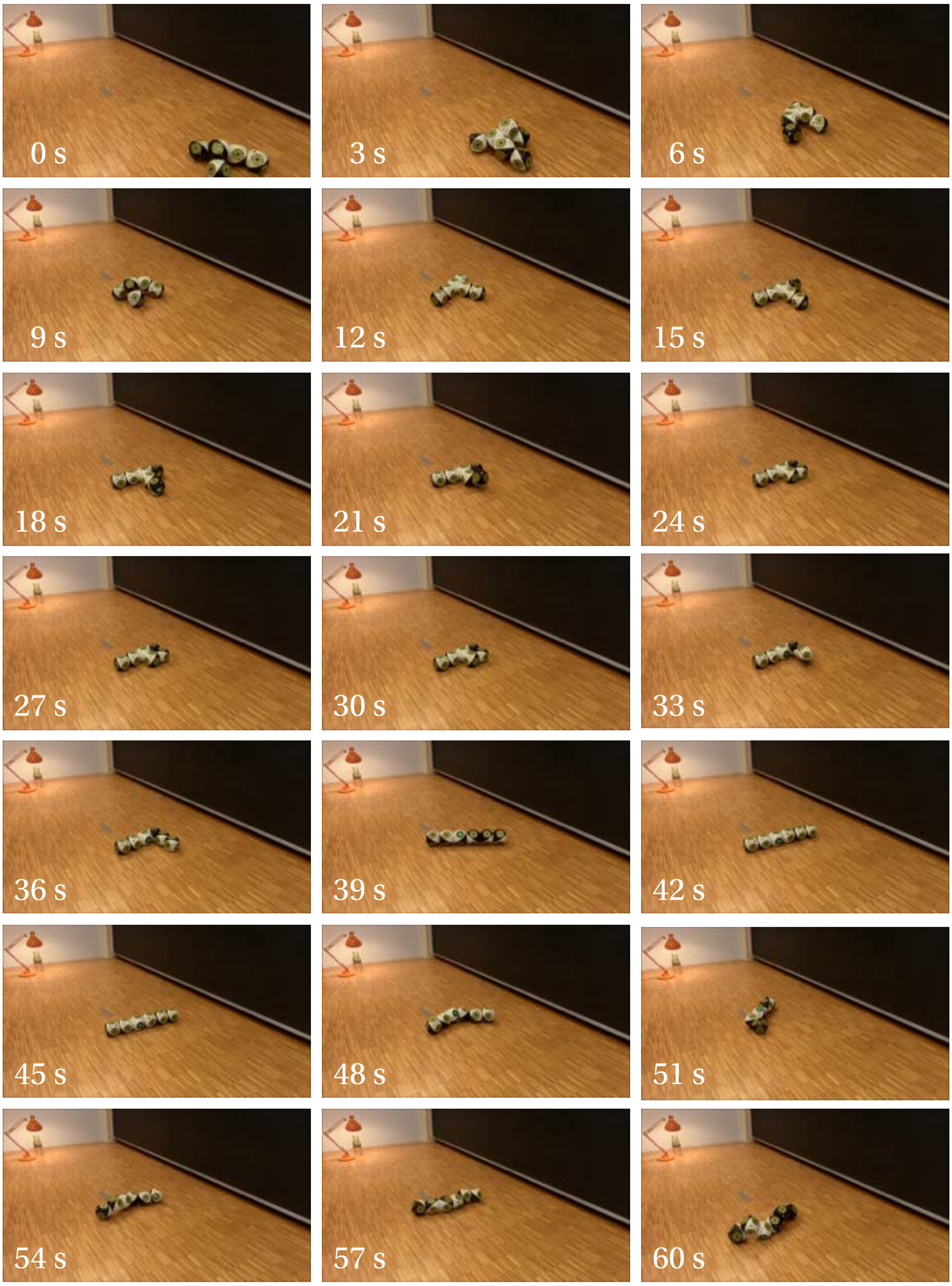


Figure D.1 – Roombots Self-Reconfiguration from Tripedal to Snake Morphology.

E Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a population based, stochastic optimization algorithm which uses so-called particles to explore the parameter search space. PSO is a cooperative algorithm in which individual particles share information about known solutions of the particular problem being solved (using principles of collaboration rather than competition). The velocity v_i and the position x_i attributed to each particle i are governed by the following equations:

$$v_i(t+1) = w \cdot v_i(t) + r_{i1} \cdot c_1 \cdot (X_i - x_i(t)) + r_{i2} \cdot c_2 \cdot (X_g - x_i(t)) \tag{E.1}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{E.2}$$

where X_i is the best solution found by particle i and X_g is the best global known solution, r_{i1} and r_{i2} are two random numbers uniformly distributed between 0 and 1, w is the inertia factor, and c_1 and c_2 are two constants that weight the contribution of local versus global search.

We chose PSO over other optimization techniques for the following reasons:

1. With its relatively low number of computations, PSO is a lightweight optimization method that can even run on slower microcontrollers.
2. In contrast to other optimization techniques, PSO does not require gradients and performs well also on more complex search spaces.

We implemented the Particle Swarm Optimizations used in Chapters 8, 9, and 10 using the Optimization Framework developed by Jesse Van den Kieboom [93]. This framework is well integrated with Cȯdẙn, with Webots, the mobile robot simulator that we used for our simulation experiments [229], and with our cluster of 18 dual quad core Intel Xeon E5504 processors[1].

---

[1] This computation power is not needed for the PSO algorithm itself, but instead to run multiple instances of Webots at the same time in order to parallelize the execution of the different particles.

Settings of the optimization framework include the number of particles and iterations and the definition of the fitness function that needs to be evaluated. Additionally, the framework offers the possibility to repeat each particle multiple times and consider its worst, average, or best performance; this is useful to test the same candidate locomotion parameters multiple times with different robot's initial conditions (e.g. different orientation) in order to evaluate their robustness.

List of Figures

2.1 Examples of Modular Robots: (a) CEBOT. (b) Molecule. (c) PolyBot G3. (d) Odin. Pictures from [57, 99, 249, 111]. 11

2.2 Overall trend of total number of Modular Robots. Data from Table A.1, reported in Appendix A. 13

2.3 More examples of Modular Robots: (a) M-TRAN. (b) SuperBot. (c) ATRON. (d) Molecubes. (e) SMORES. (f) Soldercubes. Pictures from [138, 180, 134, 265, 223, 142] 14

2.4 Main areas of research of the Roombots project shown for the automatic assembly of a table made of Roombots modules and lightweight passive elements: (1) Locomotion on non-structured ground (off-grid). (2) (On-grid) locomotion through self-reconfiguration. (3) Collaborative manipulation of passive parts. (4) User interfaces. Tiles on the floor are connector ports that can be used to perform on-grid locomotion. Image adapted from [204]. 15

2.5 The hardware implementation of the 8-DoF Lola-OP™ robot with eight degrees of freedom and in-series compliant elements. 16

3.1 Rendered exploded view of a Roombots module. (1) Rapid prototyped shells. (2) ACMs. (3) Passive ports. (4) Outer (diametrical) actuators with 305:1 reduction and absolute encoder. (5) Inner actuator with 366:1 reduction and absolute encoder. 22

3.2 Steps required to sketch a Roombots hemisphere. 24

3.3 Naming convention and main axes in a Roombots module. Each module is composed of four hemispheres connected together by three DoF. Ten sockets are available for active or passive connection ports. 25

3.4 Roombots module approaching a concave corner using its three DoF simultaneously to keep its rightmost connector parallel to the vertical surface at all times. Image from [204]. 25

3.5 Active Connection Mechanism (ACM): (a) ACMv4 (more details in Section 5.3). (b) Latches emerging from the connector. 27

3.6 Roombots electronics: active connection mechanism control board (CONN), motor-driver board (MB), power board (PB), batteries (BATT), Bluetooth communication board (BT), and slip ring (SR). Image adapted from [204]. 28

3.7 Bending of a Roombots module. (a) Cantilevered Roombots metamodule under the effect of gravity. (b) Gap between hemispheres caused by the deformation of the main plates. 29

3.8 Finite Element Analysis (FEA), using SolidWorks Simulation, of a main plate under load. (a) Original flat design in glass fiber. (b) Proposed design in aluminum with reinforcement ribs. 30

3.9 Broken out section of the POM ring. (1) H_2 hemisphere. (2) Main plate fixed to H_2 (pink). (3) POM ring (green). (4) Main plate fixed to H_3 (blue). (5) H_3 hemisphere. Colors shown for descriptive purposes only. Only half of a Roombots module is shown. 31

4.1 Exploded view of the Gearbox A design. A DC brushed motor (M) drives a first drive chain of spur gears (S-1-2). A two-stage planetary gear train (3-4-5) further increases the reduction ratio. An internal hollow axis allows electrical slip ring wires (6) to pass through the gearbox. A thin-section ball bearing (B) guarantees smooth motion. Image adapted from [200]. 37

4.2 104 elements necessary to assemble Gearbox A. The labels correspond to the labels of the exploded view in Figure 4.1. Gears (S), (1), and in the assemblies (2), (4), and (5) are off-the-shelf. Annulus gear (3) is 3D printed. 39

4.3 Exploded view of the Gearbox B design (motor not shown). The labels correspond to the labels of the exploded view in Figure 4.1. 40

4.4 95 elements necessary to assemble Gearbox B. 13 additional elements are needed to assemble the POM ring (top-right of the figure). The labels correspond to the labels of the exploded view in Figure 4.1. Gears (S), (1), and in the assembly (2) are off-the-shelf. Gears in the assemblies (4) and (5) are machine-milled in POM. Annulus gear (3) is 3D printed. 40

4.5 Exploded view of the Gearbox C design (motor not shown). The labels correspond to the labels of the exploded view in Figure 4.1. This design also incorporates a 12-bit absolute encoder (E) at the gearbox output. 42

4.6 102 elements (13 for the POM ring). 89 elements necessary to assemble Gearbox C. 13 additional elements are needed to assemble the POM ring. The labels correspond to the labels of the exploded view in Figure 4.1. All the gears are machine-milled in POM. 42

4.7 Bench test setup: (M) DC motor. (T) Temperature probe. (I) Input encoder. (G) Gearbox. (E) Output encoder. (S) Electrical slip ring. (P) Pulley. (R) Rope. Not pictured: payload, regulated power supply, Arduino control board with DC motor shield. 44

4.8 Damage to Gearbox A after failing to hold a payload equivalent to 95% of the nominal torque of the motor: (a) Broken tooth from the second planetary stage. (b) Damage to an $n = 28:12$ gear from the three-stage spur gear set. 46

4.9 Comparison of the four gearbox designs for four selected tests: (a) Temperature increase and (b) current consumption during continuous rotation with no output load tests. The spikes are noise in the sensor readings. (c) Current consumption for the ±360 deg oscillation with no load tests. (d) Current consumption for 20 cycles of payload holding (2.5 Nm). From the plot we can infer that Gearbox B has more internal friction. 46
4.10 Results from the stress test of Gearbox D: six continuous rotation with no load tests, 5 oscillation with no load tests, and 5 load-holding tests. Units are: [degrees] for relative, absolute, and command positions, [A] for the current, and [°C] for the temperature. 47

