Université Libre de Bruxelles
Ecole Polytechnique de Bruxelles
IRIDIA - Institut de Recherches Interdisciplinaires
et de Développements en Intelligence Artificielle

Beyond self-assembly: Mergeable nervous systems, spatially targeted communication, and supervised morphogenesis for autonomous robots

Nithin MATHEWS

Promoteur de thèse:
Prof. **Marco DORIGO**

Thèse présentée en vue de l'obtention du titre de
Docteur en Sciences de l'Ingénieur et Technologie
Année académique 2017/2018


SCIENTIA VINCERE TENEBRAS
UNIVERSITAS BRUXELLENSIS


Doctor
Europaeus

Statement

This thesis describes an original research carried out by the author and was only supported by the declared resources. It has not been submitted either in the same or in different form to this or any other university for the award of any degree. The content of this thesis is partially based on articles that the author, together with his co-authors, has made public in the scientific literature. A detailed list of published articles related to this thesis can be found on page 4. Copyrights of some of the presented material (including image and video material) may have been transferred to external journal or conference proceedings publishers. Permission for reprints or reproductions of any kind may need to be requested directly to the copyright holder. In favor of reproducibility of the results presented in this thesis, experimental setups have been described in great details in the respective sections and all robot control code have been made publicly available [1].

February, 2018

Composition of the jury

Michael Allwright, Ph.D.
Postdoctoral Researcher, Université Libre de Bruxelles, Brussels, Belgium

Hugues Bersini, Ph.D.
Professor, Université Libre de Bruxelles, Brussels, Belgium

Mauro Birattari, Ph.D.
Senior Research Associate, Belgian National Funds for Scientific Research

Anders Lyhne Christensen, Ph.D.
Assistant Professor, University Institute of Lisbon, Lisbon, Portugal

Rehan O'Grady, Ph.D.
Cryptosphere Systems
Université Libre de Bruxelles, Brussels, Belgium

Kasper Støy, Ph.D.
Professor, IT University of Copenhagen, Copenhagen, Denmark

Marco Dorigo, Ph.D.
Professor, Université Libre de Bruxelles, Brussels, Belgium

Abstract

The study of self-assembling robots represents a promising strand within the emerging field of modular robots research. Self-assembling robots have the potential to autonomously adapt their bodies to new tasks and changing environments long after their initial deployment by forming new or reorganizing existing physical connections to peer robots. In previous research, many approaches have been presented to enable self-assembling robots to form composite morphologies. Recent technological advances have also increased the number of robots able to form such morphologies by at least two orders of magnitude. However, to date, composite robot morphologies have not been able to solve real-world tasks nor have they been able to adapt to changing conditions entirely without human assistance or prior knowledge.

In this thesis, we identify three reasons why self-assembling robots may not have been able to fully unleash their potential and propose appropriate solutions. First, composite morphologies are not able to show sensorimotor coordination similar to those seen in their monolithic counterparts. We propose "mergeable nervous systems" – a novel methodology that unifies independent robotic units into a single holistic entity at the control level. Our experiments show that mergeable nervous systems can enable self-assembling robots to demonstrate feats that go beyond those seen in any engineered or biological system. Second, no proposal has been tabled to enable a robot in a decentralized multirobot system select its communication partners based on their location. We propose a new form of highly scalable mechanism to enable "spatially targeted communication" in such systems. Third, the question of when and how to trigger a self-assembly process has been ignored by researchers to a large extent. We propose "supervised morphogenesis" – a control methodology that is based on spatially targeted communication and enables cooperation between aerial and ground-based self-assembling robots. We show that allocating self-assembly related decision-making to a robot with an aerial perspective of the environment can allow robots on the ground to operate in entirely unknown environments and to solve tasks that arise during mission time. For each of the three propositions put forward in this thesis, we present results of extensive experiments carried out on real robotic hardware. Our results confirm that we were able to substantially advance the state of the art in self-assembling robots by unleashing their potential for morphological adaptation through enhanced sensorimotor coordination and by improving their overall autonomy through cooperation with aerial robots.

Acknowledgements

I thank Professor Marco Dorigo for giving me the chance to do research at IRIDIA. I also thank Anders Lyhne Christensen and Rehan O'Grady for the guidance and support they have given me to develop the ideas that resulted in this thesis. I am grateful to Alessandro Stranieri, Alexander Scheidler, Arne Brutschy, Eliseo Ferrante, and Gabriele Valentini for their immense contributions. I express my sincere gratitude to Carlo Pinciroli for writing ARGoS, the state-of-the-art simulator used for experimentation throughout this thesis. To all other colleagues I had the chance to meet and spend time with during my years at IRIDIA – what a pleasure!

Most of the research presented in this thesis was carried out within the framework of SWARMANOID, a project funded by the Future and Emerging Technologies programme (IST-FET) of the European Commission, under grant IST-022888. I also acknowledge support from the European Research Council through the ERC Advanced Grant "E-SWARM: Engineering Swarm Intelligence Systems" (contract 246939) and from Wallonia-Brussels-International (WBI) through a Scholarship for Excellence grant.

Contents

CHAPTER 1

Introduction

Self-assembling robots are mobile robots that can autonomously form reversible physical connections with peer robots. By re-arranging these connections, they are able to change their morphologies — i.e., their physical composition — at a group-level. The primary research interest in self-assembling robots is driven by their potential for morphological adaptation allowing an already deployed robotic system to react to changing environments or tasks by forming different morphologies. The long term vision of this research is a future in which robots are no longer built for a particular task. Instead, application systems will manufacture and deploy composable robotic units that build robot morphologies of required shapes and sizes entirely by themselves.

To date, most of the research effort was directed towards the development of control algorithms [2, 3, 4] that enable independent robots to form distinctive morphologies (see Figure 1.1a for examples). By combining the technological advances made in recent years, researchers have also demonstrated the formation of morphologies of previously unattainable scales [5]. In this particular work, the formation of morphologies composed of up to a thousand independent units were shown. However, behavioral coordination in composite robots is either entirely absent or limited to a predefined set of morphologies. As shown in Figure 1.1b, such coordination is a prerequisite to successfully solve real-world tasks. Furthermore, to initiate the formation, researchers often depend on environmental cues [6] or, alternatively, they pre-program [7] the scale and precise shape of morphologies to be formed prior to deployment. These measures render existing self-assembling robot systems semi-autonomous in the best case and unapt to solve real-world tasks.

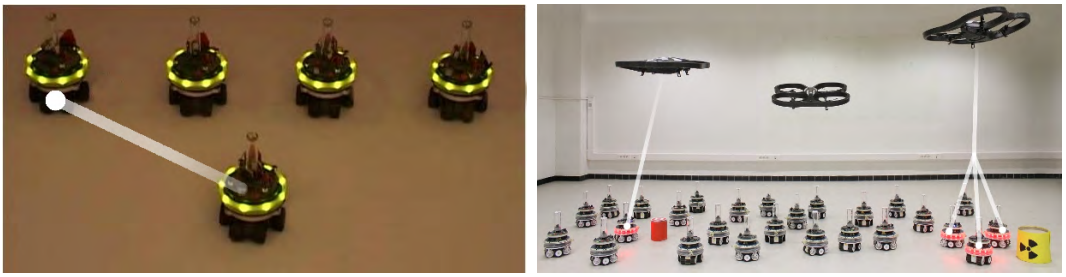
We identify three main reasons why application scenarios based on self-assembling robots were not considered neither by the scientific community nor by commercial applications. First, in existing systems, group-level sensorimotor coordination is limited or absent. Robots remain individually autonomous even when physically connected to each other. The behavioral control paradigm adopted for these systems is often distributed or based on self-organization showing a resemblance to certain natural systems (such as unicellular slime molds [8, 9], ants [10, 11] or honey bees [12]) that also form collective structures. Similar to their biological counterparts, current self-assembling robot systems thus lack a nervous system with a single decision-making unit enabling sophisticated sensorimotor coordination as seen in higher order animals. Second, spatial positioning of communicating robots often plays a crucial role in multirobot systems that require spatial coordination. In most communication modalities used today, such as in wireless Ethernet, spatial infor-

(a) (b)

Figure 1.1: Composite robots or morphologies formed by self-assembling robots. (a) A series of example morphologies that can be formed by self-assembling robots. (b) An illustration of an object-pushing task that requires behavioral coordination among connected robots.

mation is largely unavailable to robots and needs to be acquired through additional position determining technologies. To date, systems that allow robots to select communication partners based on location either rely on centralized solutions or consider robots that have a global view of the system – approaches not available to most decentralized systems including self-assembling robot systems. Third, existing self-assembling robot systems consider homogeneous robots groups that operate on the same planar surface. Robots in such groups are equipped with an identical set of actuators and sensors making their perception of the environment rather similar to each other. No single robot in such a group may be able to gather significantly different information about the task or the environment than that available to its peer robots. Therefore, decisions regarding when to trigger self-assembly and which morphology may be suited to solve the task at hand often cannot be made by a single robot alone, unless pre-programmed or assisted otherwise.

In this thesis, we exploit the technological progress that fuels today's self-assembling robots in terms of computation power, memory, sensing and communication capabilities and propose "mergeable nervous systems for robots" as a solution to the sensorimotor coordination problem. This proposal transforms a composite multi-robot system into a new unified robot with a single decision-making unit. We define the robot nervous system as the signaling architecture of a robot together with the processing unit (the "brain unit") that takes high-level decisions for the whole robot. To form an holistic robot that is composed of initially independent units, one of the robotic units needs to assume authority (and become the brain unit) while all others cede authority. For this purpose, we apply a hierarchical tree structure that follows the physical connections made within a morphology. We also divorce the control logic in a merged nervous system from the morphology and its available sensors and actuators such that a single control logic can take every physically possible morphology into account. Constituent robotic units process sensor data in a distributed fashion and transmit high-level representations of the perceived environment to the brain unit. In turn, the brain unit issues high-level commands about actions the body should take, and the merged nervous system takes responsibility for translating those commands into instructions for individual actuators in each constituent unit. We show that self-assembling robots that implement mergeable nervous systems become a new class of robots with capabilities beyond those of any existing artificial

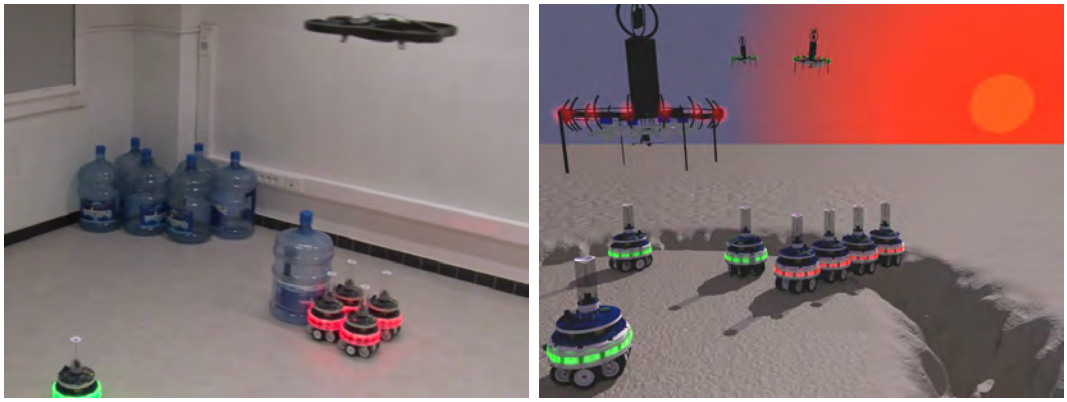


(a) (b)

Figure 1.2: An illustration of spatially targeted communication in two different multirobot systems. The overlays show established communication links. (a) An homogeneous robot system composed of five ground-based robots in which one robot on the bottom communicates with the robot on the top-left. (b) An heterogeneous robot system composed of aerial and ground-based robots in which the aerial robots use spatially targeted communication to issue commands to specific robots on the ground located next to objects of interests. On the left, a one-to-one link and on the right, a one-to-three link is shown.

or biological system. Such robots can merge into larger bodies with a single brain unit, split into separate bodies with independent brains, and self-heal by removing or replacing malfunctioning body parts, even a malfunctioning brain.

To solve the communication problem, we propose "spatially targeted communication". Spatially targeted communication relies on situated communication [13] – a communication modality in which messages also reveal the relative position of the message sender. In a sense, spatially targeted communication provides decentralized multirobot systems with the inverse functionality offered by situated communication. That is, messages do not have to be broadcast and processed by all recipient robots to determine the addressees. Instead, a message sending robot can select the recipient robots based on their spatial position in the environment and establish dedicated communication links without requiring any prior information or centralized solutions. We show how dedicated communication links can be established to interestingly located robots using light-emitting diodes (LEDs) that then can be detected using cameras mounted on the robots. The proposed method does not require centralized control, environment maps, unique IDs, or external hardware installation to localize the robots. It can also be applied in outdoor or indoor environments by robots that are not able to share a common frame of reference (such as GPS) or by robot that are deployed on missions that require higher precision than that offered by civilian GPS systems. We present an iterative elimination process that allows robots to narrow down the number of broadcast message recipients until a dedicated communication link is established to a single peer robot in the communication range (see Figure 1.2a). An iterative growth process can be then executed to expand the link to a coherent group of co-located robots. See Figure 1.2b for an example one-to-one and one-to-three spatially targeted communication link. We also show that these link establishment processes are highly scalable with respect to the total number of robots in the communication range that participate in the processes. Spatially targeted communication can be applied to both ground-based homogeneous robot groups as well as to heterogeneous robot teams operating in three-dimensional environments.



(a) (b)

Figure 1.3: Supervised morphogenesis shown in two different scenarios. (a) An aerial robot has recruited and guided four self-assembling robots to form a morphology adequate to the object-pushing task. (b) Another aerial robot supervises the formation of a sufficiently long chain-like morphology that helps the robots cross a ditch too wide for a single robot to cross.

Finally, we integrate the work described so far and present "supervised morphogenesis" – a control methodology that allows self-assembling robots to rely on aerial robots to act as an "eye-in-the-sky" with decision-making authority. In such heterogeneous groups, aerial robots exploit their elevated position and agility to survey the ground on which self-assembling robots operate to determine when self-assembly is required and what morphologies need to be formed as a function of the perceived task or environment. For this purpose, the aerial robots first build a model of the environment and then execute on-board simulations based on the model. Depending on the outcome of the simulations, aerial robots initiate and control (hence supervise) the formation of morphologies on the ground. On-board simulations allow aerial robots to assess the performance of different morphologies in a particular environment prior to their costly and physical realization on the ground. A robot on the ground is then selected by the aerial robot using spatially targeted communication that then can *seed* (or initiate) the growth of a task-dependent morphology – see Figure 1.3 for two example scenarios. In simulation-based experiments, we also quantify the benefits of cooperation between an aerial robot and self-assembling robots in terms of task completion times. Our experiments show that supervised morphogenesis allows self-assembling robots to adapt to previously unknown tasks and environments by cooperating with an aerial robot and thus significantly increases their level of autonomy.

1.1 Thesis structure and original contributions

Here we present a structural overview of the thesis by describing the contents of its chapters and by listing the scientific publications that resulted from each of them.

In Chapter 2, we present the robot platforms used for experimentation in this thesis. We present details of two different self-assembling robot platforms, namely the *s-bot* and its successor the *marXbot*. We also describe the aerial platforms *eye-bot* and the commercially available *AR.Drone* we used in our experiments.

In Chapter 3, we present a novel methodology to control composite robot morphologies. We refer to it as "mergeable nervous systems for robots". In Section 3.1, we first detail the basic connection forming mechanism we developed to enable autonomous self-assembly between two self-assembling robots. The underlying algorithms and experimental results were published in the following international conference:

- **N. Mathews**, A. L. Christensen, R. O'Grady, P. Rétornaz, M. Bonani, F. Mondada, and M. Dorigo. Enhanced directional self-assembly based on active recruitment and guidance. In *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011)*, pages 4762–4769. San Francisco, USA. September 2011.

  - Symposium on (self-)assembly from the nano to the macro scale: state of the art and future directions

Second, in Section 3.2, we detail how a mechanism developed in previous research can be combined with communication schemes developed in this thesis to control the growth of pre-defined topology-aware, robot morphologies. Third, in Section 3.3, we present the key contribution of this chapter by detailing the control and communication logic we developed for physically connected multirobot morphologies. This logic enables composite morphologies demonstrate previously unseen features and was published in the following journal article:

- **N. Mathews**, A. L. Christensen, R. O'Grady, F. Mondada, and M. Dorigo. Mergeable nervous systems for robots. *Nature Communications*, 8(439), 2017.

  - Journal impact factor at the time of publication: 12.124 (as of 2016)
  - Reached an Altmetric score of 340 within 5 days after publication (top 5% of all research outputs scored by Altmetric)

In Section 3.4, we review existing modular robot systems with a focus on their coordination mechanism before we summarize the contribution of the chapter in Section 3.5 and discuss potential limitations.

In Chapter 4, we describe how "spatially targeted communication" links can be established in decentralized multirobot systems. Preliminary results of this approach were published in an international conference while the more comprehensive work was published in a journal:

- **N. Mathews**, A. L. Christensen, E. Ferrante, R. O'Grady, and M. Dorigo. Establishing spatially targeted communication in a heterogeneous robot swarm. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems* (AAMAS 2010), pages 939–946. Toronto, Canada. May 2010.

  - Nominated for the CoTeSys Best Robotics Paper Award

- **N. Mathews**, G. Valentini, A. L. Christensen, R. O'Grady, A. Brutschy, and M. Dorigo. Spatially targeted communication in decentralized multirobot systems. *Autonomous Robots*. 38(4): 439-457, 2015.

In Section 4.3, we discuss potential shortcomings of the presented approach together with solution proposals. In Section 4.4, we review communication modalities applied in existing multirobot systems that enable spatial coordination. We summarize the chapter in Section 4.5.

In Chapter 5, we integrate the work presented in Chapters 3 and 4 and present "supervised morphogenesis" – an approach that enables aerial robots increase the autonomy of self-assembling robots on the ground. Preliminary ideas and results were published in the following video proceeding:

- **N. Mathews**, A. L. Christensen, R. O'Grady, and Marco Dorigo. Spatially targeted communication and self-assembly. In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2012)*, pages 2678–2679. Vilamoura, Algarve, Portugal. October 2012.
  - Best IROS 2012 Video Award Finalist
  - More than 685 000 views on YouTube (as of February, 2018)

The main methodology is presented in Section 5.1 followed by two case studies in Section 5.2 and Section 5.3, respectively. In simulation, we also quantify the performance benefits attainable to self-assembling robots through cooperation with aerial robots and present the results in Section 5.4. These efforts were published in the following two international conferences:

- **N. Mathews**, A. Stranieri, A. Scheidler, and Marco Dorigo. Supervised morphogenesis - morphology control of ground-based self-assembling robots by aerial robots. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*,pages 97–104. Valencia, Spain. June 2012.

- **N. Mathews**, A. L. Christensen, R. O'Grady, and M. Dorigo. Cooperation in a heterogeneous robot swarm through spatially targeted communication. In *Proceedings of the 7th International Conference on Ant Colony Optimization and Swarm Intelligence (ANTS 2010)*, volume 6234 of Lecture Notes in Computer Science, pages 400–407. Brussels, Belgium. September 2010.