5.1 Actuation schematics of an ACMv3 unit. (1) DC brushed motor with 150:1 reduction. (2) Spur gear ($n = 10$). (3) Spur gear ($n = 28 : 12$). (4) Spur gear ($n = 50$). (5) Eccentric disk. (6) Crank drive. (7) Slider. (8) Moving axis. (9) Latch. (10) Fixed axis. (11) Connector board. (12) Position sensor. Part names as in [200]. Target connector greyed out at the bottom. Image adapted from [200]. 52
5.2 Crank-slider mechanism (top view) in four states: (a) Fully retracted. (b) Partially retracted. (c) Fully protracted. (d) Self-locked. Image from [200]. 53
5.3 Improvement of the scooping movement of the connection mechanism: (a) ACMv3. (b) ACMv4. In green: own's connector plate. In pink: target connector. For clarity, only one of the four latches is shown. 54
5.4 Hybrid ACMs. Top connector in cross sectional view. The central magnet is used for passive guidance only. This version of the ACM is gendered because we used magnets with axial magnetization. 55

6.1 Use of passive parts to increase mechanical stability and reduce the weight of a structure. The leftmost structure is built using 8 Roombots modules. The others combine 4 Roombots modules and 2 passive parts. 61
6.2 Types of compliance 64
6.3 Types of in-series compliance explored in this work: (a) "Omnidirectional" compliance. (b) Torsional compliance. 66
6.4 Compliant elements used for the experiments: (a) Five compliant rods made in POM. (b) Top: torsional spring construction for TS_1–TS_4, using off-the-shelf springs; bottom: torsional spring for TS_5, with a custom designed POM spring. Image from [224]. 67
6.5 CAD model of a segment of the Lola-OP™ robot, with a broken-out view of a C3-type compliant element: (1) Dynamixel AX-12A® actuators. (2) Bioloid Flange F2. (3) Machined POM parts. (4) Super-elastic Nitinol wire (for C1, C2, and C3) or POM rod (for C4). (5) Bioloid Flange F3. Dimensions are expressed in mm. Image from [225]. 70
6.6 Simulation of compliant 8-DoF Lola-OP™ robots under the effect of gravity. The first module is fixed to a frame. All servos are locked at zero position to show only the bending of the compliant elements. Image adapted from [225]. 71

7.1 Examples of Roombots CPG topologies with one oscillator per degree of freedom and nearest-neighbor coupling. Each structure has 26 parameters subject to optimization: 9 amplitudes, 9 offsets, and 8 phase lags. 80

8.1 Example of application of the BLF algorithm: (a) A quadruped structure composed of 9 Roombots modules. (b) Corresponding graph representation. (c) Detected body and limbs. (d) Articulations detected using the set of rules described herein. Image from [17]. 82

8.2 Step by step application of the body-limb finder algorithm on *struct10* (a), a structure with 10 Roombots modules used for our experimental validation. (b) The structure is converted into an indirected graph in which each node represents a module and each edge represents a physical connection. (c) Detection of bi-connected components (bcc) in the graph and sorting depending on the number of nodes they contain. (d) Cycles found in the previous step (i.e. bcc containing more than 2 nodes) are tested to find body parts (bcc4 is one). (e) Detection of articulation nodes (marked with an A) for the remaining bcc. (f, g) Articulation joints are tested to check whether they are still part of the body (if the removal of each articulation joint leads to more than two sub-graphs). The two bcc2 are not part of the body. Image from [17]. 85

8.3 Test structures used to validate the algorithm: (a) *quad5-sym*, a quadrupedal structure composed of 5 Roombots modules with symmetric limbs. (b) *quad5-unsym*, similar to *quad5-sym*, but one of the limbs is not symmetric since it is connected to the body with a different orientation. (c) *struct10*, a pseudo random structure of 10 modules with no symmetric limbs. Image adapted from [17]. 88

8.4 The mean value of the fitness function over twenty optimizations for *quad5-sym* (a), *quad5-unsym* (b), and *struct10* (c). Results are displayed in semi-log scale. Images adapted from [17]. 92

8.5 In black, mean value of the fitness function for the four best FO optimization runs of the *struct10* structure. In red, mean fitness value of four randomly chosen optimization solutions in the BLF case. Image adapted from [17]. . . . 93

8.6 The mean fitness values and standard deviation at iteration 5, 25, 50, and 100 for the different network topologies applied to *quad5-sym* (a) and *quad5-unsym* (b). Images from [17]. 94

9.1 Hybrid optimization is a cyclic process combining (1) offline optimization, (2) exploration of control parameters found in simulation on hardware, and (3) meta-optimization to improve matching of software models and hardware. . . 100

9.2 Test arena used to run the hardware experiments. A 2×2m-wide area is in the field of view of a Microsoft Kinect mounted overhead. A control computer sends sets of CPG parameters to the robot and tracks the Roombots's projection over time. Image adapted from [130]. 102

9.3 Similarity measured as overlap between the robots' projections onto the ground as observed by a virtual and a hardware Kinect. Image adapted from [130]. . . 104

9.4 Snapshots of a Roombots metamodule gait in hardware (top) and simulation (bottom) after a cycle of hybrid optimization. Similarity ratio R is about 0.4. Time between snapshots is 12 seconds. Images from [130]. 105

10.1 Two Roombots modules interconnected with a compliant rod. Image adapted from [224]. 111

10.2 CPG topology used to control two Roombots modules with in-series compliant elements. The topology is similar to the one used to control a Roombots metamodule: each joint is represented by one oscillator and all oscillators have nearest-neighbor coupling. Image from [224]. 113

10.3 Results of the experiments with the Compliant Rod (CR) elements: (a) Mean robot speed for each type of pattern elements. The error bars represent the standard deviation for 10 repetitions. The circled data points are the direct result of on-line optimization. (b) Maximum robot speed for each level of compliance. The speed in body lengths per second was calculated using a length of five Roombots units ($0.55m$). Points two and four are not the result of a dedicated optimization but obtained by running the *medium pattern* CPG parameters. Images from [224]. 116

10.4 Results of the experiments with the Torsional Spring (TS) elements: (a) Mean robot speed for each type of pattern elements. The error bars represent the standard deviation for 10 repetitions. The circled data points are the direct result of on-line optimization. (b) Maximum robot speed for each level of compliance. The speed in body lengths per second was calculated using a length of four Roombots units ($0.44m$). Points two and four are not the result of a dedicated optimization. Point two was obtained by running the *soft pattern*; point four was obtained by running the *medium pattern* CPG parameters. Images from[224].116

10.5 Simulated model of the 8-DoF Lola-OP™ robot with in-series compliant elements. Image from [226]. 118

10.6 Detail of the CAD model of the simulated Lola-OP™ robot with in-series compliant elements. Only two modules and one compliant element are shown. Dimensions are expressed in mm. Image from [226]. 120

10.7 Simulation of compliant 8-DoF Lola-OP™ robots with eight different levels of stiffness, under the effect of gravity. All servos are blocked at 0 position to show solely the bending of the compliant elements. The leftmost module is rigidly fixed to a frame. Image adapted from [226]. 121

10.8 Topology of the CPG network implemented for the Lola-OP™ robot. Even numbered DoFs are coupled together. Same goes for odd numbered DoF. A bidirectional coupling between modules m_0 and m_1 allows to adjust the phase shift between the two orthogonal directions of motion. Image from [226]. . . . 123

10.9 Grid search for the rolling gait on the four terrains: (left) Speed. (right) Cost of Transport. Each value is the average of five repetitions of the same experiment (with random initial rotation). Images adapted from [226]. 126

10.10 Speed and CoT results for sidewinding gait on FLAT and RT1. Each point is the average of 20 repetitions of the best particle from each PSO optimization. Images adapted from [226]. 127

10.11 Speed and CoT results for sidewinding gait on RT2 and RT3. Each point is the average of 20 repetitions of the best particle from each PSO optimization. Images adapted from [226]. 128

10.12 Validation of optimization results: speed of the hardware robot sidewinding gaits on flat ground (in bold) compared to simulation runs on FLAT. Image adapted from [226]. 130

10.13 Snapshots from hardware experiments. Rolling gait with *"soft"* elements (left) and *"stiff"* elements (right). Locomotion frequency is 0.5 Hz. Vertical lines are 1 m apart. Images from [226]. 131

10.14 Snapshots from hardware experiments. Sidewinding gait with *"soft"* (top) and *"stiff"* elements (bottom) at 0.5 Hz (left) and 1 Hz (right). Vertical lines are 1 m apart. Images from [226]. 131

10.15 Compliant 8-DoF Lola-OP™ robot locomotion on horizontal pipe environment. 132

10.16 Hardware implementation of the Lola-OP™ robot with eight degrees of freedom and in-series compliant elements. Robots are shown from stiff (left) to compliant (right). Image from [225]. 133

10.17 Front and lateral views of the three pipes used to study the effect of in-series compliance on snake robot locomotion. 8-DoF Lola-OP™ robots shown for size comparisons. Images from [225]. 135

10.18 Diametrical rough terrain, obstacle, and continuous obstacle (as defined in [143]) added to the medium pipe (P2). Dimensions are in meters. Image from [225]. 136

10.19 CAD dimensions of the complete terrain used to show the adaptability of the robots to changes in pipe diameter. Dimensions are in meters. Image from [225]. 136

10.20 Experimental setup equipped with 14 infrared motion capture cameras (MoCap) and a current probe and digital oscilloscope for measuring the robot's power consumption. 138

10.21 Locomotion speed for 7 values of amplitude, for each level of compliance on the small pipe (P1). 105 degrees is the maximum range allowed by the actuators. Each parameter was tested 5 times. Image from [225]. 139

10.22 Locomotion speed for 6 values of amplitude, for each level of compliance on the medium pipe (P2). The stiff robot is not able to run with amplitude values greater than 75 degrees. Each parameter was tested 5 times. Image from [225]. 141

10.23 Locomotion speed for 5 values of amplitude, for each level of compliance on the large pipe (P3). Each parameter was tested 5 times. Image from [225]. . . . 141

10.24 Speed of locomotion (thin lines) and current (tick lines) over time for three compliant robots while crossing the pipe P2 with bumps. C1 is not reported because the robot could not cross the continuous obstacle. Image from [225]. 143

10.25 Snapshots of a C3 compliant robot successfully crossing all the features of our pipe terrain. Images from [225]. 144

11.1 Artist illustration of the Plug-n-Play Robotic Elements being used in a household to increase quality of life. Image courtesy of the Biorobotics Laboratory, EPFL. 150

11.2 Examples of external attachments being manipulated by Plug-n-Play Robotic Elements: (left) Re-orientable speaker. (center and right) Smart lighting system. 151

11.3 Concept of a smart patio with plug-n-play robotic lighting elements and reconfigurable robotic furniture. Rendering created by Augustin Mercier. 151

11.4 Examples of standard pieces of furniture augmented with Plug-n-Play Robotic Elements: IKEA LACK table (a) and wooden box (b) equipped with four passive connection ports and four active Roombots modules connected to them. . . . 152

11.5 Snapshots of an IKEA LACK table augmented using four Roombots modules. The robots are using their continuous-rotation capability to act as wheels. The augmented table is manually operated and able to move across the room and lean over to drop carried objects inside a box. 152

11.6 Snapshots of an augmented wooden box performing articulated locomotion. The gait was first optimized in simulation (left side of the snapshots) and then transferred to the robots (right side). The wooden box is equipped with four Roombots modules used as legs. Each Roombots module is also connected to passive rubber feet extensions used to increase ground clearance. 153

11.7 Demonstration of the smart lighting system: (a) Spotlight external attachment being manipulated by a Roombots metamodule connected to a ceiling passive port. (b) Example scenario in which the user is able to control the orientation and intensity of the spotlight. Images from [140]. 154

12.1 Vision based lighting control: a Kinect mounted overhead detects user's hand gestures which are converted into robot control signals. By placing the arm at a height h_1 the user can redirect the spotlight. By raising the arm at a height h_2, the user can control the light brightness. 156

12.2 Vision based lighting control: (a and b) Spotlight directionality control. (c and d) Spotlight brightness control. Images from [140]. 157

12.3 Natural Roombots User Interface setup. Pointing gestures of the user are recognized by one Kinect. A second Kinect tracks the grid state. Visual feedback is given using LEDs in the Roombots modules and in the connector ports. 158

12.4 RGB LED illumination: (a) Grid tiles LEDs in full intensity, showing different combinations of colors. (b and c) LEDs on the Roombots's diametrical degrees of freedom. Images from [147]. 159

12.5 Natural Roombots User Interface in action to control two modules (only of of them equipped with LEDs). (a) Roombots module 1 is selected. (b) Target connector port is selected. Module 1 starts an on-grid locomotion sequence to reach the target. (c) Roombots module 2 is selected. (d) Target connector port for module 2 is selected. Meanwhile, module 1 is still moving toward the target. (e) Module 1 stops because module 2 obstructs its path. (f) After module 2 reaches its target, module 1 and its target are selected again. (g) Module 1 resumes its locomotion sequence. (h) Both Roombots modules reach their targets. Images from [147]. 159

B.1 A single Roombots module overcoming a concave edge. (a) Initial configuration with the module placed in its SRZ configuration, one grid unit away from the wall. (b) The module approaches the wall using inverse kinematics. (c) Connection to the vertical plate (first row) and disconnection from the horizontal one. (d) The module approaches the wall with a simple rotation of a diametrical DoF. (e) Connection to a passive port on the second row and disconnection from the first row. (f) The module escapes the concave corner upwards using inverse kinematics. Images from [204]. 174

B.2 Single Roombots module on-grid locomotion in 3D experiment: (a) The sequence of 14 atomic moves used to climb the wall. IK: inverse kinematic based motion, R move to the right, U move up, L move to the left, in the global coordinate system. A connection and disconnection phase took place between each move. (b) Detail of four atomic moves. Images from [204]. 175

B.3 A Roombots module overcomes a convex edge by joining with a helper module into a Roombots metamodule. (a) Module 1 (M1) is one grid unit away from the wall. Module 2 (M2) waits on the top surface. (b) M1 climbs the wall for one grid unit. (c) M2 connects its ACM to the approaching M1. (d) M1 is lifted onto the horizontal plane. Images from [204]. 176

C.1 Locomotion of a quadrupedal structure made of five Roombots modules. . . . 177

D.1 Roombots Self-Reconfiguration from Tripedal to Snake Morphology. 180

List of Tables

3.1 Hardware specifications of a Roombots module. 23

4.1 Summary of the four Roombots gearbox versions. Test torque based on the results presented in Section 4.3. Easiness of production, easiness of design, and cost evaluations are qualitative, based on own experience. Gearbox A is not easy to produce because the off-the-shelf (ots) components have to be manually modified to fit the design. Some of these issues have been addressed in Gearbox B with custom designed milled parts. The double support design in Gearbox C makes the assembly much easier. Gearbox D has the production and design advantages of the previous versions; however the metal design makes the machining process much harder, significantly increasing the overall production cost. 43

6.1 Compliant elements specifications. $k_{CR,F}$: flexural stiffness (from Equation 6.1); $k_{CR,T}$: beam's torsional stiffness (from Equation 6.2); k_{TS}: torsional spring stiffness (from datasheet (TS_1–TS_4)) or FEA (TS_5)). 68

6.2 Compliant elements specifications. 70

8.1 Boundaries for the amplitude parameter depending on the type of articulation considered. 86

8.2 Number of optimized network parameters for the three case structures in the four different conditions. Numbers in brackets correspond to previously defined networks; these cases are optimized only once. 89

8.3 Particle Swarm Optimization settings for the different structures. 90

8.4 CPG networks for the three case structures and different conditions. In the FO case, circles represent generic oscillators. For the BLF and BLF-SYM cases, the limbs are represented in green, the body in orange, and the shape coding is as follows: spine oscillator are circles, hip oscillators are squares, knee oscillators are hexagons, and ankle oscillators are crosses. For the BLF-SYM and SYM cases, symmetric oscillators have similar stripe pattern. 91

9.1 Simulation and model parameters subject to meta-optimization. Units follow the Webots standards. 103

9.2 Improvement of similarity ratio R between simulated and hardware robot after meta-optimization. The training set corresponds to the 6 gaits used for meta-optimization. The remaining 11 selected gaits are part of the test set. 105

10.1 Open CPG parameters used for the on-line PSO optimization of two Roombots modules with in-series compliant elements. 113
10.2 Characteristics of the terrains used for the snake robot locomotion with in-series compliant elements. The maximum slope is computed as $tan^{-1}(\frac{h_{max}-h_{min}}{spacing})$.120
10.3 Open CPG parameters for the two gaits considered in this work. Rolling parameters have been tested using a grid search; we used Particle Swarm Optimization for sidewinding. All values are expressed in radians. 124
10.4 Hardware runs of the Lola-OP™ robot on flat ground. Each experiment was repeated 5 times. 129
10.5 Specifications of the experimental setup used for the locomotion experiments on horizontal pipes. 140

A.1 List of 111 modular robots, from 1988 until 2015. Data from [204] and [2], integrated with our best knowledge on most recent publications. 169

Bibliography

[1] B. B. P. Adams. “Evolutionary, developmental neural networks for robust robotic control”. PhD thesis. Massachusetts Institute of Technology, 2006.

[2] H. Ahmadzadeh and E. Masehian. “Modular robotic systems: Methods and algorithms for abstraction, planning, control, and synchronization”. In: *Artificial Intelligence* 223 (June 2015), pp. 27–64.

[3] R. M. Alexander. “Three Uses for Springs in Legged Locomotion”. In: *The International Journal of Robotics Research* 9.2 (Apr. 1, 1990), pp. 53 –61.

[4] B. K. An. “Em-cube: cube-shaped, self-reconfigurable robots sliding on structure surfaces.” In: *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on.* 2008, pp. 3149–3155.

[5] S. Aoi, H. Sasaki, and K. Tsuchiya. “A Multilegged Modular Robot That Meanders: Investigation of Turning Maneuvers Using Its Inherent Dynamic Characteristics”. In: *SIAM Journal on Applied Dynamical Systems* 6.2 (Jan. 2007), pp. 348–377.

[6] *Ax-12+ Half Duplex USART Communication.* URL: http://savageelectrtonics.blogspot.com/.

[7] T. Baba et al. “A snake robot propelling inside of a pipe with helical rolling motion”. In: *SICE Annual Conference 2010, Proceedings of.* 2010, pp. 2319–2325.

[8] J. Baca, M. Ferre, and R. Aracil. “A heterogeneous modular robotic design for fast response to a diversity of tasks”. In: *Robotics and Autonomous Systems* 60.4 (2012), pp. 522–531.

[9] E. Badri. “Elasticity compensation using explicit learning”. MA thesis. Biorob, EPFL, 2011.

[10] R. Belisle, C.-h. Yu, and R. Nagpal. “Mechanical design and locomotion of modular-expanding robots”. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2010* (2010), pp. 17–23.

[11] A. A. Biewener. “Muscle-tendon stresses and elastic energy storage during locomotion in the horse”. In: *Comparative Biochemistry and Physiology Part B: Biochemistry and Molecular Biology* 120.1 (May 1998), pp. 73–87.

[12] *Bioloid Comprehence Kit.* URL: http://www.robotis.com.

[13] J Bishop et al. “Programmable parts: A demonstration of the grammatical approach to self-organization”. In: *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on.* IEEE. 2005, pp. 3684–3691.

[14] H. Bojinov, A. Casal, and T. Hogg. “Emergent structures in modular self-reconfigurable robots”. In: *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on.* Vol. 2. IEEE. 2000, pp. 1734–1741.

[15] S. Bonardi et al. “Collaborative manipulation and transport of passive pieces using the self-reconfigurable modular robots roombots”. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* Nov. 2013, pp. 2406–2412.

[16] S. Bonardi. “Control and interaction strategies for self-reconfigurable modular robots”. PhD thesis. 2014.

[17] S. Bonardi et al. “Automatic generation of reduced CPG control networks for locomotion of arbitrary modular robot structures”. In: *Proceedings of Robotics: Science and Systems.* 2014.

[18] S. Bonardi et al. “Design and evaluation of a graphical iPad application for arranging adaptive furniture”. In: *RO-MAN, 2012 IEEE.* 2012, pp. 290–297.

[19] S. Bonardi et al. “Locomotion through Reconfiguration based on Motor Primitives for Roombots Self-Reconfigurable Modular Robots”. In: *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on.* 2012, pp. 1–6.

[20] J Bongard and H. Lipson. “Once more unto the breach: Co-evolving a robot and its simulator”. In: *Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9).* 2004, pp. 57–62.

[21] J. Bongard, V. Zykov, and H. Lipson. “Resilient machines through continuous self-modeling”. In: *Science* 314.5802 (2006), pp. 1118–1121.

[22] D. Brandt and D. J Christensen. “A new meta-module for controlling large sheets of ATRON modules”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007. IROS 2007.* Oct. 29, 2007, pp. 2375–2380.

[23] D. Brandt, D. J Christensen, and H. H Lund. “ATRON Robots: Versatility from Self-Reconfigurable Modules”. In: *International Conference on Mechatronics and Automation, 2007. ICMA 2007.* Aug. 5, 2007, pp. 26–32.

[24] L. Brodbeck, S. Hauser, and F. Iida. “Morphological Evolution of Physical Robots through Model-Free Phenotype Development”. In: *PLoS ONE* 10.6 (June 2015), e0128444.

[25] L. Brodbeck and F. Iida. “An extendible reconfigurable robot based on hot melt adhesives”. In: *Autonomous Robots* 39.1 (2015), pp. 87–100.