A comprehensive work composed of the research presented in this chapter has also been submitted to a journal for review:

- **N. Mathews**, A. L. Christensen, A. Stranieri, A. Scheidler, and M. Dorigo. Supervised morphogenesis: exploiting morphological flexibility of self-assembling multirobot systems through cooperation with aerial robots. *Under review.*

We present a literature review of existing air-ground robot teams and detail the kind of tasks they solve in Section 5.5 before presenting a summary of the chapter in Section 5.6.

In Chapter 6, we conclude the findings presented in this thesis and discuss possible future work.

1.2 Other contributions

The doctoral studies presented in this thesis have also made contributions to a number of related research topics not reported here:

- G. Podevijn , R. O'Grady, **N. Mathews**,, A. Gilles, C. Fantini-Hauwel, M. Dorigo. Investigating the effect of increasing robot group sizes on the human psychophysiological state in the context of human-swarm interaction. Swarm Intelligence. 10(3): 193-210, 2016.

- M. Dorigo, D. Floreano, L.M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy, D. Burnier, A. Campo, A. L. Christensen, A. Decugnière, G. Di Caro, F. Ducatelle, E. Ferrante, A. Forster, J. Martinez Gonzales, J. Guzzi, V. Longchamp, S. Magnenat, **N. Mathews**, M. Montes de Oca, R. O'Grady, C. Pinciroli, G. Pini, P. Rétornaz, J. Roberts, V. Sperati, T. Stirling, A. Stranieri, T. Stützle, V.Trianni, E. Tuci, A. E. Turgut, and F. Vaussard. Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics & Automation Magazine.* 20(4):60-71, 2013

  - More than 240 citations according to Google Scholar (as of February, 2018)

- C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, **N. Mathews**, E. Ferrante, G. Di Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, M. Dorigo. ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4):271-295, 2012.

  - More than 160 citations according to Google Scholar (as of February, 2018)

- "Swarmanoid." Daily Planet. Prod. Cindy Bahadur. Discovery Channel Canada, Toronto, Canada. 27th of March, 2012.

  - Media coverage of the Swarmanoid project – includes visuals from the research carried out on "Supervised Morphogenesis"

- M.Dorigo, M. Birattari, R. O'Grady, L. M. Gambardella, F. Mondada, D. Floreano, S. Nolfi, T. Baaboura, M. Bonani, M. Brambilla, A. Brutschy, D. Burnier, A. Campo, A. L. Christensen, A. Decugnière, G. Di Caro, F. Ducatelle, E. Ferrante, J. Martinez Gonzales, J. Guzzi, V. Longchamp, S. Magnenat, **N. Mathews**, M. Montes de Oca, C. Pinciroli, G. Pini, F. Rey, P. Rétornaz, J. Roberts, F. Rochat, V. Sperati, T. Stirling, A. Stranieri, T. Stützle, V. Trianni, E. Tuci, A. E. Turgut, and F. Vaussard. Swarmanoid, The Movie.

  - Best Video Award @ AAAI-11 Video Competition (AIVC 2011). San Francisco, September 2011
  - Botsker Award for the Most Innovative Technology @ 2nd Annual Robot Film Festival. New York City, July 2012
  - Prix Wernaers 2012, Brussels, August 2012
  - More than 435 000 views on YouTube (as of February, 2018)

- C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, **N. Mathews**, E. Ferrante, G. Di Caro, F. Ducatelle, T. Stirling, A. Gutiérrez, L. M. Gambardella, and M. Dorigo. ARGoS: A modular, multi-engine simulator for heterogeneous swarm robotics. In *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011)*, pages 5027–5034. San Francisco, USA. September 2011.

- – More than 100 citations according to Google Scholar (as of February, 2018)

- E. Ferrante, A. E. Turgut, **N. Mathews**, M. Birattari, and M. Dorigo. Flocking in stationary and non-stationary environments: a novel communication strategy for heading alignment. In *Proceedings of 11th International Conference on Parallel Problem Solving from Nature (PPSN XI)*, number 6239 in Lecture Notes in Computer Science, pages 331–340. Krakow, Poland. September 2010.

- M. A. Montes de Oca, E. Ferrante, **N. Mathews**, M. Birattari, and M. Dorigo. Opinion dynamics for decentralized decision-making in a robot swarm. In *Proceedings of the 7th International Conference on Ant Colony Optimization and Swarm Intelligence (ANTS 2010)*, volume 6234 of Lecture Notes in Computer Science, pages 252–263. Brussels, Belgium. September 2010.

  – Nominated for the Best Paper Award

- M. A. Montes de Oca, E. Ferrante, **N. Mathews**, M. Birattari, and M. Dorigo. Optimal collective decision-making through social influence and different action execution times. In *Proceedings of the Workshop on Organisation, Cooperation and Emergence in Social Learning Agents of the European Conference on Artificial Life* (ECAL 2009). Budapest, Hungary. September 2009.

1.3 Publication summary

According to Google Scholar, my h-index (as of February, 2018) is 10. All original scientific contributions that resulted from this doctoral thesis are summarized in the following. First author contributions are indicated with (⧫).

6 Journal papers

Robotics and Autonomous Systems (under review), 2017 (⧫)

Nature Communications, 2017 (⧫)

Swarm Intelligence, 2016

Autonomous Robots, 2015 (⧫)

IEEE Robotics & Automation Magazine, 2013

Swarm Intelligence, 2012

8 Peer-reviewed conference papers

International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2012 (⧫)

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012 (⧫)

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2011 (⧫)

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2011

International Conference on Parallel Problem Solving from Nature (PPSN), 2010

International Conference on Ant Colony Optimization and Swarm Intelligence (ANTS), 2010 (⧫)

International Conference on Ant Colony Optimization and Swarm Intelligence (ANTS), 2010

International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2010 (⧫)

2 Video proceedings

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012 (⧫)

AAAI Conference on Artificial Intelligence, 2011

3 Visual media coverage

Discovery Channel, 2012

MSN News, 2017

The Wall Street Journal, 2017

1 Workshop proceeding

European Conference on Artificial Life (ECAL), 2009

CHAPTER 2

Robot platforms

The experiments described in this thesis have been conducted using multiple robot platforms. This is because, during the course of this thesis, technological progress experienced an unprecedented upsurge making advanced robotic hardware cheaper and rather easily available. Hence, we regularly adapted our studies to new hardware to allow our algorithms and approaches take full advantage of this progress. We present four robot platforms used for experimentation in this thesis: two ground-based robot platforms named the *s-bot* and the *marXbot* and two aerial platforms named the *eye-bot* and the *Parrot AR.Drone* (hereafter referred to as AR.Drone). Control programs for the s-bots were developed using the TwoDee simulator [14]. For the development of controllers of the marXbot and the eye-bot platforms, we used a novel, physics-based simulation framework [15, 16]. This framework, developed specifically for the study of large-scale heterogeneous robot systems, was also used to conduct scalability experiments that required more than twelve robots. For the AR.Drone, we used an adaptation [17] of the SDK provided by the manufacturer to control the robot using a joystick or using a robot control program we developed for autonomous control.

We conducted our initial experiments using ground-based, autonomous robots called s-bots [18] (see Figure 2.1A). S-bots were developed as part of the *Swarm-bots* project [19] and became first available in 2005. In this project, a particular importance was given to the design and development of robots that can physically interact with each other and self-assemble. The resulting s-bot is a mobile robot with a circular chassis of 11.6 cm diameter powered by an XScale CPU running at 400 MHz. The robot features a wide range of sensors and actuators: 3-axes accelerometer, light sensors, 15 IR proximity sensors, ground sensors, microphones and loudspeakers, WiFi communication, a temperature sensor, a traction sensor to measure horizontal forces applied to the robot chassis, and a system for locomotion composed of both tracks and wheels providing differential drive capabilities. A distinct feature of this robot is its 8 RGB colored controllable LEDs allowing the s-bot to change the ambient color in its immediate vicinity. The LEDs are evenly placed inside a transparent ring part of the circular chassis. The s-bot's most innovative and visible feature is its rigid gripper that enables self-assembly between multiple s-bots. Each s-bot is also equipped with an omni-directional camera (0.3 MP) pointing upward at a hemi-spherical mirror. The mirror is mounted at the top end of a transparent tube reflecting panoramic images of the s-bot's vicinity up to a distance of 70 cm, depending on light conditions. A rechargeable, fixed mount battery provides the s-bot with an autonomy of around 2 h after which it needs to be shutdown

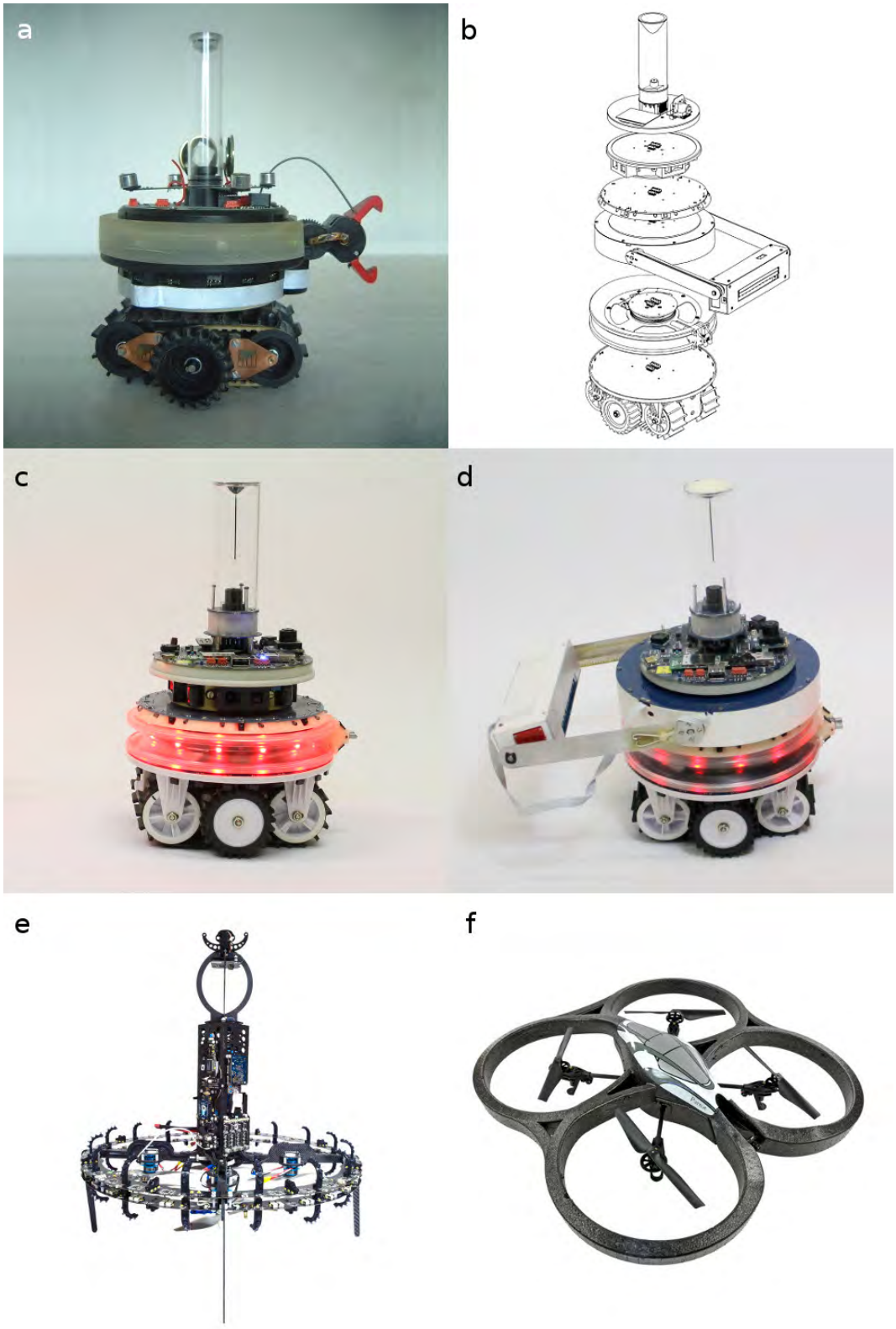


Figure 2.1: The two ground-based and the two aerial robot platforms used for experimentation in this thesis. (A) An s-bot, (B) an exploded view of the modular marXbot platform showing the modules considered in this thesis. (C) A foot-bot and (D) a manipulator foot-bot with its magnetic gripper module mounted. (E) An eye-bot and (F) an AR.Drone with its indoor hull (source: `www.ifixit.com/`).

for charging. The s-bots have been used in several experiments in studies focusing on topics such as self-assembly [20], group transport [21], morphology control [22], and negotiation of goal direction [23]. In this thesis, the s-bots are used in the experiments described in Chapter 4.

The marXbot [24] is the revised version of the s-bot platform and was first introduced in 2010 (see Figure 2.1B). It was developed as part of the *Swarmanoid* project [25] – the follow-up project of the Swarm-bots project. While there are certain physical similarities between the marXbot and its predecessor, the marXbot is a fundamentally different robotic platform designed with the idea of component modularity as the key driver. As shown in Figure 2.1B, sensors and actuators are built into modules that can be swapped in and out of robot configurations. This feature essentially allows researchers to develop tools (such as simulation frameworks, toolchains, etc.) targeting a single platform while being able to use those tools for an array of different module configurations. Each such configuration is a robot on its own right featuring distinct physical properties. At the time of writing, new modules (with a maximum diameter of 17 cm) are still being produced for the marXbot platform. For what concerns the experiments described in this thesis, however, we considered two different module configurations resulting in two physically different robots: a foot-bot (see Figure 2.1C) and a manipulator foot-bot (see Figure 2.1D).

With the exception of the microphones and loudspeakers, the foot-bot matches all capabilities available to the s-bot. In the following, we list improvements of existing features or complete novelties solely available to the foot-bot: an ARM 11 processor (i.MX31 clocked at 533 MHz and with 128 MB RAM) running a Linux-based operating system, 12 RGB-colored LEDs, a 2D distance scanner, 24 IR proximity sensors, a 3-axis gyroscope, one omni-directional (3 MP) and one ceiling (3 MP) camera. The foot-bot also has a hot-swappable battery system that provides a much longer autonomy than that of the s-bot. That is, a battery running out of charge can be replaced with a pre-charged one without requiring a complete shutdown of the robot. A key novelty of the foot-bot is its communication device [26] that also allows the simultaneous estimation relative positions (i.e., the range and bearing) of peer robots. A detailed description of the functioning of this device is given in Section 3.1.1. The self-assembly module enables multiple foot-bots to physically connect to one another, that is, to *self-assemble*. This module includes a rotatable docking module composed of an active docking unit with three fingers and a passive docking ring. A physical connection is formed when a foot-bot inserts its docking unit into the docking ring of another foot-bot and then opens its three fingers. The foot-bot also supports rapid development and debugging of robot behaviors enabled by an event-based control architecture [27] implemented at the micro-controller level. The foot-bot module configuration has been successfully used for other studies including robot flocking [28], self-organized decision making [29] and task-partitioning [30] in robot swarms. In this thesis, the foot-bot has been used for experimentation described in Chapters 3, 4, and 5. The manipulator foot-bot has a similar configuration as that of a foot-bot with a magnetic gripper module [31] mounted instead of the distance scanner module. The magnetic gripper module has 3-DoF and allows the robot to manipulate its environment by retrieving, transporting and depositing ferromagnetic objects. The manipulator foot-bot has been used in an experiment presented in Chapter 3.

Figure 2.1E shows an eye-bot [32], the first aerial platform we used for experimentation. The thrust and control of the eye-bot is provided by its 8 rotors mounted

in a co-axial quadrotor configuration. The eye-bot chassis is made out of carbon-fiber and weighs 270 g. At a height of 54 cm and with a chassis diameter of 50 cm, the on-board battery provides the eye-bot with 10-20 minutes of autonomy. A novel ceiling attachment system composed of active magnets allow the eye-bot to extend this autonomy considerably in indoor environments by attaching to metal ceilings or bars and by remaining static. Other features include a downward-pointing 2 MP HD 360° pan-and-tilt camera that allows the eye-bot to survey the ground for other robots and objects, a ring encompassing the robot chassis with 16 built-in RGB LEDs, an altitude sensor, and a magnetometer to detect heading direction. The eye-bot is also the first robot with a 3D relative positioning and communication device [33]. This on-board device has a maximum range of 12 m and allows an eye-bot to communicate with other eye-bots in flight as well as with marXbot configurations on the ground equipped with the range and bearing module, such as the foot-bot. The eye-bot is one of the three robots developed as part of the Swarmanoid project. We used the eye-bots in experiments described in Chapters 4 and 5.

The AR.Drone [34] is an electrically powered quadcopter (see Figure 2.1F) produced by a French manufacturer named Parrot SA. Introduced in January 2010, it was the first commercially available aerial platform to gain widespread popular attention in the European consumer electronics market. This is mainly because of the fact that it can be piloted over WiFi with a smartphone or a tablet and also its affordable pricing. We, however, considered the AR.Drone as a research platform because of three reasons: i) its open API allowing third party developers to control the AR.Drone via software, ii) its well documented SDK, and iii) its built-in aerial stabilization behaviors and assisted maneuvers that allow the platform to be flown by non-experts without a background in avionics or aeronautical engineering. The AR.Drone has an autonomy of up to 12 minutes at a maximum flight weight of 420 g while being able to fly at speeds up to 18 km/h. The platform's main computational unit is an ARM 9 processor running at 468 MHz with 128 MB of DDR RAM. The AR.Drone is composed of a central plastic body housing a sensor and control board, one front-facing and another downward-pointing camera, a carbon-fiber crossbar connecting four brushless motors, and one removable indoor and another outdoor hull. Further features include a 6-DoF inertial measurement unit and an ultrasound altimeter. The manufacturer provides an API to access sensory information such as the altitude, battery level, and images from the cameras, and allows communication with the AR.Drone at 30 Hz via an ad-hoc wireless Ethernet network. Rather than using the SDK provided by the manufacturer directly, we used an adaptation of the SDK presented in [17] due to better usability. This adaptation software relies on SDK version 1.6 and firmware version 1.3.3 installed on the robot. We used this SDK adaptation to channel video streams from the AR.Drone to a remote PC where vision algorithms were run. Position control data computed on the basis of these streams were then transmitted from the PC back to the AR.Drone in real time using the same channel. The AR.Drone has been used in experiments presented in Chapters 4 and 5. Note that Parrot introduced the successor of the AR.Drone (named AR.Drone 2.0) with updated hardware specifications and improved API in 2012.

CHAPTER 3

Mergeable nervous systems for robots

The nervous system of a robot can be defined as the signaling architecture composed of the robot's main processing unit, the various subsystems including the sensors and actuators, and the wires and cables (electrical or otherwise) that connect them both to each other. In today's autonomous robots, this signaling architecture is defined at design time and is mapped to their morphologies even more strictly than in biological systems [35, 36]. While some robots can reconfigure their morphologies [37], share sensor and actuator capabilities [38, 39] or autonomously self-assemble [5, 40, 41] to form entirely new morphologies, the underlying signaling architecture of the robotic units put forward at design time is not subject to change. That is, even when functioning at a group-level, the units remain individually autonomous depriving the robot of a group-level sensorimotor coordination. To date, no two or more robots have shown the ability to merge their nervous system providing the composite robot structure with a sensorimotor coordination similar to that observed in most monolithic robots.