[26] R. A. Brooks. “Artifical life and real robots”. In: *Toward a practice of autonomous systems: Proceedings of the First European Conference on Artificial Life.* MIT press. 1992, pp. 3–10.

[27] A. Brunete, M. Hernando, and E. Gambao. “Modular multiconfigurable architecture for low diameter pipe inspection microrobots”. In: *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on.* IEEE. 2005, pp. 490–495.

[28] K. Caluwaerts et al. “Design and control of compliant tensegrity robots through simulation and hardware validation”. In: *Journal of The Royal Society Interface* 11.98 (Sept. 6, 2014), p. 20140520.

[29] B. L. Carmichael and C. M. Gifford. “Modeling and Simulation of the Seismic TETwalker Concept”. In: *Center for Remote Sensing of Ice Sheets, University of Kansas, KS, Tech. Rep., CReSIS TR* 134 (2007).

[30] G. Carpino et al. “A Novel Compact Torsional Spring for Series Elastic Actuators for Assistive Wearable Robots”. In: *Journal of Mechanical Design* 134.12 (Oct. 19, 2012), pp. 121002–121002.

[31] A. Castano, A. Behar, and P. Will. “The Conro modules for reconfigurable robots”. In: *IEEE/ASME Transactions on Mechatronics* 7.4 (Dec. 2002), pp. 403–409.

[32] T. Chenal et al. “Variable stiffness fabrics with embedded shape memory materials for wearable applications”. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014).* Sept. 2014, pp. 2827–2831.

[33] G. Chirikjian. “Kinematics of a metamorphic robotic system”. In: *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on.* 1994, pp. 449–455.

[34] O. Chocron. “Evolving modular robots for rough terrain exploration”. In: *Mobile Robots: The Evolutionary Approach.* Springer, 2007, pp. 23–46.

[35] D. J. Christensen, J. Campbell, and K. Stoy. “Anatomy-based organization of morphology and control in self-reconfigurable modular robots”. In: *Neural Computing and Applications* 19.6 (Sept. 1, 2010), pp. 787–805.

[36] D. J. Christensen, U. P. Schultz, and K. Stoy. “A distributed and morphology-independent strategy for adaptive locomotion in self-reconfigurable modular robots”. In: *Robotics and Autonomous Systems* (2013).

[37] D. J. Christensen, A. Spröwitz, and A. J. Ijspeert. “Distributed online learning of central pattern generators in modular robots”. In: *From Animals to Animats 11.* Springer, 2010, pp. 402–412.

[38] D. Christensen, J. Larsen, and K. Stoy. “Adaptive strategy for online gait learning evaluated on the polymorphic robotic LocoKit”. In: *2012 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS).* May 2012, pp. 63 –68.

[39] *Codyn – Open source software framework for modeling and simulating Coupled Dynamical Systems.* URL: http://www.codyn.net/.

[40] G. S. Copen and T. C. Mowry. “Claytronics: An instance of programmable matter”. In: *Wild and Crazy Ideas Session of ASPLOS* (2004).

[41] S. Coros et al. “Locomotion skills for simulated quadrupeds”. In: *ACM Transactions on Graphics (TOG)*. Vol. 30. 4. ACM. 2011, p. 59.

[42] A. Crespi and A. Ijspeert. “Online optimization of swimming and crawling in an amphibious snake robot”. In: *Robotics, IEEE Transactions on* 24.1 (2008), pp. 75–87.

[43] M. D’Angelo, B. Weel, and A. E. Eiben. “HyperNEAT Versus RL PoWER for Online Gait Learning in Modular Robots”. In: *Applications of Evolutionary Computation*. Ed. by A. I. Esparcia-Alcázar and A. M. Mora. Lecture Notes in Computer Science 8602. Springer Berlin Heidelberg, Apr. 23, 2014, pp. 777–788.

[44] M. D’Angelo, B. Weel, and A. Eiben. “Online gait learning for modular robots with arbitrary shapes and sizes”. In: *Theory and Practice of Natural Computing*. Springer, 2013, pp. 45–56.

[45] S. Dastoor and M. Cutkosky. “Variable impedance due to electromechanical coupling in electroactive polymer actuators”. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sept. 2011, pp. 774–779.

[46] J. Davey, N. Kwok, and M. Yim. “Emulating self-reconfigurable robots - design of the SMORES system”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Oct. 2012, pp. 4464 –4469.

[47] J. Davey et al. “ModLock: A manual connector for reconfigurable modular robots.” In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2012, pp. 3217–3222.

[48] Dennis Krupke et al. “Printable modular robot: an application of rapid prototyping for flexible robot design”. In: *Industrial Robot: An International Journal* 42.2 (Mar. 2015), pp. 149–155.

[49] B. R. Donald et al. “An untethered, electrostatic, globally controllable MEMS microrobot”. In: *Microelectromechanical Systems, Journal of* 15.1 (2006), pp. 1–15.

[50] M. Eaton. *Evolutionary Humanoid Robotics*. Springer, 2015.

[51] F. Enner, D. Rollinson, and H. Choset. “Motion estimation of snake robots in straight pipes”. In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. 2013, pp. 5168–5173.

[52] J. A. Escalera et al. “ROBMAT: Teleoperation of a modular robot for collaborative manipulation”. In: *Knowledge-Based Intelligent Information and Engineering Systems*. Springer. 2007, pp. 1204–1213.

[53] A Faiña et al. “First steps towards a heterogeneous modular robotic architecture for intelligent industrial operation”. In: *Workshop Reconfigurable Modular Robotics: Challenges of Mechatronic and Bio-Chemo-Hybrid Systems, IROS*. 2011, p. 6.

[54] A. Faíña et al. “EDHMoR: Evolutionary designer of heterogeneous modular robots”. In: *Engineering Applications of Artificial Intelligence* 26.10 (2013), pp. 2408–2423.

[55] J. Florez et al. “Soft pneumatic actuators for legged locomotion”. In: *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. Dec. 2014, pp. 27–34.

[56] G. Francesca et al. “An experiment in automatic design of robot swarms”. In: *Swarm Intelligence*. Springer, 2014, pp. 25–37.

[57] T. Fukuda et al. “Self Organizing Robots Based on Cell Structures - CEBOT”. In: *, IEEE International Workshop on Intelligent Robots, 1988*. Oct. 1988, pp. 145–150.

[58] K. C Galloway, R. Jois, and M. Yim. “Factory floor: A robotically reconfigurable construction platform”. In: *2010 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 3, 2010, pp. 2467–2472.

[59] K. C Galloway et al. “Experimental investigations into the role of passive variable compliant legs for dynamic robotic locomotion”. In: *2011 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 9, 2011, pp. 1243–1249.

[60] K. C. Galloway, J. E. Clark, and D. E. Koditschek. “Design of a Tunable Stiffness Composite Leg for Dynamic Locomotion”. In: *ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (Jan. 2009), pp. 215–222.

[61] R. F. M. Garcia et al. “A vacuum-based bonding mechanism for modular robotics”. In: *Robotics, IEEE Transactions on* 27.5 (2011), pp. 876–890.

[62] J. Germann, B. Schubert, and D. Floreano. “Stretchable electroadhesion for soft robots”. In: *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. Ieee. 2014, pp. 3933–3938.

[63] J. Germann et al. “Active connection mechanism for soft modular robots”. In: *Advanced Robotics* 26.7 (2012), pp. 785–798.

[64] K. Gilpin and D. Rus. “Modular Robot Systems”. In: *IEEE Robotics & Automation Magazine* 17.3 (Sept. 2010), pp. 38–55.

[65] K. Gilpin, A. Knaian, and D. Rus. “Robot pebbles: One centimeter modules for programmable matter through self-disassembly”. In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE. 2010, pp. 2485–2492.

[66] K. Gilpin et al. “Miche: Modular shape formation by self-disassembly”. In: *The International Journal of Robotics Research* 27.3-4 (2008), pp. 345–372.

[67] K. Glette et al. “Evolution of locomotion in a simulated quadruped robot and transferral to reality”. In: *Proceedings of the Seventeenth International Symposium on Artificial Life and Robotics*. 2012, pp. 1–4.

[68] J González-Gómez, E Aguayo, and E Boemo. “Locomotion of a Modular Worm-like Robot using a FPGA-based embedded MicroBlaze Soft-processor”. In: *Climbing and Walking Robots*. Springer, 2005, pp. 869–878.

[69] S. T. Griffith. “Growing machines”. PhD thesis. Massachusetts Institute of Technology, 2004.

[70] E. Guan et al. “Self-reconfiguration path planning design for M-lattice robot based on genetic algorithm”. In: *Intelligent Robotics and Applications*. Springer, 2011, pp. 505–514.

[71] E. Haasdijk, A. A. Rusu, and A. Eiben. "HyperNEAT for locomotion control in modular robots". In: *Evolvable systems: from biology to hardware.* Springer, 2010, pp. 169–180.

[72] B. von Haller, A. Ijspeert, and D. Floreano. "Co-evolution of Structures and Controllers for Neubot Underwater Modular Robots". In: *Advances in Artificial Life.* Ed. by M. S. Capcarrère et al. Red. by D. Hutchison et al. Vol. 3630. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 189–199.

[73] R. Ham et al. "Compliant actuator designs". In: *IEEE Robotics & Automation Magazine* 16.3 (Sept. 2009), pp. 81–94.

[74] G. J. Hamlin and A. C. Sanderson. "Tetrobot modular robotics: Prototype and experiments". In: *Intelligent Robots and Systems' 96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on.* Vol. 2. IEEE. 1996, pp. 390–395.

[75] S. Hirose and H. Yamada. "Snake-like robots [Tutorial]". In: *Robotics Automation Magazine, IEEE* 16.1 (2009), pp. 88–98.

[76] S. Hirose, T. Shirasu, and E. F. Fukushima. "Proposal for cooperative robot "Gunryu" composed of autonomous segments". In: *Robotics and Autonomous Systems* 17.1 (1996), pp. 107–118.

[77] W. Hong, S. Wang, and D. Shui. "Reconfigurable robot system based on electromagnetic design". In: *Fluid Power and Mechatronics (FPM), 2011 International Conference on.* IEEE. 2011, pp. 570–575.

[78] K. Hosokawa et al. "Self-organizing collective robots with morphogenesis in a vertical plane". In: *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on.* Vol. 4. IEEE. 1998, pp. 2858–2863.

[79] S. G. M. Hossain, C. A. Nelson, and P. Dasgupta. "Hardware Design and Testing of ModRED: A Modular Self-Reconfigurable Robot System". In: *Advances in Reconfigurable Mechanisms and Robots I.* Ed. by J. S. Dai, M. Zoppi, and X. Kong. Springer London, 2012, pp. 515–523.

[80] J. W. Hurst. *The Role and Implementation of Compliance in Legged Locomotion.* ProQuest, 2008. 101 pp.

[81] A. J. Ijspeert. "Central pattern generators for locomotion control in animals and robots: A review". In: *Neural Networks* 21.4 (May 2008), pp. 642–653.

[82] A. J. Ijspeert et al. "From Swimming to Walking with a Salamander Robot Driven by a Spinal Cord Model". In: *Science* 315.5817 (Mar. 9, 2007), pp. 1416–1420.

[83] N. Inou, K. Minami, and M. Koseki. "Group robots forming a mechanical structure-development of slide motion mechanism and estimation of energy consumption of the structural formation". In: *Computational Intelligence in Robotics and Automation, 2003. Proceedings. 2003 IEEE International Symposium on.* Vol. 2. July 2003, 874 –879 vol.2.

[84] N. Inou, H. Kobayashi, and M. Koseki. “Development of pneumatic cellular robots forming a mechanical structure”. In: *Control, Automation, Robotics and Vision, 2002. ICARCV 2002. 7th International Conference on.* Vol. 1. IEEE. 2002, pp. 63–68.

[85] N. Jakobi. “Evolutionary robotics and the radical envelope-of-noise hypothesis”. In: *Adaptive behavior* 6.2 (1997), pp. 325–368.

[86] M. Kalontarov et al. “Hydrodynamically driven docking of blocks for 3D fluidic assembly”. In: *Microfluidics and Nanofluidics* 9.2-3 (2010), pp. 551–558.

[87] A. Kamimura et al. “Automatic locomotion pattern generation for modular robots”. In: *Robotics and Automation, 2003. Proceedings. ICRA’03. IEEE International Conference on.* Vol. 1. IEEE, 2003, pp. 714–720.

[88] A. Kawakami et al. “SMC Rover: Planetary Rover with transformable wheels”. In: *Experimental Robotics VIII.* Springer, 2003, pp. 498–506.

[89] V. Kee et al. “Hinged-Tetro: A self-reconfigurable module for nested reconfiguration”. In: *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM).* July 2014, pp. 1539–1546.

[90] J. Kennedy and R. Eberhart. “Particle swarm optimization”. In: *Neural Networks, 1995. Proceedings., IEEE International Conference on.* Vol. 4. IEEE, 1995, pp. 1942–1948.

[91] S. Kernbach et al. “Symbiotic robot organisms: REPLICATOR and SYMBRION projects”. In: *Proceedings of the 8th workshop on performance metrics for intelligent systems.* ACM. 2008, pp. 62–69.

[92] C. Khairallah. *Modular articulated robot structure.* US Patent 6,323,615. 2001.

[93] J. van den Kieboom. “On the dynamics of human locomotion and co-design of lower limb assistive devices”. PhD thesis. EPFL, 2014.

[94] S. Kim et al. “Whole body adhesion: hierarchical, directional and distributed control of adhesive forces for a climbing robot”. In: *2007 IEEE International Conference on Robotics and Automation.* Apr. 2007, pp. 1268–1273.

[95] B. Kirby et al. “Catoms: Moving Robots Without Moving Parts.” In: *Proceedings of the National Conference on Artificial Intelligence.* Vol. 20. 4. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999. 2005, p. 1730.

[96] G. Klaus, K. Glette, and J. Tørresen. “A comparison of sampling strategies for parameter estimation of a robot simulator”. In: *Simulation, Modeling, and Programming for Autonomous Robots.* Springer, 2012, pp. 173–184.

[97] S. Koos, J.-B. Mouret, and S. Doncieux. “Crossing the reality gap in evolutionary robotics by promoting transferable controllers”. In: *Proceedings of the 12th annual conference on Genetic and evolutionary computation.* ACM. 2010, pp. 119–126.

[98] M. Koseki, K. Minami, and N. Inou. “Cellular Robots Forming a Mechanical Structure”. In: *Distributed Autonomous Robotic Systems 6.* Ed. by R. Alami, R. Chatila, and H. Asama. Springer Japan, 2007, pp. 139–148.

[99] K. Kotay et al. “The self-reconfiguring robotic molecule: Design and control algorithms”. In: *Workshop on Algorithmic Foundations of Robotics.* Citeseer. 1998, pp. 376–386.

[100] H. Kurokawa et al. “A 3-d self-reconfigurable structure and experiments”. In: *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on.* Vol. 2. IEEE. 1998, pp. 860–865.

[101] H. Kurokawa et al. “Distributed self-reconfiguration of M-TRAN III modular robotic system”. In: *The International Journal of Robotics Research* 27.3-4 (2008), pp. 373–386.

[102] H. Kurokawa et al. “Motion simulation of a modular robotic system”. In: *IECON-Proceedings.* Vol. 4. 2000, pp. 2473–2478.

[103] H. Kurokawa et al. “Self-reconfigurable modular robot (M-TRAN) and its motion design”. In: *Control, Automation, Robotics and Vision, 2002. ICARCV 2002. 7th International Conference on.* Vol. 1. IEEE. 2002, pp. 51–56.

[104] S. W. Kwok et al. “Magnetic assembly of soft robots with hard components”. In: *Advanced Functional Materials* 24.15 (2014), pp. 2180–2187.

[105] J. Liedke et al. “The collective self-reconfigurable modular organism (cosmo)”. In: *Advanced Intelligent Mechatronics (AIM), 2013 IEEE/ASME International Conference on.* IEEE. 2013, pp. 1–6.

[106] P. Liljeback et al. “Compliant control of the body shape of snake robots”. In: *Robotics and Automation (ICRA), 2014 IEEE International Conference on.* 2014, pp. 4548–4555.

[107] H. Lipson and J. B. Pollack. “Towards continuously reconfigurable self-designing robotics”. In: *Robotics and Automation, 2000. Proceedings. ICRA’00. IEEE International Conference on.* Vol. 2. IEEE, 2000, pp. 1761–1766.

[108] H. Lipson et al. “Evolutionary Robotics for Legged Machines: From Simulation to Physical Reality.” In: *IAS.* 2006, pp. 11–18.

[109] J. Liu et al. “Center-configuration selection technique for the reconfigurable modular robot”. In: *Science in China Series F: Information Sciences* 50.5 (Oct. 2007), pp. 697–710.

[110] G. E. Loeb. “Learning from the spinal cord”. In: *The Journal of physiology* 533.1 (2001), pp. 111–117.

[111] A. Lyder, R. Garcia, and K. Stoy. “Mechanical design of odin, an extendable heterogeneous deformable modular robot”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008.* IEEE, Sept. 22, 2008, pp. 883–888.

[112] A. Lyder, R. F. M. Garcia, and K. Stoy. “Genderless connection mechanism for modular robots introducing torque transmission between modules”. In: *Proceedings of the ICRA Workshop on Modular Robots, State of the Art.* 2010, pp. 77–81.

[113] J. Mämpel et al. “A modular robot climbing on pipe-like structures”. In: *Autonomous Robots and Agents, 2009. ICARA 2009. 4th International Conference on.* IEEE. 2009, pp. 87–91.

[114] Y. Mantzouratos et al. “On embeddability of modular robot designs”. In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. 2015, pp. 1911–1918.

[115] D. Marbach and A. J. Ijspeert. “Co-evolution of configuration and control for homogenous modular robots”. In: *Proceedings of the eighth conference on intelligent autonomous systems (IAS8)*. 2004, pp. 712–719.

[116] D. Marbach and A. J. Ijspeert. “Online optimization of modular robot locomotion”. In: *Mechatronics and Automation, 2005 IEEE International Conference*. Vol. 1. IEEE, 2005, pp. 248–253.

[117] A. Mehta, J. DelPreto, and D. Rus. “Integrated Codesign of Printable Robots”. In: *Journal of Mechanisms and Robotics* 7.2 (May 2015), pp. 021015–021015.

[118] K. Melo, M. Hernandez, and D. Gonzalez. “Parameterized Space Conditions for the Definition of Locomotion Modes in Modular Snake Robots”. In: *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*. 2012, pp. 2032–2038.

[119] K. Melo and L. Paez. “Modular snake robot gaits on horizontal pipes”. In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. 2012, pp. 3099–3104.

[120] K. Melo, L. Paez, and C. Parra. “Indoor and outdoor parametrized gait execution with modular snake robots”. In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. 2012, pp. 3525–3526.

[121] K. Melo et al. “Open Source VSA-CubeBots for Rapid Soft Robot Prototyping”. In: *Robot Makers - Workshop in conjunction with 2014 Robotics Science and Systems*. July 12, 2014. Berkeley, California, 2014.

[122] K. Melo et al. “The Modular Snake Robot Open Project: Turning animal functions into engineering tools”. In: *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on*. 2013, pp. 1–6.

[123] K. Melo and L. Paez. “Experimental determination of control parameter intervals for repeatable gaits in modular snake robots”. In: *Safety, Security, and Rescue Robotics (SSRR), 2014 IEEE International Symposium on*. 2014, pp. 1–7.

[124] Y. Meng et al. “Cross-Ball: A new morphogenetic self-reconfigurable modular robot”. In: *2011 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 9, 2011, pp. 267–272.

[125] R. Merali and D. Long. “Actuated responsive truss”. In: *Proc. of ICRA 2010 Workshop Modular Robots: State of the Art*. 2010, pp. 36–40.

[126] F. Meyer, A. Spröwitz, and L. Berthouze. “Passive compliance for an RC servo-controlled bouncing robot”. In: *Advanced Robotics* 20.8 (2006), pp. 953–961.

[127] S. Miyashita, M. Hadorn, and P. E. Hotz. “Water floating self-assembling agents”. In: *Agent and Multi-Agent Systems: Technologies and Applications*. Springer, 2007, pp. 665–674.

[128] S. Mobes et al. "Toward a 2D Modular and Self-Reconfigurable Robot for Conveying Microparts". In: *2012 Second Workshop on Design, Control and Software Implementation for Distributed MEMS (dMEMS)*. Apr. 2012, pp. 7 –13.

[129] *Modular Robotics | MOSS & Cubelets | Robot Toys*. URL: http://www.modrobotics.com/.

[130] R. Moeckel et al. "Gait optimization for roombots modular robots - Matching simulation and reality". In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Nov. 2013, pp. 3265–3272.

[131] R. Moeckel et al. "Exploring adaptive locomotion with YaMoR, a novel autonomous modular robot with Bluetooth interface". In: *Industrial Robot: An International Journal* 33.4 (2006), pp. 285–290.

[132] F. Mondada et al. "Swarm-Bot: A New Distributed Robotic Concept". In: *Autonomous Robots* 17.2 (Sept. 2004), pp. 193–221.

[133] R. Moreno and J. Gomez. "Central pattern generators and hormone inspired messages: A hybrid control strategy to implement motor primitives on chain type modular reconfigurable robots". In: *2011 IEEE International Conference on Robotics and Automation (ICRA)*. May 2011, pp. 1014 –1019.

[134] P. Moubarak and P. Ben-Tzvi. "Modular and reconfigurable mobile robotics". In: *Robotics and Autonomous Systems* 60.12 (Dec. 2012), pp. 1648–1663.

[135] S. Murata, H. Kurokawa, and S. Kokaji. "Self-assembling machine". In: *, 1994 IEEE International Conference on Robotics and Automation, 1994. Proceedings*. IEEE, May 8, 1994, 441–448 vol.1.