In this chapter, we use both foot-bots and manipulator foot-bots to show how robots – independent of their hardware makeup – can merge their nervous systems together with their physical bodies. We refer to these robots as mergeable nervous system (MNS) robots and show how they represent a new class of robots with capabilities beyond those of any existing machine or biological organism. MNS robots can merge into larger bodies with a single brain, split into separate bodies with independent brains, and self-heal by removing or replacing malfunctioning body parts, even a malfunctioning brain. Note that in Table 3.1, we define the terminology used in this chapter.

This chapter is organized as follows. In Section 3.1, we present the basic mechanism we use to enable connections between two foot-bots (without merging their nervous systems). In Section 3.2, we detail how we apply these connections to control the formation of larger morphologies of which the shape and size can be precisely defined. We also present the concept of a brain unit – a key feature in a mergeable nervous system. We then provide details on how to program MNS robots and address scalability properties in Section 3.3. We continue to demonstrate four features unique to MNS robots. In Section 3.4, we review the state of the art in existing modular robot systems with a focus on their coordination mechanisms. Finally, we summarize the contribution of this chapter in Section 3.5 and present our conclusions.

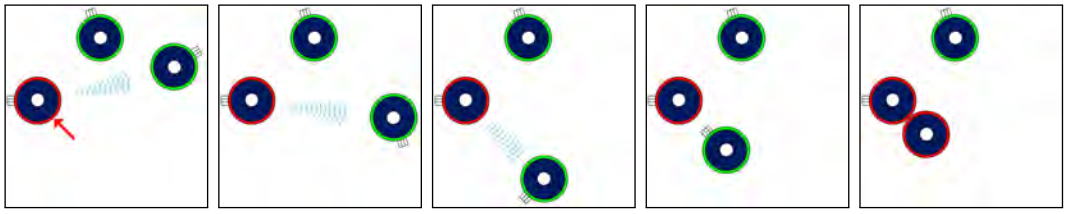
Table 3.1: Glossary of terminology used in this chapter.

Term	Definition
morphology	collective robot structure composed of two or more independent robotic units physically connected to each other
target morphology	(task-dependent) morphology of a precise shape and size
seed robot	robot triggering the self-assembly process
free robot	robot not pursuing any task and available for connection forming
extension point	an angle around the robot's chassis at which a connection is required
extending robot	robot (independent or part of a morphology) with an extension point
filled extension point	an extension point that has already received a connection from a free robot
brain unit	decision-making robot in a morphology
robotic unit	robots in a morphology that are not the brain unit

3.1 Enhanced directional self-assembly (EDSA)

In previous research carried out using s-bots [42], the term "directional self-assembly" was first introduced to refer to a basic connection forming mechanism between two s-bots. In this work, s-bots were shown to form direction specific connections using cameras and LEDs. Directional self-assembly enabled a connection extending s-bot to have multiple extension points around its chassis to which another s-bot could connect to. Since the mechanism was based on colors displayed on LEDs, it was impossible for the extending s-bot to either select which neighboring s-bot reacted to the lights first and finally connected to which extension point. The mechanism was susceptible to density related inefficiencies as it relied on stochastic movement of the s-bots. Also, due to perturbation caused in the displayed directionality, it was impossible for an s-bot to have two open extension points open at the same time. Hence, multiple extension points on a single s-bot had to be filled in a strictly sequential manner prolonging the overall time required to form multiple connections.

Here we propose "enhanced directional self-assembly" (or EDSA for short) as the basic mechanism that enables the formation of connections between two self-assembling foot-bots. EDSA is based on the mxRAB device (presented in Section 3.1.1) available to the foot-bots and avoids the inefficiencies caused by stochastic robot movement and allows for parallel morphology growth from a single extending foot-bot. A foot-bot seeking for an extension of its morphology can invite a physical connection by actively recruiting the best located peer robot and by guiding it to the location where the connection is required. Robots available for self-assembly broadcast their availability to be recruited. The extending robot recruits the best situated robot for the connection that it wishes to receive. The extending robot subsequently



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

Figure 3.1: In EDSA, a connection is formed between (a) an extending robot (shown on center-left with the arrow indicating an expected extension point) and the best situated free robot with respect to an extension point. The extending robot recruits the best situated free robot and (b) guides this recruit by transmitting a series of messages that allow (c) the recruit to maneuver and align itself with the extension point. (d) Once aligned, the recruit rotates to face the extending robot and approaches the extension point. (e) The extending robot and the recruit form a physical connection (hence merge their bodies) at the expected extension point.

guides the recruit to the point on its body at which the recruit should connect. Once the recruit has successfully formed the connection, it sends an acknowledgement to the extending robot. Figure 3.1 illustrates how a connection is formed between a connection inviting foot-bot and the best situated free foot-bot with respect to the extension point. The underlying recruitment and guidance mechanisms presented in Section 3.1.2. We also present results of benchmark experiments we conducted using the same setup and measures defined in [42]. The results of these experiments presented in 3.1.3 show that EDSA goes beyond the state-of-the-art by achieving higher speeds and precision to form connections. We go on to show that EDSA also enables previously unattainable types of morphology growth.

3.1.1 The mxRAB device and messages communicated

The mxRAB (**m**ar**X**bot **R**ange **A**nd **B**earing) device is the main communication device of the foot-bots. It is used for communication between foot-bots but can also be used for communication with the eye-bots and with the climbing robots presented in [43]. In addition to high-speed communication (1 Mbps), the mxRAB device provides communicating robots with range and bearing estimates (with an additional elevation estimate for robots not in the same plane) to other robots in the line-of-sight at a refresh rate of 10 Hz. That is, the mxRAB device enables *situated communication* – a form of communication in which a received message also reveals the relative location of its sender in the environment [13].

The technology underlying the mxRAB device is a combination of 16 infra-red transmitters, 8 infra-red receivers, and one radio transceiver. This transceiver operates at 2.4 GHz and is used to send and receive data can function any light condition. Additionally, it is used to synchronize a turn-taking algorithm among all the mxRAB devices (and therefore all foot-bots). The turn taking algorithm (with configurable time division multiplexing) underlying the mxRAB device is implemented by predefining a sequence of unique IDs assigned to each mxRAB device in the system. Due to the 10 Hz refresh rate and the minimum time (ca. 4 ms) needed for a infra-red signal to stabilize, read and then dissipated, the current version of the mxRAB device allows the simultaneous deployment of up to 25 different devices. This means

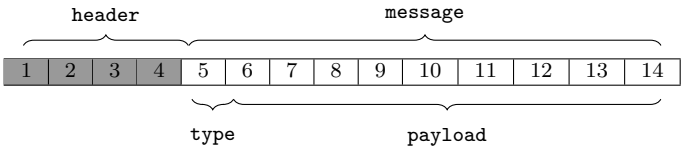


Figure 3.2: A schematic of the 14 byte data packet exchanged between the mxRAB devices mounted on the foot-bots. The first 4 bytes (shown in shades) are reserved for the operation of the device itself and are unavailable to robot control programs. For the remaining 10 bytes, we define a message type (5th byte) and a message payload (bytes 6 to 14).

that each mxRAB device can estimate relative positions and receive data from up to 24 other robots in every control step – that is every 100 ms. When given its turn, a foot-bot simultaneously sends data over the radio and emits a signal using its 16 infra-red transmitters distributed around its chassis. When it is not its turn, a foot-bot uses its radio transceiver to receive data and its 8 infra-red receivers to detect the infra-red signal sent by a data sending foot-bot. A data receiving foot-bot feeds the strongest signal strength detected into a series of cascaded amplifiers to determine the range to the data sending foot-bot. The receiver with the strongest signal strength and both its adjacent receivers are used to triangulate the bearing to the data sending foot-bot. The maximum error amounts to 6.5% for range and 2.6% for bearing for the distance interval of 0 cm to 500 cm. These results were presented in [26].

In every control step, the mxRAB device sends a data packet of 14 bytes (see Figure 3.2) and it receives up to 24 packets from other robots. From the 14 bytes of data sent, the first four bytes are the *header* and are used by the mxRAB device. The first byte is the unique ID of the mxRAB device and the other three bytes are used to ensure compatibility with the eye-bot. The remaining 10 bytes are what we refer to as a *message* of which the first byte is the *message type* and the remaining 9 bytes are the *message payload*. While most messages used in the thesis do not require more payload, certain messages (such as bodyplans of entire topologies described in Section 3.2.2) may require the transmission of more bytes at once. In these cases, we create a temporary ad-hoc wireless Ethernet connection between two robots and send the excess data over this connection. The foot-bot communication layer has been extended to make the creation and reading of such data seamless to all robot controllers using the device.

For application scenarios that seek for truly scalable solutions or require the simultaneous deployment of more than 25 mxRAB devices, the current implementation needs to be improved. The 25 robot limit thus need to be raised by reducing the refresh rate. Alternatively, the dynamic formation of local robots groups that perform time division multiplexing independently could be considered. Each group in this case may have a maximum size 25 robots, but the total number of groups, and therefore the total number of robots in the system, could be potentially unlimited. The operation of mxRAB devices at short distances is not reliable. Due to infra-red saturation, range cannot be estimated for distances shorter than 30 cm. At distances shorter than 15 cm, also the bearing estimation is noisy and becomes unreliable. These shortcomings of the mxRAB device need to be considered when designing behavioral control or communication algorithms that rely on the positioning information provided by the device.

Table 3.2: Messages exchanged when executing EDSA

Message type	Message payload
`AVAILABLE`	*bodyplan*
`RECRUITED`	ID
`GUIDANCE`	α, σ_i, ID
`CONNECTED`	σ_i, ID

We have listed all messages exchanged between two connection forming foot-bots in Tab. 3.2. The message `AVAILABLE` is used to announce availability by either an unconnected foot-bot or by another brain robot that has already invited and received previous connections. The message payload is the robot's bodyplan containing the location and orientation of all its sensors and actuators. The bodyplan of a particular marXbot configuration (see Section 2) allows an extending robot, for instance, to differentiate between a foot-bot and a manipulator foot-bot on the basis of a received message. Alternatively, the message payload can also provide the bodyplan of an entire robot topology as discussed in Section 3.2.2 where also the format of the message payload is provided. For the sake of clarity, however, we assume in this section that a foot-bot that can be invited to form a connection has not invited and received any other connection. A recruit, i.e., a foot-bot that has already been recruited by an extending robot, sends the message `RECRUITED` while maneuvering towards the extension point. In order to be distinguishable to multiple simultaneously extending robots, the mxRAB ID of the extending robot is included in the `RECRUITED` message payload. The remaining two messages and their semantics are explained in the following Section 3.1.2. In that section, we use the set $R = \{r_1, \ldots, r_n\}$ to denote the output of the mxRAB device, where r_i is a neighboring foot-bot and $n \leq 24$ is the number of foot-bots that are within the range of 5 m. Each $r \in R$ is a tuple of the form (ID, *range*, *bearing*, *message*).

3.1.2 Recruitment, guidance, and maneuvering algorithms

The recruitment and guidance algorithms are based on R – the set accessible to each foot-bot with all recent messages transmitted by peer foot-bots including their relative positions. The algorithms are sequentially executed by an extending robot in each control step to recruit an unconnected foot-bot or a brain robot (with other robots attached) and guide it precisely to the location on the chassis where a connection is required. The recruitment may be dictated by the actual task, that requires a connection in order to acquire a certain capability from another particular robot (for instance a foot-bot inviting connection from a manipulator foot-bot in order to acquire the capabilities of the magnetic gripper) or to extend the current morphology by an already connected structure such that the resulting morphology resembles the one required by the task. Both decisions can be based on the *bodyplan* sent by peer robot sending the `AVAILABLE` message. If a connection is required without defining a particular bodyplan, the recruitment and guidance algorithms extend the morphology by inviting the robot (independent of its capabilities and morphology) with the shortest bearing to the extension point.

A connection can be formed at any given angle σ around the perimeter of a foot-bot's chassis, except between 340° and 20° where the docking unit is mounted. In the remaining arc that corresponds to 320°, up to six foot-bots can be simultaneously

Algorithm 1: The extending robot's recruitment logic.

Input: Output of the mxRAB device R, extension points P
Output: Map $M : p_i \in P \mapsto r_i \in R$

```
1  foreach Extension point p_i ∈ P do
2      r_r ←GetCurrentRecruit(M, p_i);
3      foreach r_i ∈ R do
4          if r_i is available || r_i is recruited by myself then
5              if r_i is not recruited for another extension point then
6                  if GetBearing(r_i) <= GetBearing(r_r) then
7                      β ←GetBodyPlan(r_i);
8                      γ ←GetBodyPlan(r_r);
9                      if β is more suitable than γ then
10                         M ←Recruit(p_i, r_i);
11                     end
12                 end
13             end
14         end
15     end
16 end
```

connected assuming no two extension angles are closer than 53° to each other. For each extension angle σ_i, the corresponding (x, y) coordinates (hereafter referred to as an extension point p_i) is computed in cm from the center of the robot to the edge of the robot chassis where the connection is required. As described in Section 3.2.2, this conversion allows brain robots of larger morphologies to reason in distances encompassing multiple robotic units rather than in angles that can be associated with a single robot only. Given a set $P = \{p_1, \ldots, p_n\}$ of extension points, the objective of an extending robot is twofold. First, recruit the closest robot r_i with the appropriate bodyplan for each p_i and update map M such that each p_i maps to a r_i, if available. Second, guide each recruit r_i in M to navigate and connect to p_i by providing it with sufficient information for the maneuver. The recruitment is continuously revised in each control step allowing the extending robot to detect robots that become newly available to react to a malfunctioning recruit. For instance, if a new robot with a more appropriate bodyplan or location becomes available for a given extension point, the current recruit is released (i.e., no more guidance information is sent) and the new robot is recruited (i.e., it receives guidance information).

In Algorithm 1, we present a pseudocode of the recruitment algorithm. As shown on line 4, an extending robot either recruits a robot sending the message `AVAILABLE`, or re-recruits a robot that has already been recruited and is sending the message `RECRUITED` with the unique ID of extending robot's mxRAB device. Bearing measurements of peer robots retrieved on line 4 are absolute values. The comparison between two bodyplans shown on line 9 may depend on the task.

In case the recruitment has been executed a brain robot with child units attached to it, the brain unit can assign the execution of the guidance algorithm to the robotic unit where the connection is actually required. This assignment requires the conversion of set M into N in which each extension point p_i is converted into σ_i, the angle of the robotic unit at which the connection is required. More details are provided in Section 3.3, where communication between brain unit and its children

Algorithm 2: The extending robot's guidance logic.

Input: Output of the mxRAB device R, map $N : \sigma_i \in \Sigma \mapsto r_i \in R$
Output: Send tailored instructions to r_σ

```
1 foreach Extension angle σ_i ∈ N do
2     σ_i ←GetBearing(p_i);
3     r_σ ←GetRecruit(σ_i, N);
4     if r_σ was guided less often than another recruit ||
        all recruits were guided equally often then
5         SendMessage(GUIDANCE, α, σ_i, ID);
6         break;
7     end
8 end
```

are discussed.

A recruit can only maneuver to the designated location where a connection is expected if sufficient guidance information is provided by the extending robot. This guidance information is sent using the `GUIDANCE` message which contains tailored instructions applicable to the recruit. These instructions include the bearing α at which the extending robot perceives the recruit, i.e., the inverse bearing of the recruit, the target bearing σ_i (i.e., the bearing at which the connection is anticipated by the extending robot), and the unique ID of the recruit's mxRAB device. This is because, by default only broadcast communication is supported by the mxRAB device. The communication of unique IDs allows recipient foot-bots to determine whether a message is addressed to it or not by comparing its own unique ID with the unique ID in the received message. Therefore, the unique ID included in the payload of the `GUIDANCE` messages allows the extending robot to address a single recruit despite using a broadcast message. Further, the mxRAB hardware only allows the transmission of a single message in each control step. Thus, if multiple recruits are being simultaneously guided by the extending robot, it must take turns sending the `GUIDANCE` message to each recruit. For example, in the extreme case in which six recruits are being simultaneously guided, each recruit can only be guided every seventh control step – that is every 700 ms. This logic is presented on line 4 in the pseudocode of the guidance algorithm shown in Algorithm 2. Although this frequency is sufficient to let recruits maneuver at maximum speed to the connection location, it could be a potential issue in a system that can support many more extension points.

An extending robot executes both recruitment and guidance algorithms until all extension points are filled. An extension point σ_i is filled when a recruit acknowledges a successful docking with the message `CONNECTED` with σ_i and the unique ID of extending robot's mxRAB device in its message payload.

Recruits are initially selected from all robots sending the message `AVAILABLE`. A robot acknowledges its recruitment when a `GUIDANCE` message is received with the unique ID of its own mxRAB device in the message payload. Acknowledgement is confirmed using the message `RECRUITED` with the unique ID of the extending robot's mxRAB device in the message payload. A recruit can be located within a radius of 5 m around the extending robot and maneuver to an extending robot's target solely based on the instructions received in the `GUIDANCE` message. The trajectory of a recruit to the extension point is chosen such that the imprecisions of the mxRAB

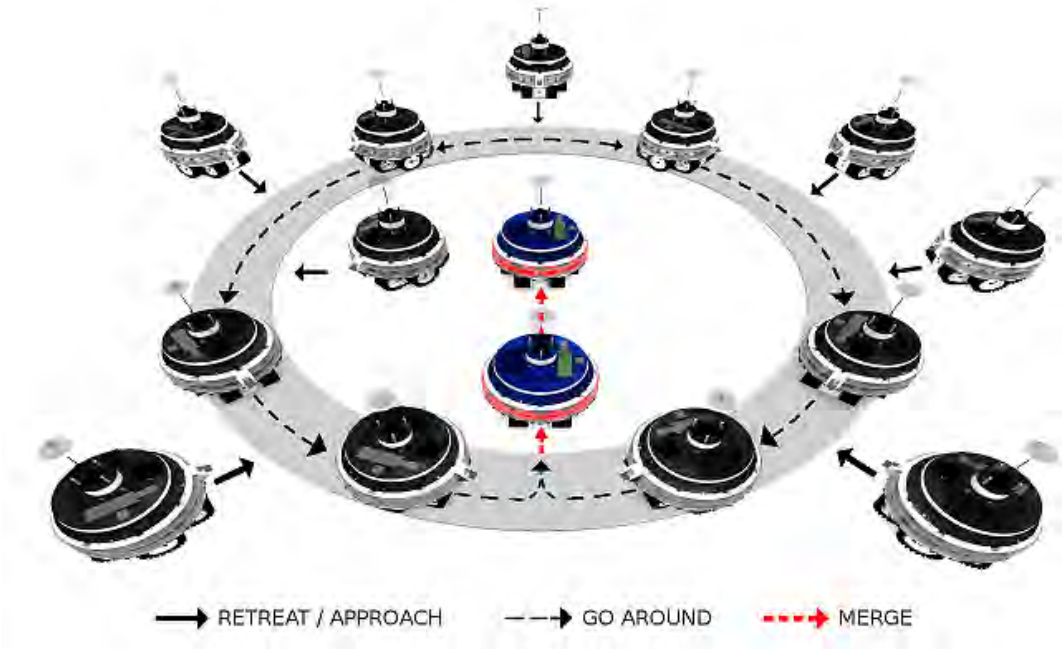
device at close distances are taken into account. Note that the range estimations for distances shorter than 30 cm are unavailable and that the bearing estimations for distances shorter than 15 cm are unreliable. We illustrate possible trajectories that could be taken by recruits starting from different positions around an extending robot in Figure 3.3a. From a control perspective, the trajectory taken by a recruit to an extension point can be split into three different segments or states that are shown using a finite state machine in Figure 3.3b: i) retreat or approach the extending robot to a certain distance, ii) go around the extending robot until aligned to the extension point, and iii) merge to the extension point.

A recruit's initial state is retreat/approach. In this state, the recruit moves away or towards the extending robot until within a ring shaped area around the extending robot with an inner radius of 40 cm and an outer radius of 60 cm. We refer to this ring as the go-around ring. This state permits the recruit to reposition itself to a distance where the mxRAB device can accurately estimate the range to the extending robot. Once inside the go-around ring, the recruit enters the go around state. In this state, the recruit navigates around the extending robot whilst staying in the go-around ring. The recruit can navigate either in clockwise or in counter-clockwise direction, whichever gives the shortest radial distance between the inverse bearing α and the target bearing σ_i received from the extending robot. Distance and bearing to the extension point is continually provided by the guidance messages sent by the recruiting constituent robot. If a recruit drives out of the go-around ring, it switches back to the retreat/approach distance state. The transition to state merge is triggered once the absolute difference between α and σ_i is less than 2. In the merge state, the recruit rotates until it faces the extension point with its docking unit. It then moves towards the extension point while updating its bearing estimation β to the extending robot using the following filtering method:

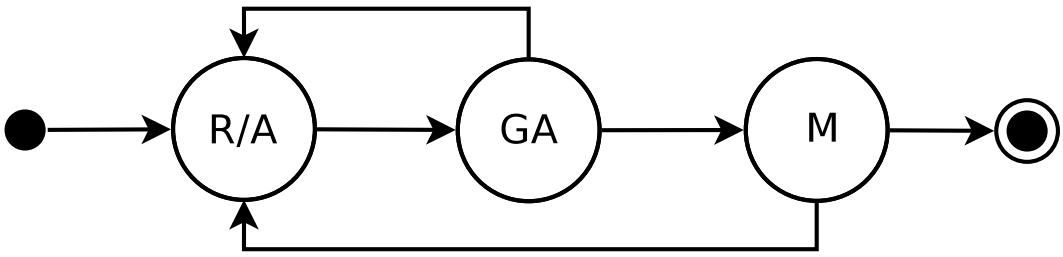
$$\beta_t = (1 - c) \cdot \beta_{t-1} + c \cdot \beta_t,$$

where t is the control step and c is $0 \leq c \leq 1$ and is the confidence level of a bearing estimation at close distance. This filtering method was implemented as the recruit is entering an area in which bearing estimations to the extending robot become unreliable. Therefore, the filtering method is applied to make recruits less reactive to noisy bearing estimations while still being able to react to major environmental changes such as a moving of the extending robot. We empirically determined that $c = 0.7$ is appropriate. The recruit uses β in the merge state when maneuvering towards the extension point until until the force sensor in the gripper registers contact with the extending robot's docking ring. The recruit then stops and forms a physical connection to the extending robot by opening the three fingers of its gripper inside the docking ring. The success of a connection is tested by the recruit by trying to rotate its docking module. If the rotation can be carried out, the connection has failed and the recruit goes back to the keep distance state. If the docking module cannot be rotated, the connection is firm. The message `CONNECTED` is sent to acknowledge the successful connection to the extending robot. The message payload includes σ_i and the unique ID of the extending robot's mxRAB device.

If a recruit does not receive `GUIDANCE` messages for 10 consecutive control steps, it comes to a halt assuming that the extended robot has released the recruitment. This may occur when a better located robot (or a robot with a more suitable morphology) becomes available after the initial recruitment. The halted robot announces its availability to other extending robots by sending the message `AVAILABLE`.



(a) The trajectories taken by potential recruits to maneuver to an extension point at the rear of the extending robot (in the center). The three arrow types represent the three segments of a recruit's trajectory. The shaded, ring shaped area represents the go around zone.



(b) The finite state machine representation the recruit's behavior. `R/A` = retreat/approach, `GA` = go around, and `M` = merge.

Figure 3.3: An illustration of the recruit's behavior: (a) the trajectories taken by potential recruits, (b) the finite state machine that segments a trajectory into three.

3.1.3 Speed, precision and other features

We study the speed and precision with which connections can be formed using EDSA. We compare the results to those achieved using "directional self-assembly" presented in [42]. For this purpose, we replicated the experimental setup presented in [42] by placing an extending robot with an open extension point to its rear (i.e., at 180°) in the center of a circle of 80 cm radius. The radius is the only difference compared to the setup used in [42] where it was 35 cm. We then placed a second robot at 12 equally separated starting positions on the circle. We considered 8 starting orientations for each starting position. For each combination of starting position and starting orientation, we let the robots execute EDSA. This resulted in 96 independent trials for which we recorded the elapsed time until a firm connection was formed. We also took images of the robots after a connection had occurred using an overhead camera placed above the center of the extending robot. We measured the angular precision (or the misalignment between where the connection

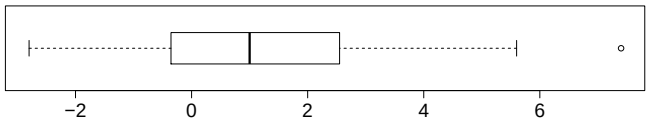


Figure 3.4: Box-plot showing the misalignment (in degrees) between the orientation of the recruit and the orientation of the extending robot after forming a connection.

was expected and where the connection actually occurred) using an open source tool named ImageJ: `http://rsbweb.nih.gov/ij/`.

The results of the angular precision are shown using a box-plot in Figure 3.4. It shows the misalignment measured in the orientation of both robots after the connection occurred. Note that an entirely aligned connection would result in a misalignment of 0°. The measured median of the misalignment is 1° making it comparable to the results reported in [42]. However, the distribution range of the measured misalignment is with 10° (interval [-3°,7°]) approximately 7x smaller than the results obtained in [42] (distribution 70°, interval [-30°,40°]). The mean measured in our experiments is 1.2°) and the standard deviation is 0.77°. The worst most extreme misalignment measured in our experiments was with 7° approximately 5x better than the 39° reported in [42]. Using the analysis conducted here, we conclude that the angular precision that can be achieved using EDSA is on average more accurate and reliable than that achieved using the mechanism reported in [42].

In terms of speed, a recruit spent on average 16.1 s maneuvering to the extension point. The comparable value presented in [42] was 54.3 s making EDSA is 3.5x faster than its predecessor mechanism. This result was achieved despite the fact that in our experiments the second robot was placed more than twice as far from the extending robot (i.e., 80 cm instead of 35 cm). We identified three factors that contribute to these results. First, the foot-bots can drive at higher speeds to the extension point than the s-bots in [42]. This is due to the fact that the mxRAB device provides the foot-bots with faster and more precise estimates of relative positions to neighboring robots than the camera and LEDs-based mechanism used by the s-bots. Second, the active guidance information provided by the extending robot allowed the recruit to chose the shortest orbiting direction around the extending robot whereas in [42], by contrast, it was chosen randomly. The third factor is inherent in the design of the robots. The foot-bots are able to form faster physical connections using the innovative docking module (that includes the docking unit with the three fingers and the docking ring) than s-bots using their gripper based mechanism.

In our experiments, all 96 trials resulted in successful connections between the extending robot and the recruit. However, we repeated 5 trials that were corrupted due to low battery voltage of the robot. In the following, we use Figure 3.5, in which snapshots from experiments are shown, to present some features of EDSA that now become available to a morphology growth system based on self-assembling foot-bots. Video footage of these features can be seen online at `http://iridia.ulb.ac.be/supp/IridiaSupp2011-001/`.

3.1.3.1 Adaptive recruitment

Self-assembling robots need to be able to adapt to changing mission conditions. The recruitment algorithm at the core of EDSA is able to adapt to such conditions including to malfunctioning recruits or to the availability of better suited robots. In

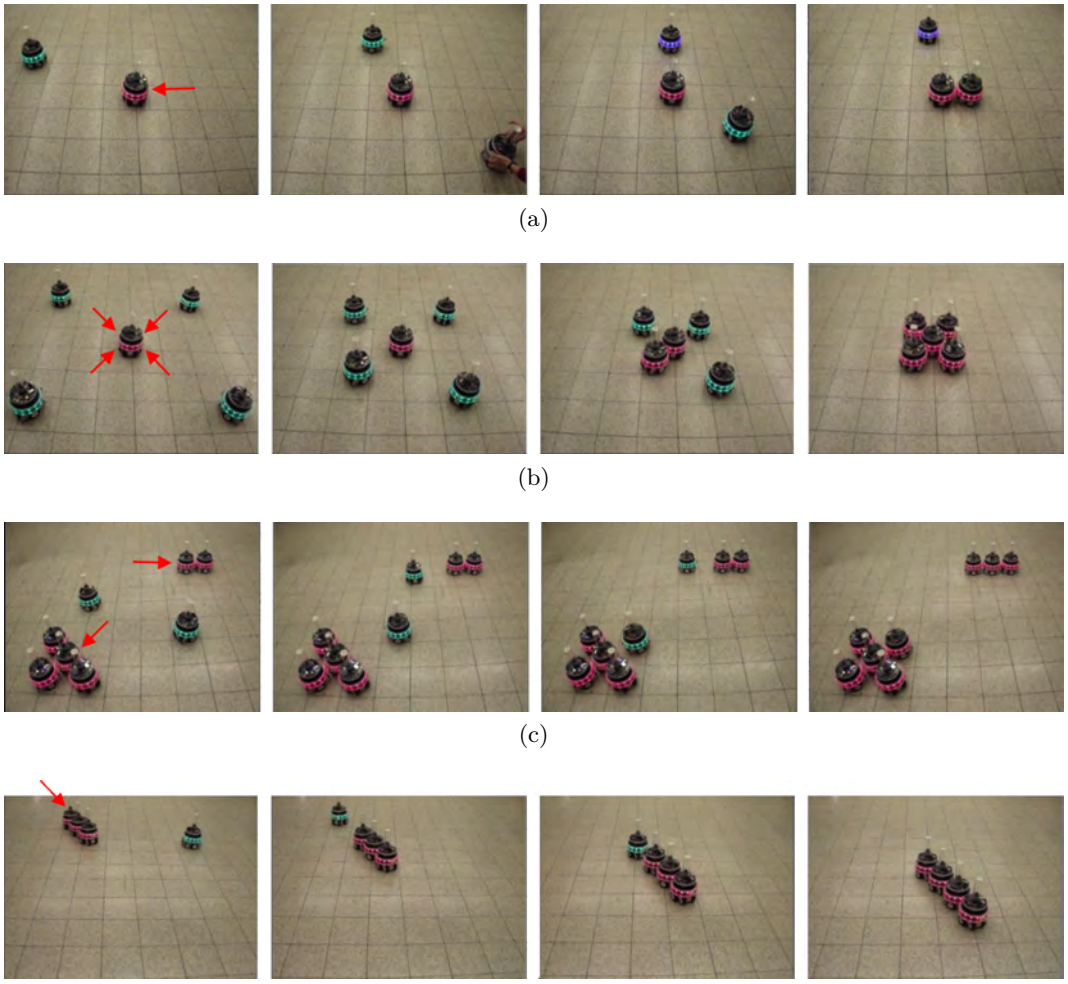


Figure 3.5: New features provided by EDSA. Robots attached to a morphology display their red LEDs while unconnected robots display their green LEDs. Robot illuminating blue LEDs has lost its recruitment. Extension points are indicated using red arrows. (a) Adaptive recruitment. (b) Enhanced parallelism. (c) Resource allocation without interference. (d) Morphology growth in motion.

Figure 3.5a, we show snapshots from an experiment in which an extending robot is able to release an existing recruitment and recruit a robot that become available after the recruitment of a first robot. This is possible because the high-speed communication provided by the mxRAB device allows the mapping from extension points to recruited robots to be updated at every control. Hence, EDSA is able to adapt to new conditions while maintaining an optimal resource allocation w.r.t. the number of robots allocated per extension point. In the experiment shown in Figure 3.5a, the extending robot placed in the center of the frame is shown to have recruited the only available robot to connect to an extension point at 90°. When a new robot is introduced closer to the extension point, the extending robot adapts to the new situation by releasing the initial recruitment and by recruiting the robot that has new become available. The robot that was initially recruited leaves the self-assembly process and becomes available for other tasks.

3.1.3.2 Enhanced parallelism

In "directional self-assembly" – the mechanism used by the s-bots for connection forming – an extending robot could only invite connections in a sequential manner. This inhibited the s-bots from terminating sophisticated morphologies rather quickly. EDSA, on the other hand, permits parallel connection formation and hence helps extending foot-bots seeking to branch a morphology. A branching occurs when a robot invites multiple free robots for connection. Figure 3.5a shows snapshots of an experiment in which an extending robot seeks to fill four extension points by inviting connections from four free robots located in the vicinity. The snapshots shows how the extending robot recruits and guides four recruits in parallel, each one to its closest extension point. The extension points are located 45°, 135°, 225°, and at 315° of the extending robot. All four connections are formed within 16 s to form a star-like morphology.

Parallel connection formation provided by EDSA is not limited to connections formed on a single extending robot. Extension points can be distributed among separate morphologies and still be filled in parallel using EDSA. That is, extending robots in each other's communication range can recruit and guide nearby free robots without interfering with each other. Figure 3.5c shows snapshots of an experiment that demonstrates this feature. Two morphologies, a chain-like morphology shown on the top-right and star-like morphology shown in the bottom-right, are shown to be grown next to each other in parallel. Each extending robot executing EDSA shares the available resources without hindering any other ongoing growth process.

3.1.3.3 Morphology growth in motion

The high-speed communication and precision inherent to the mxRAB device allows EDSA to demonstrate morphology growth of a moving morphology for the first time. This feature could play a crucial role in real-world systems as it may allow morphology formation times to be reduced significantly. For instance, a transportation task could already be started during an ongoing morphology growth process. Figure 3.5d shows snapshots from an experiment in which a chain-like morphology is extended during motion. The third link of the morphology opens an extension point at its 180° and recruits a free robot that is static. The recruit first aligns itself to the extension point, and then forms the connection by driving at a velocity that is higher than that of the extending robot in motion.

3.2 From EDSA to topology-aware larger morphologies

A foot-bot is controlled by a program that processes the data returned by its sensors and makes decisions based on the data about future actions to take. Essentially, each foot-bot has a centralized decision-making entity – a brain – that also is aware of the body (such as chassis dimensions and relative positions of sensors and actuators) it controls. The collective robot structures formed using EDSA, on the other hand, have as many brains as there are foot-bots in the respective structures and no foot-bot is aware of the topology of the overall structure it is a member of. Additionally, EDSA represents a basic connection forming mechanism between a connection inviting and a connection seeking robot and is not sufficient to form larger morphologies as the one shown in Figure 3.5d. These morphologies require connection

Algorithm 3: An extended version of a SWARMORPH-script describing the formation of a chain-morphology of size three.

```
1  if seed then
2  |  switchLeds(red)
3  |  // Toggle on extension points from the pre-defined
4  |  // set of angles (45, 90, 135, 180, 225, 270, 315)
5  |  extensionPoints(0 0 0 1 0 0 0)
6  |  // Send further branching/growth instructions for filled extension points
7  |  sendBranchingInstructions(0 0 0 1 0 0 0)
8  |  awaitAckOnTermination()
9  end
10 if not seed then
11 |  switchLeds(green);
12 |  searchForRecruitment();
13 |  maneuverToExtentionPoint();
14 |  ep[] ← receiveBranchingInstructions();
15 |  // Instructions for middle foot-bot:
16 |  if sum(ep[]) != 0 then
17 |  |  bi[] ← decrementReceivedBranchingInstructions(ep[])
18 |  |  extensionPoints(ep[]);
19 |  |  sendBranchingInstructions(bi[]);
20 |  |  propagateAckToParent();
21 |  end
22 |  // Instructions for the final foot-bot:
23 |  if sum(ep[]) == 0 then
24 |  |  sendAckToParent();
25 |  end
26 end
```

formation to be controlled within an entire group of self-assembling robots and require connections to be formed between robots that did not initiate the self-assembly process.

In this section, we show how we rely on a mechanism previously developed at our laboratory to control the formation of larger morphologies. We then present further mechanisms we developed to ensure that a morphology is controlled by a single brain unit that is aware of the body it controls at all times.

3.2.1 Controlled morphology formation

To enable the formation of larger morphologies in a controlled manner, we rely on an extended version of a morphology control language presented in previous research named SWARMORPH-script [3]. SWARMORPH-scripts contain pre-defined sequences of instructions or atomic self-assembly related behaviors describing the formation of a precise morphology. They can be executed by any number of autonomous robots in parallel and enable the formation of arbitrary morphologies in a distributed manner. Refer to Algorithm 3 for an example SWARMORPH-script that describes the formation of a line or chain morphology of size three, that is, a linear structure composed of three robots in which each robot besides the first one – the seed – is connected to the rear of the preceding robot. Note that, the seed robot needs to be pre-defined using other mechanisms such that different logic becomes available to seed and non-seed robots.

A key feature underlying SWARMORPH-scripts is the serialization mechanism that allows entire control logics to be translated into binary strings that then can be exchanged between robots. A robot receiving the serialized string can, in turn, de-serialize it in a way that it is immediately executable. This feature is particularly useful when control logics describing morphologies crucial to a task needs to be sent

from a robot with a better perception of the environment to another or when target morphologies have been determined or learned by robots through costly methods and need to be shared within a group of peer robots.

SWARMORPH-scripts were initially developed for the s-bot platform that preceded the foot-bots and was limited to LEDs and cameras-based communication between robots. In our extended version, we replace this low-bandwidth communication modality to make full use of the higher bandwidth and speed provided by the mxRAB device available to the foot-bots. Additionally, when required ad-hoc WiFi connections can be created between foot-bots to transfer even higher payload than the 10 bytes supported by the mxRAB device. We also integrated EDSA-related behaviors such that the recruitment guidance algorithms are translated into appropriate atomic behaviors. Contrary to is predecessor, the extended version of SWARMORPH-scripts supports the formation of multiple connections in parallel, coordinated motion, communication with multiple robot platforms (for instance a control logic can be received either from another foot-bot or an aerial robot), and a variety of merged nervous nervous system related behaviors – a comprehensive list is provided in Table 3.3 presented in Section 3.3.