[136] S. Murata et al. "Hardware design of modular robotic system". In: *2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings*. Vol. 3. 2000, 2210 –2217 vol.3.

[137] S. Murata, K. Kakomura, and H. Kurokawa. "Toward a scalable modular robotic system". In: *Robotics & Automation Magazine, IEEE* 14.4 (2007), pp. 56–63.

[138] S. Murata and H. Kurokawa. "Prototypes of Self-Organizing Robots". English. In: *Self-Organizing Robots*. Vol. 77. Springer Tracts in Advanced Robotics. Springer Tokyo, 2012, pp. 105–130.

[139] R. R. Murphy. *Disaster Robotics*. The MIT Press, 2014.

[140] M. Mutlu et al. "Natural User Interface for Lighting Control: Case Study on Desktop Lighting Using Modular Robots". Submitted to: 10th ACM International conference on Tangible, Embedded and Embodied Interaction (TEI). 2015.

[141] J. Neubert, A Rost, and H. Lipson. "Self-Soldering Connectors for Modular Robots". In: *IEEE Transactions on Robotics* Early Access Online (2014).

[142] J. Neubert and H. Lipson. "Soldercubes: a self-soldering self-reconfiguring modular robot system". English. In: *Autonomous Robots* (2015), pp. 1–20.

[143] C. Nie, X. Pacheco Corcho, and M. Spenko. “Robots on the Move: Versatility and Complexity in Mobile Robot Locomotion”. In: *Robotics Automation Magazine, IEEE* 20.4 (2013), pp. 72–82.

[144] *Open Dynamics Engine (ODE)*. URL: http://www.ode.org/.

[145] E. H. Østergaard et al. “Hydra: From cellular biology to shape-changing artefacts”. In: *Artificial Neural Networks: Biological Inspirations–ICANN 2005*. Springer, 2005, pp. 275–281.

[146] E. H. Østergaard et al. “Design of the ATRON lattice-based self-reconfigurable robot”. In: *Autonomous Robots* 21.2 (Aug. 25, 2006), pp. 165–183.

[147] A. Özgür et al. “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.

[148] M. Pacheco et al. “Fable: Design of a modular robotic playware platform”. In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE. 2013, pp. 544–550.

[149] M. Pacheco et al. “Fable II: Design of a modular robot for creative learning”. In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE. 2015, pp. 6134–6139.

[150] L. Paez, K. Melo, and C. Parra. “Center of mass displacements using rolling gaits for modular robots on the outside of pipes”. In: *Robotics Symposium, 2011 IEEE IX Latin American and IEEE Colombian Conference on Automatic Control and Industry Applications (LARC)*. 2011, pp. 1–6.

[151] M. Park, S. Chitta, and M. Yim. “Isomorphic gait execution in homogeneous modular robots”. In: *Robitcs: Science and Systems Workshop on Self-reconfigurable Modular Robots, Philadelphia*. 2006.

[152] C. Parrott, T. Dodd, and R. Gross. “HiGen: A high-speed genderless mechanical connection mechanism with single-sided disconnect for self-reconfigurable modular robots”. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*. Sept. 2014, pp. 3926–3932.

[153] J. Paulos et al. “Automated Self-Assembly of Large Maritime Structures by a Team of Robotic Boats”. In: *IEEE Transactions on Automation Science and Engineering* PP.99 (2015), pp. 1–11.

[154] R. Pfeifer and J. Bongard. *How the body shapes the way we think: a new view of intelligence*. MIT press, 2006.

[155] R. Pfeifer, M. Lungarella, and F. Iida. “Self-Organization, Embodiment, and Biologically Inspired Robotics”. In: *Science* 318.5853 (Nov. 16, 2007), pp. 1088 –1093.

[156] *Pneubotics*. URL: http://www.pneubotics.com/.

[157] R. Poli, J. Kennedy, and T. Blackwell. “Particle swarm optimization”. In: *Swarm intelligence* 1.1 (2007), pp. 33–57.

[158] S. Pouya et al. “Automatic Gait Generation in Modular Robots: to Oscillate or to Rotate? that is the question”. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*. 2010, pp. 514–520.

[159] S. Pouya et al. “Locomotion Gait Optimization For Modular Robots; Coevolving Morphology and Control”. In: *Procedia Computer Science* 7 (2011), pp. 320–322.

[160] M. J. Powell. “An efficient method for finding the minimum of a function of several variables without calculating derivatives”. In: *The computer journal* 7.2 (1964), pp. 155–162.

[161] G. A Pratt and M. M Williamson. “Series elastic actuators”. In: *1995 IEEE/RSJ International Conference on Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings*. Vol. 1. IEEE, Aug. 5, 1995, 399–406 vol.1.

[162] G. A. Pratt et al. “Stiffness isn't everything”. In: *Experimental Robotics IV*. Ed. by O. Khatib and J. K. Salisbury. Vol. 223. London: Springer-Verlag, pp. 253–262.

[163] G. Qiao et al. “Design of a self-reconfigurable wireless network system for modular self-reconfigurable robots”. In: *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2012, pp. 1337–1342.

[164] M. H. Raibert. “Trotting, pacing and bounding by a quadruped robot”. In: *Journal of Biomechanics* 23, Supplement 1 (1990), pp. 79–98.

[165] V Ramchurn, R. C. Richardson, and P. Nutter. “ORTHO-BOT: a modular reconfigurable space robot concept”. In: *Climbing and Walking Robots*. Springer, 2006, pp. 659–666.

[166] S. Rasakatla, K. M. Krishna, and B. Indurkhya. “Mod-Leg a modular legged robotic system”. In: *ACM SIGGRAPH 2010 Posters*. ACM. 2010, p. 12.

[167] M. Reis and F. Iida. “An Energy-Efficient Hopping Robot Based on Free Vibration of a Curved Beam”. In: *IEEE/ASME Transactions on Mechatronics* 19.1 (Feb. 2014), pp. 300–311.

[168] D. Rollinson, A. Buchan, and H. Choset. “State estimation for snake robots”. In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. 2011, pp. 1075–1080.

[169] D. Rollinson and H. Choset. “Gait-based compliant control for snake robots”. In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. 2013, pp. 5138–5143.

[170] D. Rollinson et al. “Design and architecture of a series elastic snake robot”. In: *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. 2014, pp. 4630–4636.

[171] D. Rollinson et al. “Torque control strategies for snake robots”. In: *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. 2014, pp. 1093–1099.

[172] D. Rollinson. “Control and Design of Snake Robots”. PhD thesis. CMU, June 1, 2014.

[173] D. Rollinson, A. Buchan, and H. Choset. “Virtual Chassis for Snake Robots: Definition and Applications”. In: *Advanced Robotics* 26.17 (2012), pp. 2043–2064.

[174] J. W. Romanishin, K. Gilpin, and D. Rus. “M-blocks: Momentum-driven, magnetic modular robots”. In: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE. 2013, pp. 4288–4295.

[175] A. Rosendo et al. “Producing alternating gait on uncoupled feline hindlimbs: muscular unloading rule on a biomimetic robot”. In: *Advanced Robotics* 28.6 (Mar. 2014), pp. 351–365.

[176] M. Rubenstein and R. Nagpal. “Kilobot: A Robotic Module for Demonstrating Behaviors in a Large Scale (2^{10} Units) Collective”. In: Institute of Electrical and Electronics Engineers. 2010.

[177] D. Rus et al. “Self-reconfiguring robots”. In: *Communications of the ACM* 45.3 (2002), pp. 39–45.

[178] G. G. Ryland and H. H. Cheng. “Design of iMobot, an intelligent reconfigurable mobile robot with novel locomotion”. In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE. 2010, pp. 60–65.

[179] H. Sadjadi et al. “Design and implementation of HexBot: A modular self-reconfigurable robotic system”. In: *Journal of the Franklin Institute* 349.7 (2012), pp. 2281–2293.

[180] B. Salemi, M. Moll, and W. Shen. “SUPERBOT: A deployable, multi-functional, and modular self-reconfigurable robotic system”. In: *Proc. 2006 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*. Citeseer, 2006.

[181] J. Sastra et al. “A biologically inspired dynamic legged locomotion with a modular reconfigurable robot”. In: *In Proc. of DSCC ASME Dynamic Systems and Control Conference, Ann Arbor, Michigan, USA* (2008).

[182] J. Sastra. “Using Modular Reconfigurable Robots for Rapid Development of Dynamic Locomotion Experiments”. PhD thesis. University of Pennsylvania, 2012.

[183] J. Sastra, S. Chitta, and M. Yim. “Dynamic Rolling for a Modular Loop Robot”. In: *The International Journal of Robotics Research* 28.6 (June 1, 2009), pp. 758 –773.

[184] J. Sastra, S. Revzen, and M. Yim. “Softer legs allow a modular hexapod to run faster”. In: *Climbing and Walking Robots (CLAWAR)*. 2012.

[185] T. Sato, T. Kano, and A. Ishiguro. “A decentralized control scheme for an effective coordination of phasic and tonic control in a snake-like robot”. In: *Bioinspiration & biomimetics* 7.1 (2012), p. 016005.

[186] T. Sato, T. Kano, and A. Ishiguro. “On the applicability of the decentralized control mechanism extracted from the true slime mold: a robotic case study with a serpentine robot”. In: *Bioinspiration & biomimetics* 6.2 (2011), p. 026006.

[187] D. Schmitz, P. Khosla, and T. Kanade. *The CMU reconfigurable modular manipulator system*. Tech. rep. CMU, 1988.

[188] E. Schweikardt. “Modular robotics studio”. In: *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction.* TEI ’11. New York, NY, USA: ACM, 2011, pp. 353–356.

[189] E. Schweikardt and M. D. Gross. “roBlocks: a robotic construction kit for mathematics and science education”. In: *Proceedings of the 8th international conference on Multimodal interfaces.* ACM. 2006, pp. 72–75.

[190] C Semini et al. “Design of HyQ – a hydraulically and electrically actuated quadruped robot”. In: *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* 225.6 (2011), pp. 831 –849.

[191] W.-M. Shen, R. Kovac, and M. Rubenstein. “SINGO: A single-end-operative and genderless connector for self-reconfiguration, self-assembly and self-healing”. In: *ICRA.* 2009, pp. 4253–4258.

[192] W.-M. Shen, Y. Lu, and P. Will. “Hormone-based control for self-reconfigurable robots”. In: *Proceedings of the fourth international conference on Autonomous agents.* ACM. 2000, pp. 1–8.

[193] W.-M. Shen, B. Salemi, and P. Will. “Hormone-inspired adaptive communication and distributed control for CONRO self-reconfigurable robots”. In: *Robotics and Automation, IEEE Transactions on* 18.5 (2002), pp. 700–712.

[194] W.-M. Shen et al. “Multimode locomotion via SuperBot reconfigurable robots”. In: *Autonomous Robots* 20.2 (Apr. 22, 2006), pp. 165–177.

[195] W.-M. Shen et al. “Rolling and climbing by the multifunctional superbot reconfigurable robotic system”. In: *Proceedings of the Space Technology International Forum.* Citeseer. 2008, pp. 839–848.