3.2.2 Topology and partial failure representation

To form a new unified robot with a single brain aware of the morphology it controls, all but one of these units need to cede decision-making authority while only a single unit assumes the role of the brain. To this end, we impose a hierarchical tree structure (i.e., a directed tree) on the physical connection topology of the robotic units. In an MNS robot, the root unit of the tree structure can always be unambiguously identified and hence assumes the role of the brain. The brain of any further connecting robot will cede authority to this root unit. For the brain unit to exert control over the morphology, it must be aware of the relative placement of sensors, actuators, and other subsystems available to the robot. In practice, for larger morphologies, the majority of the sensors and actuators a brain unit will be in charge of will be found on the bodies of other robotic units.

In an MNS robot, each constituent unit maintains a recursive representation of its and its descendants hardware configurations together with their relative positions. Hardware configuration of a robotic unit is a set of geometric relationships that includes the geometry of a robotic unit and the physical arrangement of sensors and actuators in the robotic unit. Providing each robotic unit with the ability to store and transmit internal representations is a key feature of MNS robots. It ensures that a brain robot's self-knowledge can be rapidly updated to reflect changes occurring anywhere in the physical composition of the morphology.

A robotic unit can also represent failures of its particular sensors and actuators in the internal representation such that they also become available to the brain unit. This ability to represent failures at the sensor/actuator-level becomes more prominent for larger MNS robots due to the increased likelihood of partial failures. Providing brain units with information regarding malfunctioning sensors and actuators also allow MNS robots to be fault tolerant and compensate for such failures. Note that the method applied by each robotic unit to detect partial failures [44] is not relevant to the brain unit. A detected partial failure is reported to the brain unit using the same update mechanism used to inform the brain about physical changes occurring to the morphology such a merge or a split. The mechanism, presented in the following Section 3.2.3, allows all ancestor units (parent, parent's parent, etc.)

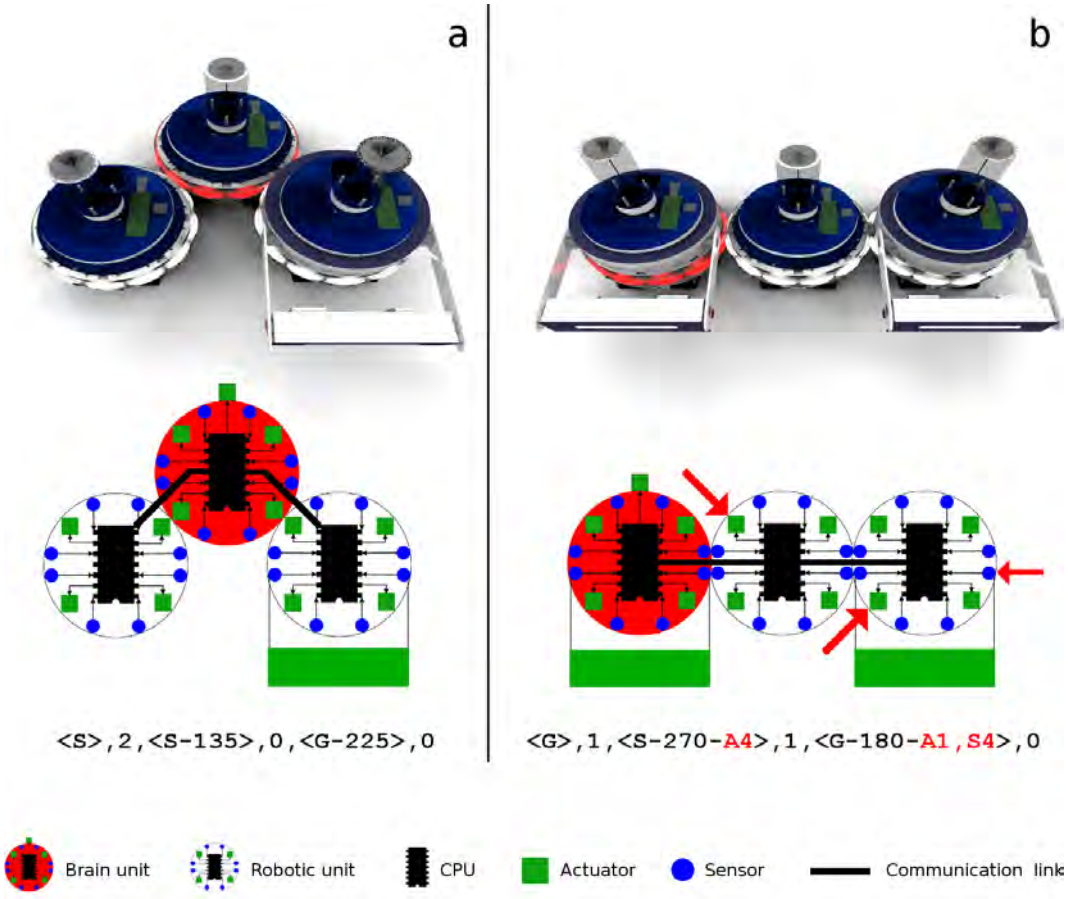


Figure 3.6: Internal representations of two different three-unit MNS robots with different morphologies. (a,b) The configuration of robotic units in the MNS robot is shown in using the rendering at the top of each pane. The corresponding merged nervous system topology is shown in the illustration underneath which also visualizes how sensors and actuators on descendant robot unit become accessible to the brain units. Each brain unit's internal representation is given as serialized strings at the bottom of each pane. For brevity, we use the letters 'S' and 'G' instead of the detailed geometric descriptions of the foot-bot and the manipulator foot-bot, respectively. Communication links between robotic units rely on ad-hoc wireless Ethernet connections. (b) The arrows indicate partial failures (i.e., faulty sensors or actuators) that also highlighted using the color red in the serialized string.

to update their internal representations with the knowledge of the partial failure.

The modular nature of the marXbot platform allows us to consider a variety of different hardware configurations in our experiments. In this thesis, we consider the standard foot-bot and the manipulator foot-bot as two different hardware configurations. In Figure 3.6, MNS robots formed with robotic units of two different hardware configurations are shown. In the figure, internal representations are also shown as strings at the bottom of each pane. This string is also used to communicate internal representations between robotic units. A recursive serialization method is used to generate the string that has the following syntax:

```
MNS:=<[robot-type]-[connection-angle]-[malfunctioning-
      sensors/actuators]>,[num-children],[MNS]
```

3.2.3 Single message-based topology update mechanism

An internal representation that accurately maps the MNS robot morphology allows the robot to react to radical morphological changes promptly. For instance, when an MNS robot splits into multiple robots with separate bodies, each root unit of the uncoupling body segments already has all the knowledge it needs to become the brain of the new independent MNS robot. It is therefore crucial for merged nervous systems to follow the physical connection topology at all times and by updating its knowledge after changes such as splits or merges occur.

Updates are propagated in form of messages transmitted from robotic units to the brain unit. For example, when a merge between two MNS robots occurs, a single message is transmitted up the merged nervous system from the connecting MNS robot ceding authority to the brain of the MNS robot to which it connects to. The information contained in the message is the internal representation (serialized string) of its own previously independent body and is incrementally updated by each intermediate unit along the path to the brain until the brain unit has an up-to-date representation of the new body. The newly formed MNS robot incorporates all the sensing, actuation and computational capabilities of the units in the new body. This instantaneous update mechanism discredits an MNS robot's need for time-consuming processes such as self-discovery [45], trial-and-error [46] or hormone-based messaging [47] to discover changes occurring to its connection topology.

We present a visualization of the update mechanism in Figure 3.7 using two MNS robots. In Figure 3.7b, an MNS robot composed of six independent units is shown to merge into another MNS robot composed of four independent units. The brain unit of the robot that is attaching (left robot) cedes authority to the brain unit of the robot to which it is attaching (right robot). The brain unit of the new merged robot still does not have an accurate internal representation of its new morphology. However, its child unit has already updated its internal representation. In Figure 3.7c, information about the morphology has propagated to the brain leaving all independent units in the merged nervous systems with an accurate representation of the morphology.

3.3 MNS robot control

In this section, we first describe the sensorimotor and spatio-temporal coordination mechanism we developed to coordinate MNS robots. We then present scalability properties of the proposed method and describe the heartbeat mechanism used to detect failures (i.e., entire robotic units that stop functioning) within an MNS robot.

The high-level control of an MNS robot is the responsibility of its brain unit. The approach taken by most robot control programs — to statically bind information regarding the placement of sensors, actuators and other subsystems into the control logic — is impractical to be adopted in the case of MNS robots as they are capable of changing their body compositions during mission time. Our solution is to express the control of an MNS robot using a high-level logic that is independent of the morphology it controls and do not consider the number and location of sensors and actuators available to the robot. That is, a merged nervous system divorces the control logic from the physical properties of the robot. Taking inspiration from nature [48], a merged nervous system allows the brain of an MNS robot to issue and propagate high-level commands about actions the body should take which are then

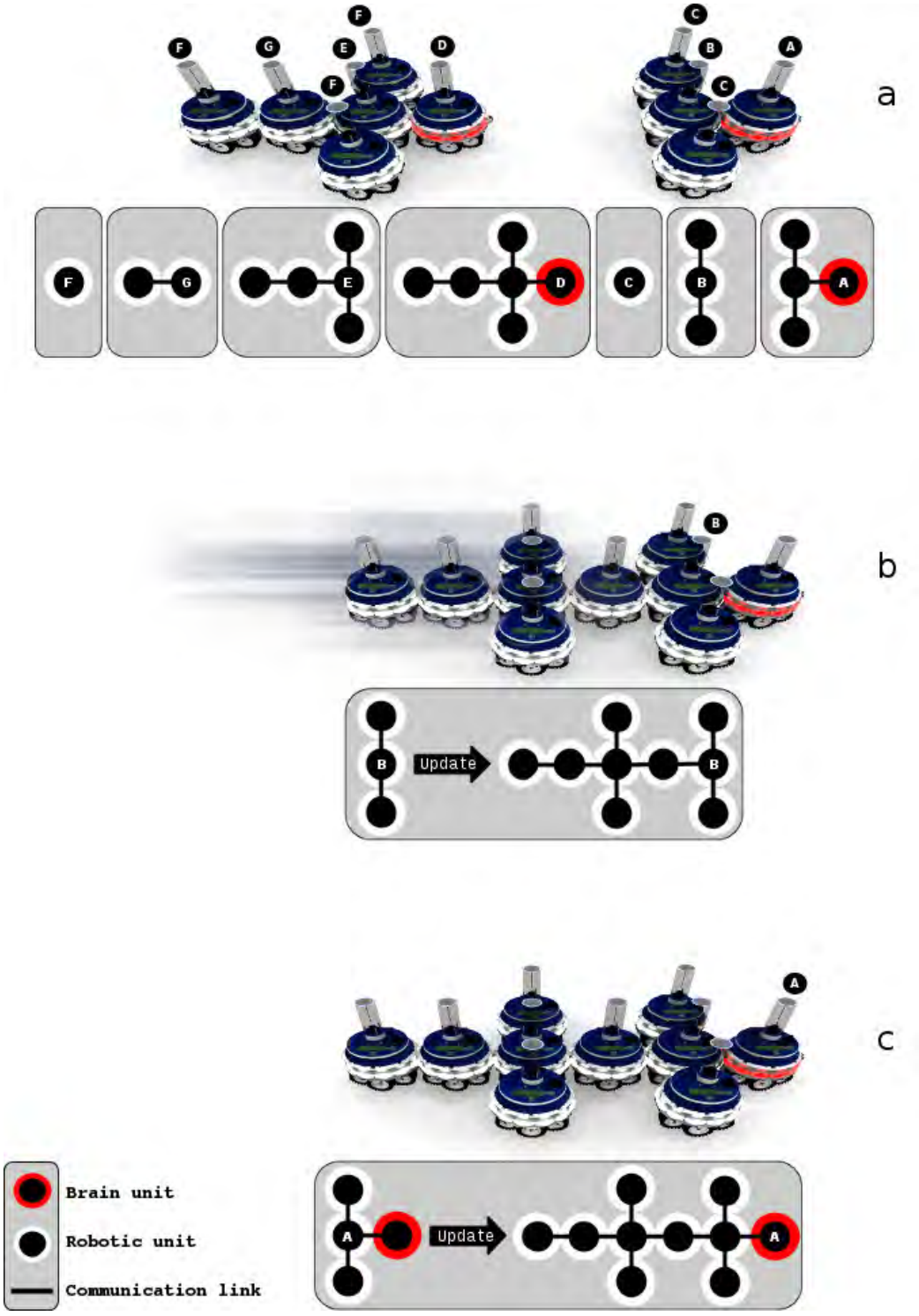


Figure 3.7: Propagation of internal representation triggered by the merging of two MNS robots. The internal representations of robotic units are shown using insets with shaded backgrounds. In each of the inset, the robot whose internal representation is indicated using corresponding letters. Note that units that are not the brain unit only have knowledge of their descendant units.

seamlessly translated into instructions for individual actuators in each constituent unit. Data resulting from the sensors are processed by the many constituent robotic units in a distributed fashion and then fused into an abstraction of the environment. This description is propagated through the merged nervous system such that it is immediately meaningful to the brain unit on arrival.

We detail how responsibility is delegated as part of the sensorimotor coordination mechanism in an MNS robot using the example shown in Figure 3.8. The figure shows a 5-unit MNS robot in the center that has detected and then responded to a stimulus. The stimulus is a mobile light source shown in green and the response requires the MNS robot to point at the stimulus by illuminating its three closest LEDs. In Figure 3.8a, we show how a stimulus moves into the sensor range of the MNS robot. More precisely, the stimulus moves into the sensor range of two robotic units which separately detect the stimulus and its relative location by analyzing their camera feeds. Computationally intensive tasks such as visual image processing is executed in a distributed manner in an MNS robot and only an abstraction of this information (e.g., existence and relative coordinates of the stimulus) is propagated to parent units. A parent unit receiving such abstracted information from multiple children fuses the information to form a more accurate estimate of the stimulus' relative location. The underlying method used to fuse sensor is irrelevant to the merged nervous system and may depend on the type of sensor and precision required for the task. The single item of fused information is then propagated by the parent unit (see Figure 3.8b) to its own parent, which in this case is the brain unit. Based on abstract information describing environmental changes occurring within or beyond its own sensor ranges, the brain unit makes decisions about what action to take, in this case, to select the LEDs to illuminate in green. As shown in Figure 3.8c, the actions are issued as high-level actuator commands that are then translated by the merged nervous system (see Figure 3.8d) into actuator instructions relevant to each robotic unit.

An addressing system integrated into the high-level commands allows the brain unit to target particular robotic units, or groups of robotic units. Address information is determined by the path along the tree-structure from the brain unit to the recipient of a message and is encoded into the header of the message. The path is composed of the sequence of relative connection angle between independent unit. For instance, a command issued with the address header [180-90-180] describes a path from the brain unit to the child unit connected at 180 degrees, to the next child unit connected at the brain unit's child at 90 degrees (the brain unit's grandchild), to finally reach the next child unit connected again at 180 degrees. The command issued therefore is intended to address the robotic unit at the end of the path in an L-shaped morphology. Multiple recipient robots can be addressed by replacing a connection angle using the wildcard '*' – requiring multiple-unit segments in the MNS robot to execute an issued command. A list of messages (commands and otherwise) available to MNS robot composed of foot-bots are shown in Table 3.3.

Although specific actuators and sensors in individual units can be independently accessed when necessary, the brain unit of an MNS robot often issues high-level commands to control the actions taken by the robot. Such commands define how the constituent units of an MNS robot coordinate both temporally and spatially. Time-based coordination is achieved by the brain unit always launching actuator actions in the future. The length of the delay until the future action is chosen based on the size of the morphology. Each unit maintains its own countdown to action execution.

Table 3.3: List of messages propagated through a merged nervous system of footbots. The header of each message includes an address that allows the brain unit to target particular robotic units, or sets of robotic units. The format of the address is sequence of angles corresponding to connections between physically connected robotic units. The `<request type>` parameter consists of one of the following values: `[once|duration=x,frequency=y|conditional=if_changed]`.

Command	**Payload**
Messages addressed to specific units or groups of units	
Extend [*address*]	`<extension point>; <recruit id>`
Disconnect [*address*]	`<extension point>`
SetLedColors [*address*]	`<led color 1>;...;<led color 12>`
Lift [*address*]	`<object bearing>`
GetCameraObjects [*address*]	`<request type>`
GetBatteryLevel [*address*]	`<request type>`
GetAccelerometerReading [*address*]	`<request type>`
GetProximitySensorReadings [*address*]	`<request type>`
GetGroundSensorReadings [*address*]	`<request type>`
Messages addressed to all units	
Heartbeat	`<emission frequency>; <reaction failure =[disconnect_retract| become_brain]>`
Move	`<speed>;<heading>`
Rotate	`<center of rotation>; <angular velocity>`
Stop	`(none)`
Update and status messages send to parent units	
Update	`<topology representation>`
Status	`<success|failed>`

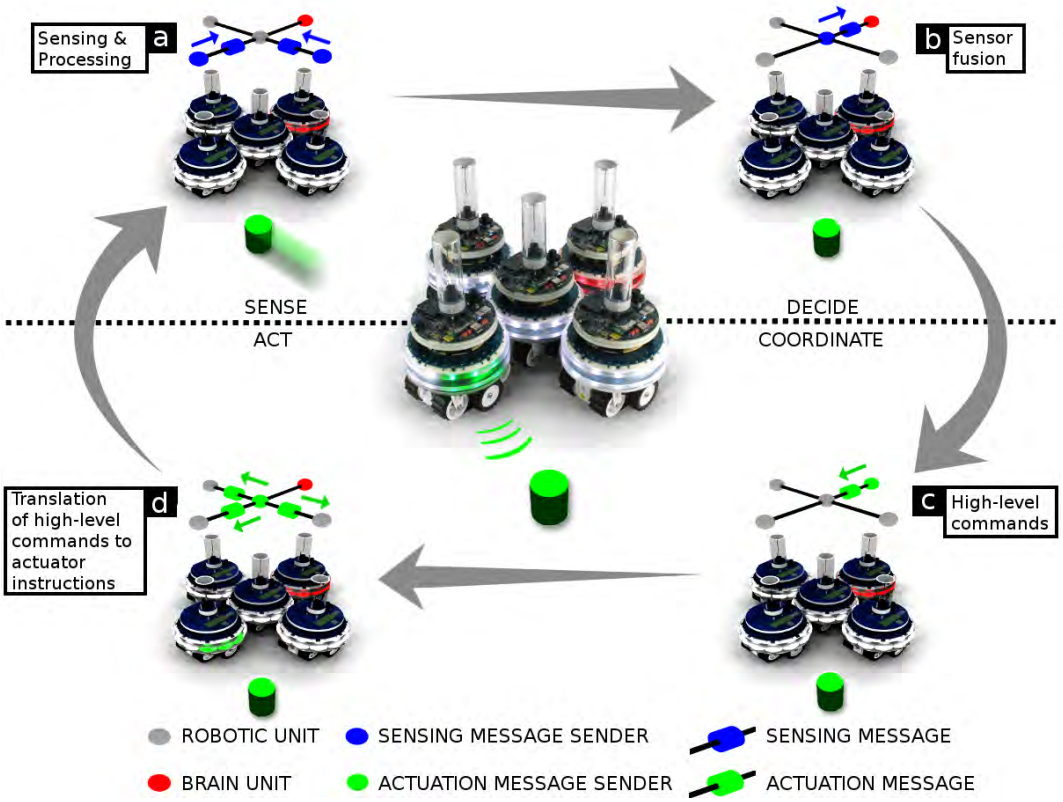


Figure 3.8: Responsibility delegation in a 5-unit MNS robot detecting and then responding to a stimulus. The MNS robot in the center shows the response outcome – highlighted using three concentric green lines shown on the ground – and the four renderings (a–d) show prior steps that led to the response. The mobile stimulus is shown in green and the merged nervous system and the messages propagated are shown above the robot in each step.

The countdown to the future action is reduced as the command gets passed from unit to unit, thus taking into account the message propagation delay. We achieve spatial coordination by enforcing robotic units to translate location-related information (such as coordinates of detected objects in the environment) every time a sensor or actuator message is propagated between units. Individual units use their knowledge of the relative locations of parent and child units to translate spatial references into the frame of reference of the receiving unit before message transmission. Messages therefore become immediately meaningful to a message recipient.

Figure 3.9 shows the spatio-temporal coordination mechanism in action. In the figure, the brain unit of a 3-unit MNS robot is shown to high-level commands that are then propagated through the merged nervous system. The MNS robot (on the left) has detected a red brick (on the right) that needs to be lifted and transported. Lifting capabilities are unique to the manipulator foot-bot requiring the MNS robot to plan and execute a rotational movement that aligns the object gripper next to the brick. To ensure temporal coordination, at timestep T0, the brain unit of the MNS robot starts a countdown of 0.4 seconds until execution. For clarity, each communication timestep is assumed to last 0.1 seconds. For example, T3 occurs 0.3 seconds after timestep T0. The countdown is chosen such that message propagation delay to furthest child unit is shorter than the countdown. The brain unit

also computes its wheel speeds and the center of rotation around which the MNS robot should turn. These information are propagated to the child (the middle unit) in the same timestep (T0). The merged nervous system takes the responsibility to translate the rotation coordinates into the middle unit's spatial frame of reference and converts the brain unit's wheel speeds into angular velocity. Subsequently, both bits of information are encoded into the payload of a "rotate" command listed in Table 3.3. At time T2, after a single timestep, the command reaches the middle unit triggering a countdown starting at 0.3 seconds (i.e., received countdown reduced by the propagation delay). At timestep T1, the middle unit initiates a similar communication process with its child unit as the brain unit in timestep T0. As a result, in timestep T4, all countdowns in the MNS robot terminate at the same time allowing the robot to execute the rotational movement in a coordinated manner. Note that the high-level "rotate" command does not contain instructions related to the wheel actuators of independent robotic units. Instead, the merged nervous system manipulates the command and provides individual units with tailored information sufficient to determine the appropriate wheel speeds (for example, the last unit will set higher wheel speeds). The rotation is stopped by the brain unit using a "stop" command as soon the brick is close enough to the manipulator foot-bot. Subsequently, a "lift" command is issued at timestep T62 together with the address of the unit that should execute the command. The address [270-180] part of the command uniquely identifies the manipulator foot-bot capable of lifting the brick as the recipient. Once arrived at the destination, the high-level command is translated into a sequence of precise actions that can be executed by the robotic unit. The final action in the sequence is to send a status message back to the brain unit to inform about the outcome of the task.

Independent of other transmitted commands, the brain unit generates and issues additional periodic commands at a fixed frequency to indicate normal operation of the MNS robot using a heartbeat protocol [49]. The payload of the heartbeat pulse is shown in Table 3.3 and includes information regarding the frequency at which further pulses can be expected by recipient robots and actions a recipient robot should take in case of an absent heartbeat pulse. A recipient of a heartbeat pulse acknowledges its reception to the parent and then propagates the pulse to its immediate children. In this manner, the heartbeat protocol allows an MNS robot to detect and react to totally malfunctioning robotic units. That is, a missing pulse within a previously received frequency indicates a faulty parent. On the other hand, the absence of an acknowledgement warns the parent of a faulty child. The robotic unit that detects a faulty parent refers to the reaction type pre-defined by the brain unit and acts accordingly. A child that becomes the brain unit of a sub-morphology starts issuing its own heartbeat pulses.

The frequency at which heartbeat pulses are issued can be altered by the brain unit during an ongoing mission. The frequency is chosen based on the size and shape of the MNS robot it controls and the level of reactivity required for the task at hand. For instance, a morphology with a high branching factor and low heartbeat pulse frequency can be just as reactive to faulty units as a long chain-like morphology with a high heartbeat pulse frequency.

3.3.1 Scalability

We consider two scalability aspects of the MNS control mechanism: computational scalability and scalability with respect to message propagation time (in either di-

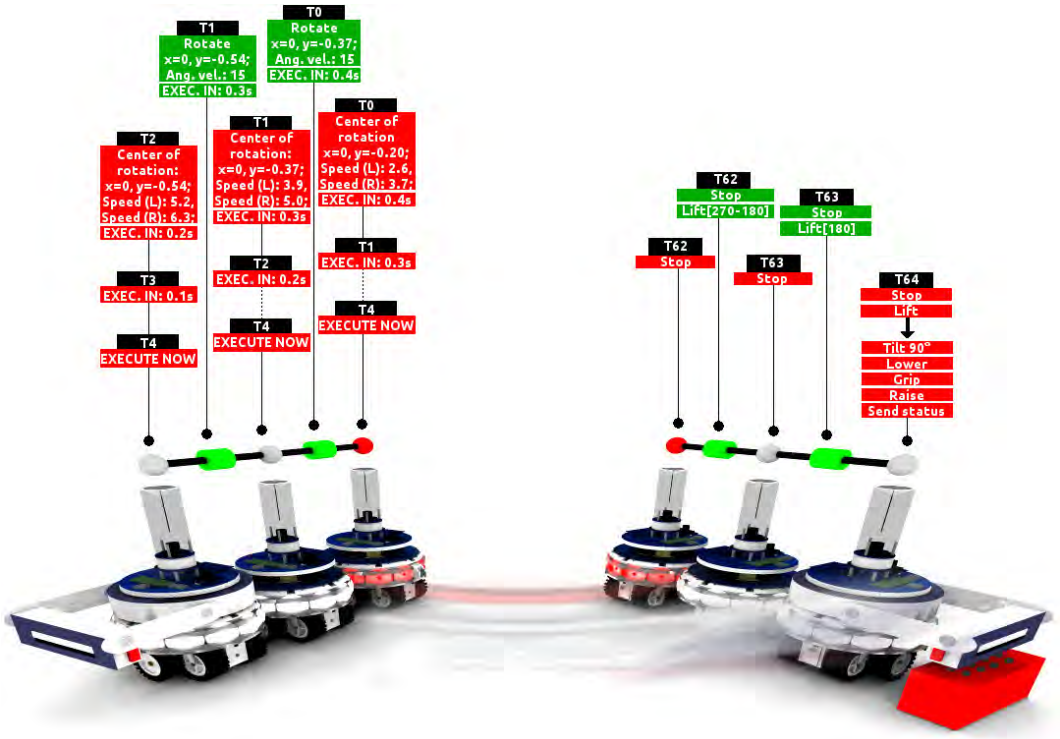


Figure 3.9: Spatial and temporal coordination in an MNS robot. A 3-unit MNS robot composed of two foot-bots and one manipulator foot-bot is shown to execute a rotational motion from left to right. The motion is coordinated by the brain unit such that the manipulator foot-bot can be used to lift a brick. Black rectangles show timesteps, red are commands planned by individual robotic units, and green are high-level commands propagated through the merged nervous system.

rection). Both aspects depend on the number of constituent units in the robot their connection topology (i.e., branching factor). In the following, we address both aspects separately.

If all sensory data (such as unprocessed camera feeds) were collected in the brain unit, scalability related to both bandwidth (transmitting sensory data from many robots) and computation (processing sensory data from many robots) may become an issue. Instead, in an MNS robot, the processing of sensor data is handled in a distributed manner. That is, each individual unit processes the data returned by its own sensors and sends only an abstraction of the date to its parent. However, as described in Section 3.3 and Figure 3.8, each unit is also responsible for the fusion of sensor data returned by all its children before they are propagated allowing the MNS control to scale gracefully and with respect to body size. The computational effort and the bandwidth (transmitting sensory data from many units to the brain unit) required by each robotic unit including the brain unit, therefore, depends only on the number of immediate descendants rather than the total number of units in an MNS robot's body. Sensor data extraction and processing thus becomes a constant cost, proportional to the number of children a single unit can have. In the case of the foot-bots and their circular chassis, this corresponds to a maximum of seven child units.

Short message propagation times are important in an MNS robot so that they maintain high reactivity to changing environmental conditions. For instance, a

robotic unit sensing a change must propagate an abstracted message to the brain unit, which in turn must make a decision based on this message and propagate a high-level command through the body as a response. This reaction time depends both on the number of constituent units k and the shape (i.e., the connection topology) of the MNS robot. The worst case scenario for propagation times are chain-like morphologies and is given by $2lp \times \tau$, where lp is the length of the path (in robotic units) from the brain unit to the final unit in the morphology. The communication delay τ between two adjacent units may depend on the underlying communication hardware – in case of the foot-bots, it corresponds to $\tau = 100$ ms required to exchange and process a message between to adjacent units. In the 4-unit MNS robot shown in Figure 3.8, the longest path $lp = 2$ and the reaction time is thus 2×2 unit $\times$ 100 ms unit^{-1}= 400 ms. Similarly, while the 6-unit MNS robot shown in Figure 3.7a and Figure 3.7b has $lp = 3$ units and therefore a reaction time of 600 ms, the 10-unit MNS robot in Figure 3.7c has $lp = 5$ units and thus a worst case reaction time of one second. In general, the reaction time of an MNS robot of size k lies between the upper and lower bounds defined by two extreme morphologies: the upper-bound is defined by a chain-like morphology of which the reaction time is $2k \times \tau$ while the lower-bound is defined by a hexagonal lattice pattern with the brain unit in the center – the most compact morphology possible for the self-assembling robots with circular chassis considered in this thesis. The longest path in such a morphology is approximately $lp = log_2 k$ units resulting in a best case reaction time of $2log_2 k \times \tau$.

3.3.2 Unprecedented features and self-healing properties

Self-assembled robot structures lack the equivalent of a nervous system that spans the whole body – similar to the communication bus that connects the submodules to the main controller in a monolithic robot. Instead, connected units in existing composite robots remain individually autonomous and rely on distributed approaches for coordination. Sensorimotor coordination in such robots is thus limited or absent which prevents them from solving tasks with the precision and reactivity provided by monolithic robots.

MNS robots constitute a new class of robots because they combine the modularity and morphological advantages of self-assembling robots with the level of sensorimotor coordination that remained limited to monolithic robots. Furthermore, the MNS approach unlocks unprecedented features and provides self-healing properties to self-assembling robot systems. Here, we present these features and demonstrate them using a series of experiments we carried out using the marXbot platform. Videos of the experiments can be found online at `https://www.nature.com/articles/s41467-017-00109-2#supplementary-information`.

3.3.2.1 Borrowing hardware capabilities of peer robots

A key feature of the MNS approach is the ability to enable a robot to tap into the signaling architecture of a peer robot and use it instantaneously and seamlessly. Once physically connected, the sensors and actuators mounted on the peer robot become *visible* to the robot that invited the connection. This is a direct consequence of the recruitment algorithm (see Algorithm 1) during which hardware configurations of neighboring robots are retrieved and then stored by each constituent unit as part of the recursive representation of all connected descendants (see Section 3.2.2).

In Figure 3.10, we present the results of an experiment we considered to study this novel feature enabled by the MNS approach. We consider a task in which an MNS robot is required to borrow physical (i.e, object lifting) capabilities exclusively available to peer robot in order to solve a object transport task successfully. We deploy two foot-bots and a manipulator foot-bot on a mission to find and retrieve one of the two bricks placed in the environment and it at a pre-defined location. As shown in Figure 3.10a, a foot-bot detects a brick suitable for transport by analyzing its omni-directional camera feed. However, the foot-bot is physically not able to neither lift nor transport the brick. The foot-bot opens an extension point and recruits the manipulator foot-bot (after retrieving the hardware configuration from both free robots) to extend its morphology and acquire new capabilities. Once the connection is formed – see Figure 3.10b – the foot-bot recognizes the new hardware that now became visible to its controller and issues the according "lift" command to the manipulator foot-bot. The new composite MNS robot is able lift the brick and is subsequently able to transport the brick to the required location, as shown in Figure 3.10c.

3.3.2.2 Autonomous adaptation to varying scales and morphologies

Mergeable nervous systems are able to autonomously adapt to all physical changes occurring to the connection topology regardless of shape and size of the MNS robot. We study this feature by providing a foot-bot with four different morphologies of varying shapes and sizes and then requiring the robot to assemble and disassemble all of them in a sequence while maintaining accurate internal representations of the morphology at all times.

Figure 3.11 shows an illustration composed of snapshots from the experiment and schematics of the internal representations retrieved from the foot-bot at the end of each morphology formation. In the center of the figure, we show the initial deployment of thirteen foot-bots from which one is pre-loaded with a SWARMOPRH-script (presented in Section 3.2.1) describing four morphologies. These morphologies vary in size (10, 7, 10, and 11 respectively) and shape (with a branching factor of 1, 2, 4, and 4 respectively). Through a series of merging and splitting of independent robotic units, the foot-bot is shown to adapt its body and nervous system (i.e., its signaling architecture) 68 times during task execution that we illustrate using 8 steps. In step 1, the foot-bot with the pre-loaded SWARMOPRH-script initiates the formation of a larger spiral-shaped MNS robot (upper left corner). In step 2, the MNS robot disassembles into independent units in a controlled, sequential fashion demonstrating the self-knowledge of the brain unit at all times and each of its robotic units becomes a one-unit MNS robot. The process of autonomous self-assembly and disassembly is repeated three times (steps 3-8) during which the MNS robots merge into three further MNS robots with different shapes ans sizes. The overall experiment lasted shortly under 16 minutes.

3.3.2.3 Morphology-independent sensorimotor coordination

The principle problem the MNS approach solves is the sensorimotor coordination problem in a composite robot composed of individually autonomous (in terms of communication, coordination, locomotion, and decision-making) units. To the best of our knowledge, the MNS robots presented in this thesis are the first self-assembling multirobot system able to solve tasks with the precision and reactivity equivalent to

that observed in monolithic robots.

Figure 3.12 shows snapshots of an experiment we carried out to demonstrate the novel coordination capability that become available to MNS robots. In the experiment, we manually design the behavioral rules for ten robotic units so that they form a series of MNS robots of different shapes and sizes. The different MNS robots all display consistent sensorimotor reaction to a moving stimulus. This reaction involves pointing at the stimulus using LEDs, and retreating from the stimulus if it is sufficiently close (i.e., proximity to any part of the MNS robot's body exceeds a threshold). When a composite MNS robot points to the stimulus, only the LEDs closest to the stimulus illuminate, independently of the robotic unit to which those LEDs belong. When moving away from the stimulus, movements of all wheel actuators on all constituent robotic units are coordinated through the robot nervous system of the MNS robot, allowing smooth motion of the composite body. In Figure 3.12a, ten independent robots are shown. We manually control the motion of the stimulus which provokes a reaction in three of the robots. In Figure 3.12b, two larger, composite MNS robots have been autonomously formed by the ten MNS robots. The two MNS robots are independent robots in their own right – they each have a single brain unit and nervous system. As part of the pointing behavior, the brain units of both robots switch on the three closest LEDs of their respective bodies green. In Figure 3.12c, the stimulus has exceed the proximity threshold to both robots causing both MNS robots to retreat from it. In Figure 3.12d, both MNS robots again merge autonomously to form a larger MNS robot consisting of all ten units. The brain unit of the 6-unit MNS robot has ceded authority to the brain unit of the 4-unit MNS robot. In Figure 3.12e, the brain unit of this newly formed 10-unit MNS robot demonstrates a previously unseen level of precision to composite robot by pointing at the stimulus using integrated sensory data that result from multiple, independent robotic. In Figure 3.12f, the MNS robot coordinates a retreat behavior away from the stimulus has moved too close.

3.3.2.4 Fault-detection and self-healing properties

The probability of total failure of robotic units increases together with the size of any composite robot. In general, the more independent software or hardware components become part of a single robot, the more susceptible the robot becomes to malfunctioning units. As opposed to partial failures described in Section 3.2.2, total failures may not allow a robot to function at all and may put the successful outcome of an ongoing mission at risk. To maintain robustness and autonomy, it is therefore important for an MNS robot to detect such faults, and then, if necessary, to self-heal in response.

We use the heartbeat mechanism integrated into the MNS control logic (see Section 3.3) to detect a failing robotic unit. That is, the absence of a heartbeat pulse from a parent unit tells a child that its parent is faulty, while the absence of an acknowledgement from a child tells the parent that its child is faulty. The heartbeat payload describes what actions should be taken as a response in the event of a detected failure. The simplest response that can be pre-defined by the brain unit is to require each unit to disconnect and retract while informing all its immediate neighbors to do the same. This mechanism may not represent an optimal solution for self-healing in terms of time and energy as it essentially disassembles the the morphology completely – as shown in Figure 3.11 – before the brain unit (or any other unit) can invite the disassembled units to reform an entirely new MNS robot

Figure 3.10: MNS robot borrowing physical capabilities from a peer robot. (a) Two foot-bots and a manipulator foot-bot are deployed on an object transport task. (b) A foot-bot has detected a suitable brick to be transported and extends its morphology to acquire new physical capabilities available exclusively to the manipulator foot-bot. (c) The new composite MNS robot is able to successfully transport the brick.

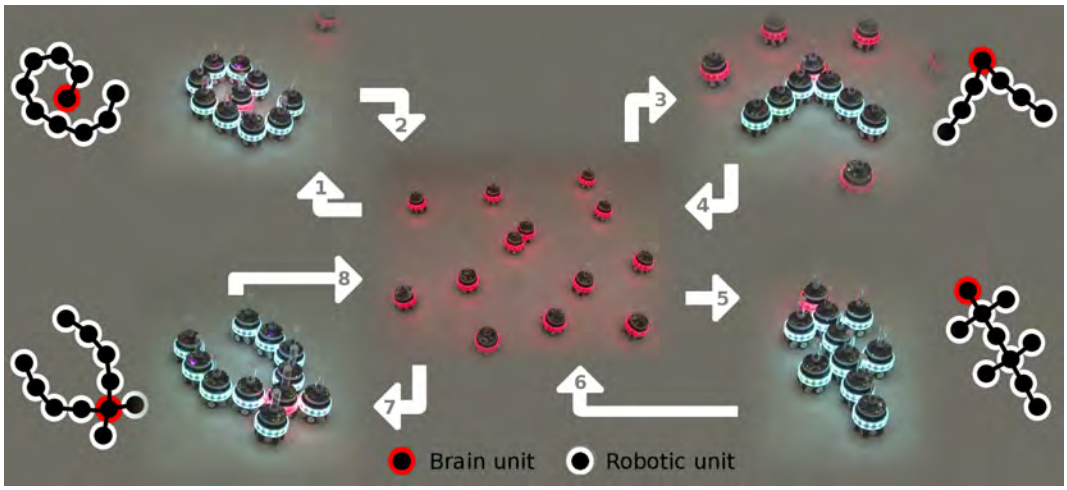


Figure 3.11: Autonomous MNS adaptation. Snapshots from an experiment in which independent robotic units autonomously form four MNS robots of different shapes and sizes (visualized using 8 steps). The schematics show merged nervous systems after the formation of each pre-defined MNS robot. The foot-bot with the pre-loaded SWARMORPH-script (the brain unit in all four morphologies) is shown in red.