[196] M. Shimizu, A. Ishiguro, and T. Kawakatsu. “A modular robot that exploits a spontaneous connectivity control mechanism”. In: *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on.* IEEE. 2005, pp. 1899–1904.

[197] B. Siciliano, L. Sciavicco, and L. Villani. *Robotics: modelling, planning and control.* Springer, 2009. 644 pp.

[198] K. Sims. “Evolving 3D morphology and behavior by competition”. In: *Artificial life* 1.4 (1994), pp. 353–372.

[199] G. C. Sirakoulis and A. Adamatzky. *Robots and Lattice Automata.* Springer, 2015.

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

[201] A. Sproewitz et al. “An active connection mechanism for modular self-reconfigurable robotic systems based on physical latching”. In: *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on.* IEEE. 2008, pp. 3508–3513.

[202] A. Sproewitz et al. “Learning to move in modular robots using central pattern generators and online optimization”. In: *The International Journal of Robotics Research* 27.3 (2008), pp. 423–443.

[203] A. Sproewitz et al. “Roombots-mechanical design of self-reconfiguring modular robots for adaptive furniture”. In: *IEEE International Conference on Robotics and Automation, 2009. ICRA ’09.* May 2009, pp. 4259–4264.

[204] A. Spröwitz et al. “Roombots: A hardware perspective on 3D self-reconfiguration and locomotion with a homogeneous modular robot”. In: *Robotics and Autonomous Systems.* Reconfigurable Modular Robotics 62.7 (July 2014), pp. 1016–1033.

[205] A. Sprowitz et al. “Roombots: Reconfigurable Robots for Adaptive Furniture”. In: *IEEE Computational Intelligence Magazine* 5.3 (Aug. 2010), pp. 20–32.

[206] K. O. Stanley and R. Miikkulainen. “Evolving neural networks through augmenting topologies”. In: *Evolutionary computation* 10.2 (2002), pp. 99–127.

[207] K. Stoy. “The deformatron robot: a biologically inspired homogeneous modular robot”. In: *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on.* May 2006, pp. 2527 –2531.

[208] K. Støy. “Reconfigurable Robots”. en. In: *Springer Handbook of Computational Intelligence.* Ed. by J. Kacprzyk and W. Pedrycz. Springer Berlin Heidelberg, 2015, pp. 1407–1421.

[209] K. Støy and H. Kurokawa. “Current topics in classic self-reconfigurable robot research”. In: *Proceedings of the IROS Workshop on Reconfigurable Modular Robotics: Challenges of Mechatronic and Bio-Chemo-Hybrid Systems.* 2011.

[210] K. Støy, W.-M. Shen, and P. Will. “Global locomotion from local interaction in self-reconfigurable robots”. In: *Proc. of the 7th Intl. Conf. on Intelligent Autonomous Systems (IAS-7).* 2002, pp. 309–316.

[211] K. Støy, W.-M. Shen, and P. Will. “How to make a self-reconfigurable robot run”. In: *Proceedings of the first international joint conference on Autonomous agents and multi-agent systems: part 2.* ACM. 2002, pp. 813–820.

[212] K. Stoy, W.-M. Shen, and P. M. Will. “Using role-based control to produce locomotion in chain-type self-reconfigurable robots”. In: *Mechatronics, IEEE/ASME Transactions on* 7.4 (2002), pp. 410–417.

[213] J. W Suh, S. B Homans, and M. Yim. “Telecubes: mechanical design of a module for self-reconfigurable robotics”. In: *IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA ’02.* Vol. 4. IEEE, 2002, 4095–4101 vol.4.

[214] S. Tang et al. “The UBot modules for self-reconfigurable robot”. In: *ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots, 2009. ReMAR 2009.* IEEE, June 22, 2009, pp. 529–535.

[215] Y. Terada and S. Murata. “Automatic Modular Assembly System and its Distributed Control”. In: *IJRR* 27.3 (Mar. 2008), pp. 445–462.

[216] Y. Terada and S. Murata. “Automatic modular assembly system and its distributed control”. In: *The International Journal of Robotics Research* 27.3-4 (2008), pp. 445–462.

[217] M. Tesch et al. “Parameterized and Scripted Gaits for Modular Snake Robots”. In: *Advanced Robotics* 23.9 (2009), pp. 1131–1158.

[218] R. Thakker et al. “ReBiS-Reconfigurable Bipedal Snake robot”. In: *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE. 2014, pp. 309–314.

[219] H. Tokashiki et al. “Development of a transformable mobile robot composed of homogeneous gear-type units”. In: *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*. Vol. 2. IEEE. 2003, pp. 1602–1607.

[220] M. T. Tolley and H. Lipson. “Fluidic manipulation for scalable stochastic 3D assembly of modular robots”. In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE. 2010, pp. 2473–2478.

[221] R. L. Tummala et al. “Reconfigurable adaptable micro-robot”. In: *Systems, Man, and Cybernetics, 1999. IEEE SMC’99 Conference Proceedings. 1999 IEEE International Conference on*. Vol. 6. IEEE. 1999, pp. 687–691.

[222] C. Unsal, H. Kiliccote, and P. K. Khosla. “I (CES)-cubes: a modular self-reconfigurable bipartite robotic system”. In: *Photonics East’99*. International Society for Optics and Photonics. 1999, pp. 258–269.

[223] M. UPenn. *SMORES (Self-assembling MOdular Robot for Extreme Shapeshifting)*. URL: http://modlabupenn.org/smores/.

[224] M. Vespignani et al. “An experimental study on the role of compliant elements on the locomotion of the self-reconfigurable modular robots Roombots”. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Nov. 2013, pp. 4308–4313.

[225] M. Vespignani et al. “Compliant snake robot locomotion on horizontal pipes”. Safety, Security, and Rescue Robotics (SSRR), 2015 IEEE International Symposium on. To appear. 2015.

[226] M. Vespignani et al. “Role of Compliance on the Locomotion of a Reconfigurable Modular Snake Robot”. Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on. To appear. 2015.

[227] M Vona and D. Rus. “A Physical Implementation of the Self-reconfiguring Cristalline Robot”. In: *Proc. of the IEEE Int. Conf. on Robotics and Automation*. 2000, pp. 1726–1733.

[228] L. Wang and F. Iida. “Physical connection and disconnection control based on hot melt adhesives”. In: *Mechatronics, IEEE/ASME Transactions on* 18.4 (2013), pp. 1397–1409.

[229] *Webots, Commercial Mobile Robot Simulation Software*. URL: http://www.cyberbotics.com.

[230] B. Weel et al. “A robotic ecosystem with evolvable minds and bodies”. In: *Evolvable Systems (ICES), 2014 IEEE International Conference on*. IEEE. 2014, pp. 165–172.

[231] H. Wei et al. “Sambot: A self-assembly modular robot for swarm robot”. In: *2010 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 3, 2010, pp. 66–71.

[232] H. Wei et al. “Sambot: A Self-Assembly Modular Robot System”. In: *IEEE/ASME Transactions on Mechatronics* 16.4 (Aug. 2011), pp. 745–757.

[233] P. White et al. “Three Dimensional Stochastic Reconfiguration of Modular Robots.” In: *Robotics: Science and Systems*. Cambridge. 2005, pp. 161–168.

[234] P. J. White and M. Yim. “Scalable modular self-reconfigurable robots using external actuation”. In: *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE. 2007, pp. 2773–2778.

[235] P. J. White et al. “A General Stiffness Model for Programmable Matter and Modular Robotic Structures”. In: *Robotica* 29 (Special Issue 01 2011), pp. 103–121.

[236] P. White, K. Kopanski, and H. Lipson. “Stochastic self-reconfigurable cellular robotics”. In: *Robotics and Automation, 2004. Proceedings. ICRA’04. 2004 IEEE International Conference on*. Vol. 3. IEEE. 2004, pp. 2888–2893.

[237] P. M. Will, A. Castaño, and W.-M. Shen. “Robot modularity for self-reconfiguration”. In: *Photonics East’99*. International Society for Optics and Photonics. 1999, pp. 236–245.

[238] K. C. Wolfe et al. “M 3 Express: A low-cost independently-mobile reconfigurable modular robot”. In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE. 2012, pp. 2704–2710.

[239] R. J. Wood et al. “Microrobot Design Using Fiber Reinforced Composites”. In: *Journal of Mechanical Design* 130.5 (May 2008), pp. 052304–11.

[240] C. Wu et al. “Motion of an underwater self-reconfigurable robot with tree-like configurations”. In: *Journal of Shanghai Jiaotong University (Science)* 18 (2013), pp. 598–605.

[241] Q. Wu et al. “Motion error analysis of modular self-reconfigurable robot M-Cubes based screw theory”. In: *Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on*. Vol. 2. 2011, pp. 859–866.

[242] H. Yamada, S. Takaoka, and S. Hirose. “A snake-like robot for real-world inspection applications (the design and control of a practical active cord mechanism)”. In: *Advanced Robotics* 27.1 (Jan. 1, 2013), pp. 47–60.

[243] M. Yim. “New locomotion gaits”. In: *, 1994 IEEE International Conference on Robotics and Automation, 1994. Proceedings*. May 1994, 2508–2514 vol.3.

[244] M. Yim, D. Duff, and K. Roufas. “PolyBot: a modular reconfigurable robot”. In: *IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA ’00*. Vol. 1. 2000, 514–520 vol.1.

[245] M. Yim, Y. Zhang, and D. Duff. “Modular robots”. In: *IEEE Spectrum* 39.2 (Feb. 2002), pp. 30–34.

[246] M. Yim. “A reconfigurable modular robot with many modes of locomotion”. In: *Proc. of Intl. Conf. on Advanced Mechatronics*. Tokyo, Japan. 1993, pp. 283–288.

[247] M. Yim. “Locomotion with a unit-modular reconfigurable robot”. PhD thesis. Citeseer, 1994.

[248] M. Yim et al. “Modular Reconfigurable Robots in Space Applications”. In: *Autonomous Robots* 14.2 (Mar. 1, 2003), pp. 225–237.

[249] M. Yim et al. “Modular self-reconfigurable robot systems [grand challenges of robotics]”. In: *Robotics & Automation Magazine, IEEE* 14.1 (2007), pp. 43–52.

[250] M. Yim et al. “Modular self-reconfigurable robots”. In: *Encyclopedia of complexity and systems science*. Springer, 2009, pp. 5618–5631.

[251] K. Yoneda et al. “Acquisition of adaptive behavior for virtual modular robot using evolutionary computation”. In: *Advances in Artificial Life. Darwin Meets von Neumann*. Springer, 2011, pp. 181–188.

[252] S.-S. Yoo et al. “Endoscopic capsule robots using reconfigurable modular assembly: A pilot study”. In: *International Journal of Imaging Systems and Technology* 24.4 (2014), pp. 359–365.

[253] Y. Yoon and D. Rus. “Shady3d: A robot that climbs 3d trusses”. In: *Robotics and Automation, 2007 IEEE International Conference on*. IEEE. 2007, pp. 4071–4076.