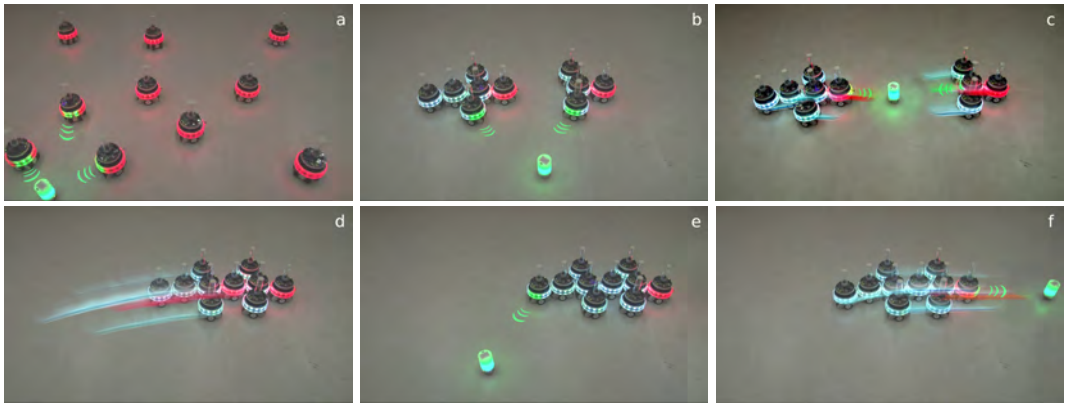


Figure 3.12: Morphology-independent sensorimotor coordination. Independently of shape and size, ten MNS robots display consistent sensorimotor reactions to a moving stimulus. For clarity, concentric green lines were added as overlays to highlight the pointing direction. See text for details on the experiment.

from scratch. Instead, as a partial disassembly may be sufficient in many cases, here we describe an approach for self-healing that excises a failed unit – even a brain unit – from the morphology and substitutes it with spare units when available.

Figure 3.13 shows a series snapshots from an experiment highlighting recovery of a 9-unit MNS robot from a faulty robotic unit. To simulate a failure, we manually disable all communication on one of the robotic units in the MNS robot. The missing acknowledgements to heartbeat pulses immediately allows its parent unit to detect the failing child. However, the physical connection created by the child to its parent unit cannot be broken by any other robot other than the one that created it. Hence, the parent unit needs to sacrifice itself by detaching itself from its own parent (faulty unit's grandparent) in order for the faulty unit to be excised from the MNS robot. Conversely, the faulty unit's potential children may also have detected the fault due to missing heartbeat pulses and detached themselves. In this way, the faulty unit, its parent or children no longer have any physical connection to the other units of the MNS robot that they were previously part of. Note that the need to sacrifice the parent robot is dictated by the current implementation of the marXbot platform, rather than by any intrinsic property of the MNS control logic. Currently, the only way to separate two connected marXbots is for the gripping robot to detach itself. The gripped robot is passive, and has no active way to release itself. A self-assembly platform that supports an active detachment mechanism by the gripped robot would obviate the need to sacrifice the parent robot when a child robot experiences a fault. The self-healing behavior of the MNS robot that has successfully excised the fault unit (and its parent unit) then maneuvers the robot away from the immobilized robotic units and recruits free robots to recreate the original morphology. In this manner, the original morphology can be maintained free of faulty units as long as there are spare/free units available.

Depending on the mission, specific reconfiguration sequences can be developed for specific morphologies and specific failures. For instance, children of a faulty unit can be required (by setting appropriate heartbeat payload) to detach only from the faulty unit and remain as a sub-morphology (instead of disassembling into individual units) that then can be immediately recruited to fill an extension point on the original MNS robot. MNS robots use such a reconfiguration logic to self-heal if its brain unit experiences a failure. To demonstrate the self-healing capability from the loss of a brain unit, we design an experiment with behavioral rules for eight robotic units so that they self-assemble into an MNS robot with an Y-shape. We then simulate a fault in the brain unit by disabling all its functionalities. As shown in Figure 3.14 using a series of snapshots, three child units react to a fault in the brain unit by detaching and creating three new independent MNS robots of which one of them becomes the brain unit. These three independent robots then merge again to form another new robot with a morphology as close as possible to the original. In this experiment, there is a natural re-arrangement of the three-child segments that results in a similarly shaped body. An MNS robot can therefore recover even from a faulty brain unit by reconfiguring its body to excise the faulty component even in the absence of spare units.

The self-healing properties described in this section enables MNS robots to display high levels of fault tolerance and robustness by combining splitting and merging capabilities once a fault has been detected in an unit. As opposed to, for example, learning new behaviors to cope with faults [46, 50], we chose an approach observed also in nature [51] to enable an MNS robot to self-heal by excising faulty body parts.

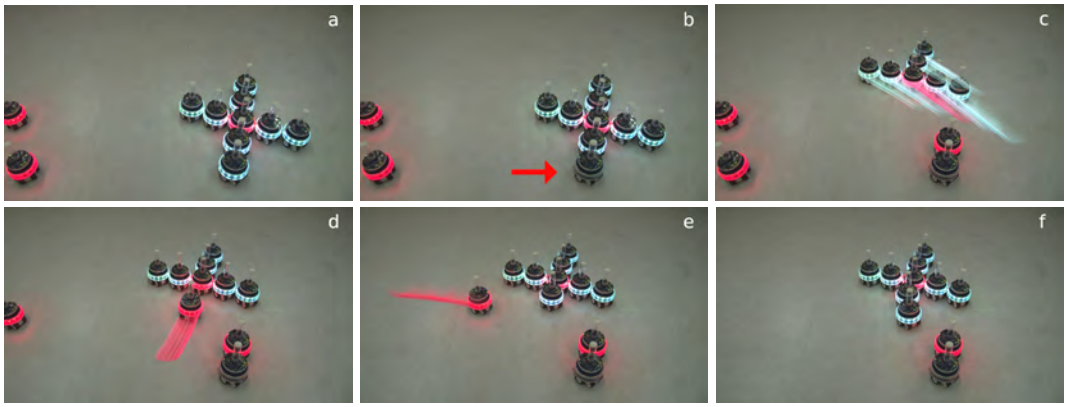


Figure 3.13: An MNS robot executing a self-healing behavior to recover from a failing robotic unit. (a) We manually program the formation of a 9-unit MNS robot and (b) disable all communication of the robotic unit marked with the arrow. (c) The parent of the faulty robotic unit detects the fault and responds by sacrificing itself. The healthy part of the MNS robot executes the pre-programmed self-repair behavior and moves to a new location. (d,e,f) Subsequently, free robots are invited to join the morphology to recreate the original robot that had experienced a fault.

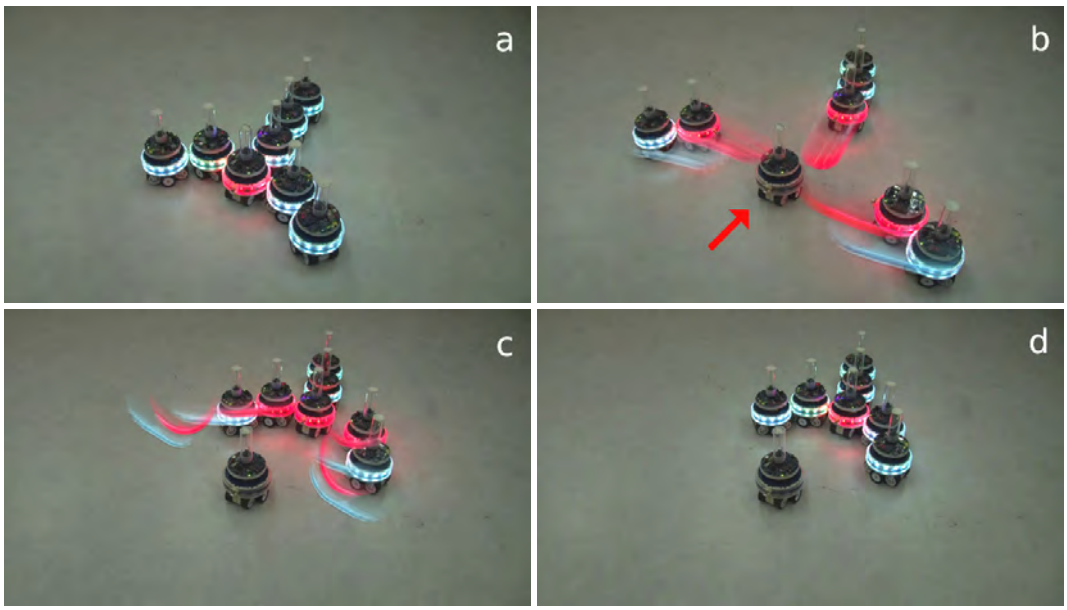


Figure 3.14: An MNS robot self-heals after its brain unit was induced with a fault. (a) We pre-program the formation of an Y-shaped 8-unit MNS robot. (b) A failure is induced in the brain unit (marked with the arrow). Using the heartbeat mechanism, the three child units of the brain unit detect the fault and respond by detaching and forming three new MNS robots each with a brain unit if its own. (c, d) The three new MNS robots merge to recreate a single composite MNS robot with a morphology close to the original.

3.4 Related work

We review self-assembling and other modular robot systems that have been introduced in the past decade. For systems that existed prior to the year of 2007, we invite the reader to refer to the extended review presented in [52].

Although no physical connections are considered, in many ways, the breakthrough work presented in [5] represents the state of the art in self-assembling robot research. For the first time, a system was presented in which more than thousand individual units were shown to cooperate and form pre-defined morphologies. The scale of cooperation demonstrated exceeds current standards in multirobot research [53] by an order of magnitude. The ability to design, manufacture and control thousand robots represents multiple orders of magnitude improvement over today's existing – i.e., not conceptual or simulated – self-assembling robot systems that consider 10 to 20 cooperating robots at most. Based on three basic behaviors (edge-following, gradient formation, and localization) and four pre-selected seed robots defining the origin and orientation of a coordinate system, individual units form morphologies in experiments that can last several hours. However, the work presented in [5] focuses exclusively on morphology formation and does not propose coordination or control mechanisms for already assembled robots.

A robot system able to coordinate motion in a self-assembled robot is Sambot [54]. Individual units are shown to detect and estimate the height of a wall placed in the environment. As a response, they assemble a snake-like or multi-legged robotic structure and coordinate their motion over the wall by generating an appropriate gait or leg movements using a central pattern generator model. The system relies on an obstacle (wall height) to morphology mapping made available to a pre-selected seed robot to initiate shape formation. A structural extension to the widely known mobile robot platform e-puck is presented in [55]. The extension allows mobile robots to form physical connections to four other robots through passive magnetic docking interfaces. In this work, the shape of target morphology cannot be pre-programmed or determined as a function of task. Instead, it rather emerges from the interaction between extension design and the environment. A controller is also presented that enables a connected 4-unit robot to coordinate motion while avoiding obstacles placed in the environment.

An entirely novel approach to overcome terrestrial obstacles has been presented by two separate research groups [56, 57]. Individual robots in both systems are hexagonally shaped and are capable of autonomously moving on the ground and dock with each other to take flight in a coordinated fashion while still being controlled in a distributed fashion. In [56], target structures are pre-defined and self-assembly is initiated by a randomly chosen seed robot. Hence, the system does not have control over where a target structure will be formed. In [57], target structures are entirely random and cannot be chosen as a function of task or environment. Certain (if not most) parameters of the self-assembly process are defined by an environmental cue: an overhead light source is used for pose-estimation and navigation towards assembly location. Nevertheless, collective robot structures generated by both systems are highly interesting because of their unique ability for vertical takeoff and stationary flight that can be useful in acquiring a different perspective of the ground.

In the previous decade, the community has not only witnessed the development of airborne self-assembling robots, but also those capable of operating in liquid environments [58, 59]. Three different types of independent units are introduced

in [58]: two capable of operating on the surface of water and one is designed to operate when immersed in water. For one particular robot type, an interesting energy supply mechanism is presented. Instead on depending on batteries for the energy required for propulsion, it draws current from from a metal ceiling (or a pantograph). In [59], robots in an experimental system were shown to form two predefined target structures in a fluidic assembly system. A computationally efficient simulator is also presented to model the stochastic fluidic assembly process. In a separate paper [60], the authors also present an automated assembly-planning algorithm without prior knowledge of the times or locations of component availability. In both systems presented [58, 59], shape formation remains the primary research focus while sensorimotor coordination for composite robots is entirely absent.

Evo-bots [61] demonstrate previously unseen forms of coordination among self-assembling robots in by being able to search for, harvest (using solar panels) and exchange energy. The square shaped modules in this work have a connector on each side and do not have motion capability. They therefore depend on their environment (an air-hockey table) to provide them with the energy required for them to undergo semi-random motion. Connections occur stochastically on collision and can be actively reversed by modules after exchanging information between each other. In its current state, the system is only able to produce chain-like, linear, structures. One important feature of this system is its ability to build replicas of existing structures using free modules in the environment. Another novel approach to build robots using existing modules is presented in [62]. In real-time, composite robots are autonomously constructed using cube-shaped modules and their shapes are incrementally improved without simulation or human intervention by a "mother robot" – a robotic manipulator with 6 degrees of freedom. In this model-free approach, candidate morphology fitness is evaluated using artificial evolution techniques based on its ability to move in a pre-defined direction on the ground. However, composite robots constructed through this time-intensive process depend on uncoordinated behaviors that result in rather rudimentary motion and unsteady trajectories.

Most impactful applications of engineered self-assembly may possibly result from research carried out at the smallest scales [63]. Modular robot systems are being designed to work at centi-, milli- and nanoscales and exploit the forces (such as capillary, thermodynamic, chemical, gravitational, magnetic etc.) they encounter in the environment to be functional. For instance, initial prototypes of centimeter-sized swallowable modules that can self-assemble into medical devices inside the human stomach and then move through the intestine have already been presented [64]. In the future, such modules have the potential to facilitate minimally invasive medical procedures including diagnosis and complex surgical interventions. In [65], researchers have proposed millimeter-sized robots that exploit the magnetic energy of magnetic resonance imaging scanners for tissue penetration. They can navigate through fluid-filled passageways of the human body and reach a target location where they self-assemble to convert the magnetic energy into kinetic energy required to achieve penetration. These robots have the potential to provide minimally invasive and highly localized therapies with minimal trauma. At the nanoscale, particles have been designed to form self-assembled aggregates inside living cells as a response to changing environment acidity levels [66]. Such aggregates shift light absorption to far-red and near-infrared enabling efficient photothermal cancer therapy. Self-assembly has also been applied to nanoparticles [67] to improve the delivery of diagnostic (i.e., imaging or contrast) agents into tumor cells. Although

today's self-assembling systems capable of operating inside the human body are mainly conceptual, in the near future, they are expected to revolutionize diagnostic and therapeutic procedures entirely. Please refer to this review [68] for applications of self-assembling nanoparticles outside of the human body.

3.5 Summary

The work presented in this chapter allows for the first time self-assembling robots to demonstrate a level of sensorimotor coordination that previously remained limited to monolithic robots only. We presented EDSA – enhanced directional self-assembly – a fast and precise connection forming mechanism that relies on active recruitment and guidance. We then showed how EDSA can be combined with previously existing mechanisms to form topology-aware morphologies in a controlled manner. Based on these results, we introduced mergeable nervous systems – a novel control concept for the control of self-assembled robotic structures. We conducted experiments to show that mergeable nervous systems enable self-assembling robots demonstrate previously unseen features. However, the work presented so far has the following limitations:

- The seed robot is pre-determined by a human operator in all experiments
- The shape and size of morphologies are pre-defined prior to deployment
- Formed morphologies are not task-dependent – they serve no purpose.

We address these limitations in the following chapters. To this end, in Chapter 4, we present a protocol that allows a robot in a multirobot system (for instance an aerial robot with a better perspective of the environment) to select an interestingly located peer robot to trigger self-assembly. The selection is based on the peers location and results in a dedicated communication link established to a specific robot in the communication range. An extension of the protocol is also presented such that further nearby robots can be added to the link. The number of selected robots defines the size of the morphology to be formed. In Chapter 5, we then show how aerial robots can use such links to control the size and shape of task-specific morphologies on the ground.

CHAPTER 4

Establishing spatially targeted communication links

In this chapter, we study spatially targeted communication in decentralized multi-robot systems. We first detail a protocol that can be used to establish a one-to-one spatially targeted communication (STC) link between an initiator and a target robot (see Figure 4.1 for a graphical explanation of the terminology used in this chapter). Then, we present another protocol that allows existing one-to-one STC links to be expanded to one-to-many STC links between an initiator and a group of co-located robots, the target group. Both protocols have been implemented and thoroughly tested on multiple robot hardware. We present the results of experiments we carried out using a homogeneous system (i.e., a system solely composed of ground-based robots) and a heterogeneous system (i.e., a system composed of both aerial and ground-based robots) together with a Markov chain model used to describe and study the underlying process formally.

The proposed protocols have a number of advantages: (i) They can be implemented using standard, low-cost components that can be mounted on most existing robot platforms. In our experimentation, we used the camera and LEDs-based communication for the experiments conducted using the homogeneous system composed of multiple s-bots. In this form of communication, images returned by the cameras are processed to detect RGB color blobs (where each color corresponds to a different message transmitted by a neighboring robot using its LEDs) of which situational information such as the relative distance and the angle to each robot can be estimated. For the heterogeneous system composed of an AR.Drone and multiple foot-bots, we combined camera and LEDs-based communication with standard wireless Ethernet. (ii) The protocols can be accurately modeled using Markov chains allowing us to study the scalability of the protocols from a theoretical perspective. (iii) The results of the model-based analysis and real robot experimentation show that both protocols are highly scalable.

This chapter is organized as follows. In Section 4.1, we present a protocol to establish one-to-one STC links followed by a Markov chain-based model and real robot experiments that we used to analyze the properties of the protocol. In Section 4.2, we present an extension of the protocol that shows how an existing one-to-one STC link can be expanded to a one-to-many STC link. We also present another Markov chain-based model that describes this protocol extension and its constituent parts. In Section 4.3, we discuss potential shortcomings of the work presented in this chapter and present possible solutions. In Section 4.4, we discuss the state of the art in multirobot communication that provide spatial coordination. Finally, in Section 4.5, we summarize the contributions of this chapter.