[254] E. Yoshida et al. “Micro self-reconfigurable robotic system using shape memory alloy”. In: *Distributed autonomous robotic systems 4*. Springer, 2000, pp. 145–154.

[255] E. Yoshida et al. “Miniaturized self-reconfigurable system using shape memory alloy”. In: *Intelligent Robots and Systems, 1999. IROS’99. Proceedings. 1999 IEEE/RSJ International Conference on*. Vol. 3. IEEE. 1999, pp. 1579–1585.

[256] C.-H. Yu, J. Werfel, and R. Nagpal. “Coordinating collective locomotion in an amorphous modular robot”. In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 2777–2784.

[257] C.-H. Yu et al. “Morpho: A self-deformable modular robot inspired by cellular structure”. In: *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE. 2008, pp. 3571–3578.

[258] J. C. Zagal, J. Ruiz-del Solar, and P. Vallejos. “Back to reality: Crossing the reality gap in evolutionary robotics”. In: *IAV 2004 the 5th IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal*. 2004.

[259] H. Zhang et al. “Development of a low-cost flexible modular robot GZ-I”. In: *Advanced Intelligent Mechatronics, 2008. AIM 2008. IEEE/ASME International Conference on*. IEEE. 2008, pp. 223–228.

[260] L. Zhang, J. Zhao, and H. G. Cai. “A substructure based motion planning method for a modular self-reconfigurable robot”. In: *Robot Motion and Control, 2004. RoMoCo’04. Proceedings of the Fourth International Workshop on*. IEEE. 2004, pp. 371–376.

[261] Y. Zhang et al. “Phase automata: a programming model of locomotion gaits for scalable chain-type modular robots”. In: *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on.* Vol. 3. IEEE. 2003, pp. 2442–2447.

[262] W. Zhen, C. Gong, and H. Choset. “Modeling rolling gaits of a snake robot”. In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on.* 2015, pp. 3741–3746.

[263] Y. Zhu et al. “Research on Locomotive Evolution Based on Worm-Shaped Configuration of Self-reconfigurable Robot HitMSR II”. In: *Electrical Power Systems and Computers.* Springer, 2011, pp. 245–252.

[264] V. Zykov et al. “Evolved and Designed Self-Reproducing Modular Robotics”. In: *IEEE Trans. Robotics* 23.2 (Apr. 2007), pp. 308–319.

[265] V. Zykov et al. “Molecubes extended: Diversifying capabilities of Open-Source modular robotics”. In: *IROS, Self-Reconfigurable Robots Workshop* (2008).

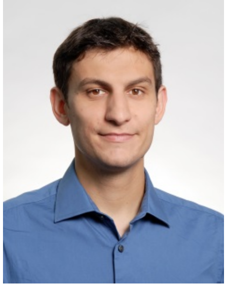
[266] V. Zykov, A. Chan, and H. Lipson. “Molecubes: An Open-Source Modular Robotics Kit”. In: *IROS-2007 Self-Reconfigurable Robotics* (2007).

- ✓ Mechanical designer
- ✓ PhD in Robotics
- ✓ Biomedical engineer

Massimo Vespignani

Avenue du Mont-d’Or, 71
1007 Lausanne, Switzerland

+41 76 612 1359
massimo.vespignani@gmail.com



EDUCATION

01/2011 – 09/2015	**PhD in Manufacturing Systems and Robotics** École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
09/2003 – 05/2009	**Master of Science in Biomedical Engineering** Università Campus Bio-Medico di Roma, Italy
09/1998 – 06/2003	**Industrial Diploma in Electronics and Telecommunications** Istituto Tecnico Industriale "P. Canonica", Bassano Romano, Italy
08/2001 – 06/2002	**High school diploma / exchange year in the United States** Somerset Area Senior High School, Somerset (PA), USA

PROFESSIONAL EXPERIENCE

09/2009 – 12/2010 **Research assistant** (full time job)
Università Campus Bio-Medico di Roma, Italy

- Mechanical and electrical design of a mechatronic platform for studies on intrinsically-motivated learning with children and capuchin monkeys. Responsible for product design from concept to manufacturing.
- Designed a graphical user interface (GUI) for a rehabilitation robot, allowing any inexperienced user to easily control the robot.

TECHNICAL SKILLS

CAD/CAE	SolidWorks, AutoCAD, Inventor, Altium Designer, Eagle, Ansys.
Manufacturing	3D printers: Fused Deposition Modelling (FDM) and Selective Laser Sintering (SLS). Programming of 3-axis CNC milling machine. Surface Mount (SMD) Soldering.
Programming	C, C++, and Python languages. MATLAB. Arduino and PIC microcontrollers. Robot Operating System (ROS) framework. LabView.
Other software	MS Project, Adobe Premiere, Photoshop.
OS	Microsoft Windows, Mac OS X, Ubuntu Linux.

PROJECTS

PhD Thesis: Challenges in the Locomotion of Self-Reconfigurable Modular Robots

- Increased the locomotion performance of a modular robot by 57% by designing task-specific compliant elements.
- Designed and manufactured the world's fastest (in body-lengths per second) running quadruped robot under 30kg (Cheetah-Cub).
- Led 3-person multidisciplinary teams of master students in 4-month projects to develop autonomous mobile robots.
- Managed project resources to ensure deadlines and budgets were met. Contributed to the writing of a successful grant application (Swiss National Science Foundation).

Master Thesis

- Design of a novel control interface with force feedback for safe human-robot interaction. Successfully validated on a rehabilitation robot to inherently increase safety for clinical use.

LIST OF PUBLICATIONS

M. Vespignani, K. Melo, M. Mutlu, and A.J. Ijspeert. "Compliant snake robot locomotion on horizontal pipes." Safety, Security, and Rescue Robotics (SSRR), 2015 IEEE International Symposium on. West Lafayette, IN, 2015.

M. Mutlu, K. Melo, M. Vespignani, A. Bernardino, and A.J. Ijspeert. "Where to Place Camera on a Snake Robot: Focus on Camera Trajectory and Motion Blur." Safety, Security, and Rescue Robotics (SSRR), 2015 IEEE International Symposium on. West Lafayette, IN, 2015.

M. Vespignani, K. Melo, S. Bonardi, and A.J. Ijspeert. "Role of Compliance on the Locomotion of a Reconfigurable Modular Snake Robot." 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Hamburg, Germany, 2015.

M. Mutlu, S. Bonardi, M. Vespignani, S. Hauser, A.J. Ijspeert, and A. Bernardino. "Natural User Interface for Lighting Control: Case Study on Desktop Lighting Using Modular Robots." 0th ACM International conference on Tangible, Embedded and Embodied Interaction (TEI). In review.

S.W. Heim, M. Ajallooeian, M. Vespignani, P. Eckert, and A.J. Ijspeert. "Simplifying Control Through Active Tail Use." Integrative And Comparative Biology, Volume 55, Pages E272-E272, 2015.

A. Özgür, S. Bonardi, M. Vespignani, R. Möckel, and A.J. Ijspeert. "Natural User Interface for Roombots." RO-MAN 2014, The 23rd IEEE International Symposium on Robot and Human Interactive Communication, Edinburgh, United Kingdom, 2014. RO-MAN 2014 Best Paper Award.

S. Bonardi, M. Vespignani, R. Möckel, J. Van den Kieboom and S. Pouya et al. "Automatic Generation of Reduced CPG Control Networks for Locomotion of Arbitrary Modular Robot Structures." Robotics: Science and Systems (RSS), Berkeley, USA, 2014.

A. Spröwitz, R. Möckel, M. Vespignani, S. Bonardi and A.J. Ijspeert. “Roombots: A Hardware Perspective on 3D Self-Reconfiguration and Locomotion with a Homogeneous Modular Robot.” Robotics and Autonomous Systems, Volume 62, Issue 7, July 2014, Pages 1016-1033, 2014.

A. Gams, J. Van den Kieboom, M. Vespignani, L. Guyot, A. Ude, and A.J. Ijspeert. “Rich periodic motor skills on humanoid robots: Riding the pedal racer.” 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 2014.

M. Ajallooeian, A. Tuleu, A. Sprowitz, P. Eckert, M. Vespignani, and A.J. Ijspeert. “Rich Locomotion Skills with the Oncilla Robot.” Dynamic Walking 2014, Zurich, Switzerland, 2014.

M. Vespignani, E. Senft, S. Bonardi, R. Möckel and A.J. Ijspeert. “An experimental study on the role of compliant elements on the locomotion of the self-reconfigurable modular robots Roombots.” 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 2013.

S. Bonardi, M. Vespignani, R. Möckel and A.J. Ijspeert. “Collaborative Manipulation and Transport of Passive Pieces Using the Self-Reconfigurable Modular Robots Roombots.” 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 2013.

R. Möckel, Y. Perov, A. The Nguyen, M. Vespignani and S. Bonardi et al. “Gait Optimization for Roombots Modular Robots - Matching Simulation and Reality.” 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 2013.

F. L. Moro, A. Spröwitz, A. Tuleu, M. Vespignani and N. G. Tsagarakis et al. “Horse-like walking, trotting, and galloping derived from kinematic Motion Primitives (kMPs) and their application to walk/trot transitions in a compliant quadruped robot.” Biological Cybernetics, p. 1-12, 2013.

A. Spröwitz, A. Tuleu, M. Vespignani, M. Ajallooeian and E. Badri et al. “Towards Dynamic Trot Gait Locomotion - Design, Control and Experiments with Cheetah-cub, a Compliant Quadruped Robot.” International Journal of Robotics Research, vol. 32, num. 8, p. 932 - 950, 2013.

A. Spröwitz, A. Tuleu, M. Vespignani, M. Ajallooeian and E. Badri. “Robot Trotting with Segmented Legs in Simulation and Hardware.” Dynamic Walking 2012, Pensacola Beach, Florida. USA, 2012.

S. Bonardi, R. Möckel, A. Spröwitz, M. Vespignani and A.J. Ijspeert. “Locomotion through Reconfiguration based on Motor Primitives for Roombots Self-Reconfigurable Modular Robots.” 7th German Conference on Robotics - Robotik 2012, Munich, Germany, 2012.

L. Zollo, A. Salerno, M. Vespignani, D. Accoto and M. Passalacqua et al. “Dynamic Characterization and Interaction Control of the CBM-Motus Robot for Upper-Limb Rehabilitation.” International Journal Of Advanced Robotic Systems, vol. 10, 2013.

F. Taffoni, M. Vespignani, D. Formica, G. Cavallo and E. P. Di Sorrentino et al. “A mechatronic platform for behavioral analysis on nonhuman primates.” Journal Of Integrative Neuroscience, vol. 11, p. 87-101, 2012.

LANGUAGES

Italian	Native
English	Fluent (C2)
French	Intermediate (B1)
German	Basic (A1)

EXTRA-CURRICULAR ACTIVITIES

Attempt to visit one new country per year.

Outdoor sports: snowboarding, running and inline skating.

PERSONAL INFORMATION

30 years old, Italian citizenship, Swiss Type-B residence permit. European driving license (B).