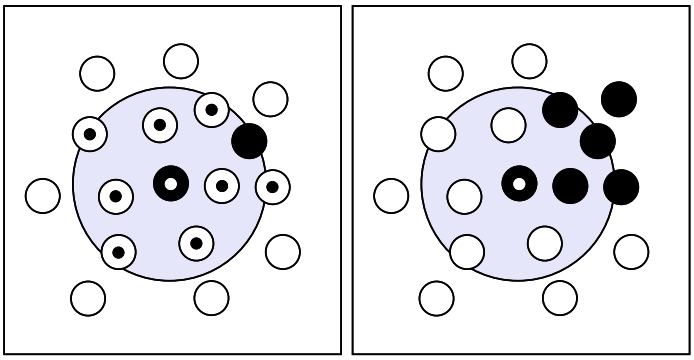


Figure 4.1: Explanation of the terminology used in this chapter. Left: the initiator robot ◉ and its signal transmission range that include the target robot ● and potential recipient robots ⊙. Right: the initiator robot and a target group of size five.

4.1 Establishing a one-to-one communication link

Given a set $C := \{c_1, \ldots, c_s : s \geq 3\}$ of distinctive signals available to all robots, we use an iterative elimination process to establish a one-to-one STC link between an initiator and a target robot. We use the designated signal c_1 initialize and terminate the elimination process. The subset $C_s := \{c_2, \ldots, c_s\}$ is reserved for the iterative component of the elimination process. In this thesis, RGB colors set using LEDs or strings sent over wireless Ethernet are used to define the sets as $C := \{red, blue, green\}$ and $C_s := \{blue, green\}$. The initiator robot indicates its intent to establish a one-to-one STC link to a preselected target robot by initializing the elimination process by emitting c_1. Robots within the reception area of the c_1 signal, that is the potential recipient robots, join the elimination process by acknowledging with c_2. As soon as the initiator robot emits a matching handshake signal c_2, both parties enter the iterative component of the elimination process. At each iteration, potential recipient robots still in the process randomly choose and emit a signal from the set C_s. The initiator robot matches the signal emitted by the target robot. At the end of the iteration, only those potential recipient robots whose signal match that of the initiator robot – and thereby that of the target robot – remain in the process. The robots whose signals were not matched quit the elimination process and stop emitting signals. This elimination process continues until the target robot is the only robot remaining in the elimination process. At this point, the initiator robot indicates the termination of the elimination process by emitting the dedicated signal c_1. The target robot acknowledges the termination by also signaling c_1. The initiator robot has now established a one-to-one STC link to the target robot.

In what follows, we assume the initiator robot has already selected a target robot based on task parameters such as the target robot's relative position to the initiator robot or to other objects of interest. We first describe the protocol at a microscopic (i.e., at a robot controller) level using finite state machines (FSMs). Based on this microscopic description, we then derive a macroscopic (i.e., a system level) model using a time-homogeneous Markov chain. We also present results of the protocol scalability analysis we conducted using data collected from physics-based simulations, real-robots experiments, and predictions of the macroscopic model.

4.1.1 The iterative elimination process and preliminary trends

Figure 4.2a shows the FSM executed by the initiator robot. It consists of three states: `STA` (start); `EP` (elimination process); and `LE` (link established). When the initiator

robot needs to establish an STC link with a particular robot, it enters the `STA` state and transmits the predefined signal red (c_1) to initialize the iterative elimination process. The transition `ta1` is triggered and state `EP` is reached if at least the target robot acknowledges the `STA` signal. This expedient guarantees that the target robot is not busy with an ongoing task and is available to join the elimination process and to be selected for communication. In state `EP`, the initiator robot continuously matches the color displayed by the target robot by taking transition `ta2`. However, if all other robots have been eliminated and the target robot is the only robot displaying a color, the transition `ta3` is taken to enter the state `LE`. The predefined color red (c_1) is displayed to confirm an established STC link to the target robot.

The FSM executed by the potential recipient robots, including the target robot, is shown in Figure 4.2b. The FSM is composed of three states: `ACK` (acknowledge); `EP` (elimination process); `LE` (link established). The transition `tw1` is taken to enter the `ACK` state as soon as the color is red (c_1) is received from an initiator robot. If available for communication with the initiator robot, the predefined color blue (c_2) is displayed to acknowledge the participation in the elimination process. The transition `tw2` is triggered as soon as the initiator robot matches the color *blue*. In state `EP`, each potential recipient robot randomly selects and displays a color from the set C_s. Simultaneously, it starts incrementing an internal timer t. Whenever t exceeds a fixed threshold τ, the robot examines the latest color received from the initiator robot to determine whether to remain in the elimination process by taking `tw3`, or to terminate the behavior by taking `tw5` as a result of mismatching colors. The timer mechanism provides the initiator robot with sufficient time to perceive, process and react to the colors displayed by the potential recipient robots. When a potential recipient robot that has already entered the elimination process perceives the predefined color *red* (c_1) from the initiator robot, it can safely assume that it is indeed the target of the initiator robot. In this case, the transition `tw4` is triggered and the potential recipient robot confirms the termination of the elimination process by also displaying the color *red* (c_1).

We carried out preliminary tests in simulation to study the impact $|C_s|$ has on the number of iterations required for the termination of the elimination process. We also tested the scalability of our approach by varying the number of potential recipient robots from which the initiator robot chooses the target robot from. We used a heterogeneous system composed of an eye-bot and multiple foot-bots in which the eye-bot always assumed the role of the initiator robot and the foot-bots that of the potential recipient robots. We developed two autonomous robot controllers: one for the eye-bots and one for the foot-bots. The controllers are completely distributed and homogeneous, i.e., all foot-bots execute the same controller. Both controllers are behavior-based and are based on the FSMs shown in Figure 4.2. As shown in Figure 4.3a, each simulation run started with an eye-bot placed at a height of 2 m in the center of an obstacle-free arena of dimensions 2 m x 2 m. The foot-bots are randomly placed within the visual range of this eye-bot and is able to perceive all foot-bots within the arena and vice versa. Initially, the foot-bots perform a random walk while avoiding other robots. Their green LEDs are switched on so that they remain visible to the eye-bot. Within the initial two seconds, i.e., before the foot-bots start moving out of its vision range, the eye-bot randomly picks a target foot-bot and starts the one-to-one STC link establishment process. All foot-bots respond to the eye-bot and become static.

Figure 4.3b shows the results of our preliminary tests. The number of iterations

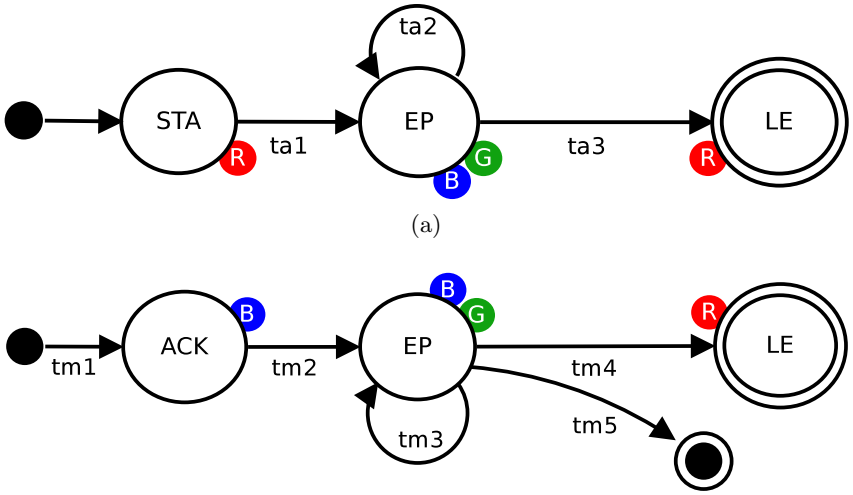


Figure 4.2: Finite state machines running on (a) the initiator robot, and (b) the potential recipient robots that enable the establishment of a one-to-one STC link. The color shades represent the RGB color signals available in the respective states. STA: start, ACK: acknowledge, EP: elimination process, LE: link established.

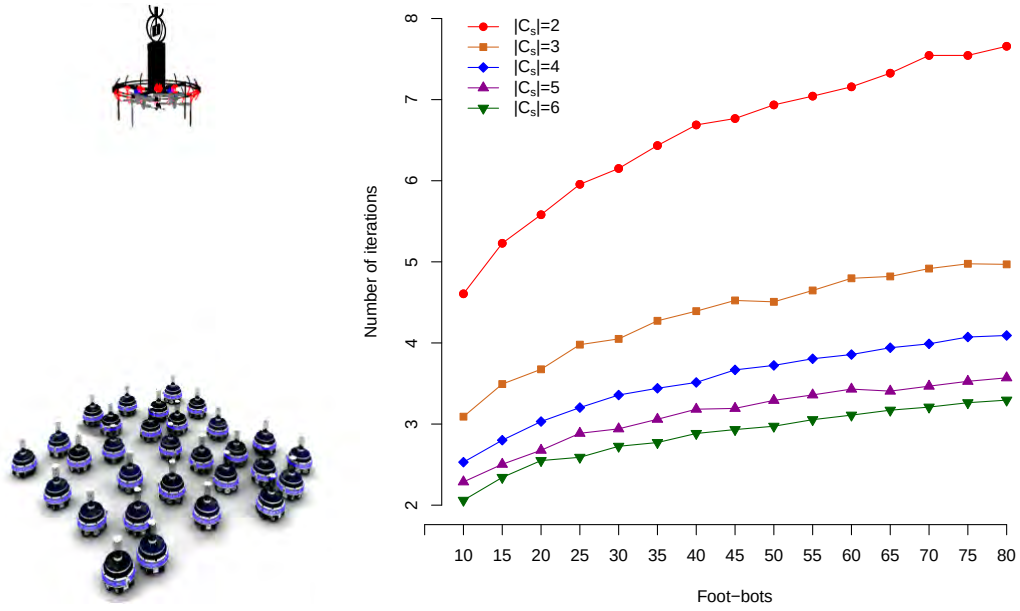


Figure 4.3: Preliminary experiments of the iterative elimination process in simulation. (a) A screenshot showing one eye-bot (the initiator robot) and thirty foot-bots. (b) Results of the experiments with varying number of foot-bots and $|C_s|$, i.e., number of signals available to the iterative elimination process. Each data point is the average of 1000 simulation runs.

required to terminate the elimination process is plotted against the number of foot-bots which we varied between 10 to 80 in steps of 5. Each curve in the graph represents a value between 2 to 6 chosen for $|C_s|$, that is the number of RGB colors available to the iterative component of the elimination process. We added new colors to C_s to match the value considered. For instance, the redefined set for $|C_s| = 6$ was $C_s := \{blue, green, orange, yellow, magenta, cyan\}$. Each data point on each curve is the average value resulting from 1000 simulation runs. The results indicate three tendencies. First, the number of iterations required for the eye-bot to establish an STC link only experience a logarithmic growth as the number of foot-bots increases. This observation is confirmed by the analysis carried out in Section 4.1.2. Second, increasing the number of signals available to the iterative elimination process, that is to $|C_s|$, have a significant impact on the speed of the elimination process by requiring less iterations until termination. For instance, in the case of 2 colors and 20 foot-bots, the average number of iterations is 5.5 while in the case of 6 colors and 20 foot-bots, the average is 2.5 iterations. This is an immediate consequence of the fact that the higher $|C_s|$ is, the more robots get eliminated from each iteration on average. Third, as separately listed in Table 4.1, an interesting trend is apparent when considering the standard deviations: the higher the value of $|C_s|$, the lower is the standard deviation. Hence, using more signals in the elimination process does not only reduce the number of iterations required, but it also make the number of iterations required more predictable. Also this observation is theoretically analysed and confirmed in the following Section 4.1.2.

Table 4.1: Standard deviations for varying values of $|C_s|$ and different numbers of foot-bots. Simulation results from 1000 repetitions for each value of $|C_s|$.

$\|C_s\|$	10	20	30	40	50	60	70	80
2	1.82	1.90	1.80	1.88	1.84	1.78	1.96	1.91
3	1.16	1.17	1.18	1.24	1.18	1.16	1.20	1.21
4	0.94	0.90	0.95	0.92	0.98	0.93	0.99	0.97
5	0.85	0.81	0.84	0.90	0.81	0.90	0.85	0.84
6	0.74	0.79	0.77	0.80	0.76	0.77	0.79	0.74

Note that the data presented so far result from empirical studies we carried out in simulation. In the following, we present a theoretical model that we use to confirm the trends observed in simulation.

4.1.2 Markov chain model and model-based analysis

For a given number of potential recipient robots N that are located in the communication range of the initiator robot, we investigate the number of iterations necessary for the iterative elimination process to terminate. That is, the time required for the initiator robot to establish a one-to-one STC link with a target robot. We consider time to be discrete due to (i) the behavioral nature of the robot controllers that are discrete and defined using FSMs [69], and (ii) the clock time spent by actual robots may depend heavily on the underlying robot platform and communication modalities. The presented model aims at abstracting away from such hardware specifications while focusing on the performance of the presented algorithm. In our model, we define time using the concept of iterations: each time step n corresponds to one iteration in the iterative elimination process. An iteration ends when all

remaining robots in the elimination process have (i) selected and transmitted a random signal from C_s and (ii) have assessed the match or mismatch of this signal with that received from the initiator robot.

In Section 4.1.1, we presented a *memory-less* process. That is, at each iteration, each potential recipient robot acts only on the basis of its current state and on the current signal received from the initiator robot. The future state of the robot therefore fulfills the Markov assumption [70] as it is independent of its past. Additionally, the threshold-based timer mechanism that is part of the FSM allows us to consider all potential recipient robots from a macroscopic point of view in the model. We use this synchronization property of the robots to abstract from individual robot states and consider subsets of robots that are in identical states simultaneously. As a result, we define a macroscopic model of the iterative elimination process using a time-homogeneous Markov chain. We use an absorbing Markov chain that features a set of transient states and a single absorbing state. The transient states represent the intermediate states of the iterative elimination process. The absorbing state guarantees the termination of the iterative elimination process in finite time.

Let $\mathbb{N}$ be the set of naturals. We define our process as a Markov chain $\{X_n, n \in \mathbb{N}\}$ with $N+1$ states. The state of the process is represented by the random variable $X = x \in \Omega := \{1, \ldots, N+1\}$. Each state is characterized by the number of robots $\eta_x = N + 1 - x$ that still remain in the elimination process. All states of the chain are transient states except for $X = N+1$ which is an absorbing state. That is, once the chain enters the state $X = N + 1$, it will remain in that state for an infinite amount of time. Accordingly, in state $X = N + 1$ the number of robots involved in the process is $\eta_x = 0$, which corresponds to the termination of the elimination process. At time step n, the process will move from the current state $X_n = i$ to the next state $X_{n+1} = j$ with a probability π_{ij}. We define the stochastic transition matrix $\Pi := (\pi_{ij} : i, j \in \Omega)$ of the Markov chain as

$$\Pi := \begin{cases} \pi_{ij} = C_{\eta_i - 1}^{\eta_j - 1} p^{\eta_j - 1} (1-p)^{\eta_i - \eta_j}, & \text{for } i < N, i \leqslant j \leqslant N \quad (4.1a) \\ \pi_{ij} = 1, & \text{for } i \geqslant N, j = N+1, \quad (4.1b) \\ \pi_{ij} = 0, & \text{otherwise.} \quad (4.1c) \end{cases}$$

The transition probability between transient states are defined in Case (4.1a) as the probability π_{ij} that $\eta_i - \eta_j$ robots will leave the elimination process when taking the transition $i \to j$. This probability follows a binomial distribution where parameter $\eta_i - 1$ is the number of robots that are remaining in the elimination process without the target robot and $p = 1/|C_s|$ is the probability for one of these robots to randomly select and emit the identical signal as the target robot. Case (4.1b) defines two probabilities: (i) the transition from the state $X_n = N$ to the absorbing state $X_{n+1} = N+1$ occurs with probability $\pi_{ij} = 1$. This deterministic step models the final handshake between the initiator and the target robot in the last iteration. (ii) the state $X = N+1$ is defined as an absorbing state. Finally, Case (4.1c) states that other transitions not covered by the previous two cases will never occur.

The iterative elimination process is formally described by the stochastic transition matrix Π. Such formal descriptions are often used to predict the performance of processes in conditions that go beyond what is possible using real world systems. Model predictions can be used to detect faults [71] in a system during runtime – for instance, when the iterative elimination process exceeds its termination time due to errors in signal transmission or malfunctioning potential recipients robots. Be-

fore the model can be used in such scenarios, it needs to be validated for accuracy. We validate the accuracy of our model by comparing model predictions with data acquired from physics-based simulations. After confirming it accuracy, we use the model to analyze the scalability of the iterative elimination process. To this end, we first define the random variable ϑ as the number of iterations required before process termination. Second, we consider the matrices Γ and Σ of the canonical decomposition of Π [70] to study the distribution of ϑ. Γ defines the transition probabilities between transient states. The fundamental matrix Σ, given by $(I - \Gamma)^{-1}$ where I represents the identity matrix, gives the expected number of visits to each transient state. We validate the accuracy of the model predictions by comparing the distribution of ϑ with the empirical data acquired from simulation. The cumulative distribution function $F(\vartheta) = P(\vartheta \leqslant n, x_0)$ can be obtained for a given initial state $X_0 = x_0$ as the infinite series

$$F(\vartheta) = 1 - \sum_{j \in \Omega} \Gamma^n_{x_0,j}. \tag{4.2}$$

The probability of the process to be in a transient state at iteration n is given by the subcomponent $\sum_{j \in \Omega} \Gamma^n_{x_0,j}$ of Equation 4.2. Followingly, the complement of this value provides the probability of entering the absorbing state prior to iteration n.

Figure 4.4a shows the cumulative distribution function $F(\vartheta)$ provided by the model plotted with the empirical distribution function $\hat{F}(\vartheta)$ obtained from 1000 simulation runs for three different values of $|C_s|$. The number of potential recipient robots N was set to 50. The theoretical predictions of the model are shown to closely match the the empirical observations for each value of $|C_s|$. Also the probability mass function $f(\vartheta) = P(\vartheta = n, x_0)$ shows a high agreement between the two data sets (see Figure 4.4b), indicating reliable model predictions. The probability mass function is right-skewed, i.e., the probability mass is concentrated under the left side of the curve. This implies the possibility of long executions of the process. However, with increasing $|C_s|$, i.e., the number of signals available to the elimination process, the variance of ϑ shrinks considerably reducing the probability mass under the right tail, and thus, the occurrence of long executions. We compute the expected value $\mathbb{E}[\vartheta]$ and the variance $\mathbb{V}[\vartheta]$ of the number of iterations necessary to establish a one-to-one STC link. from the matrices Γ and Σ according to

$$\mathbb{E}[\vartheta] = \xi\Sigma, \tag{4.3}$$

and

$$\mathbb{V}[\vartheta] = (2\Sigma - I)\mathbb{E}[\vartheta] - \mathbb{E}_{sq}[\vartheta]. \tag{4.4}$$

The term ξ in Equation (4.3) represents a column vector of all $1s$. That is, for every (transient) initial states, the expectation value of ϑ is given by the row sum of the fundamental matrix Σ. In Equation (4.4), I denotes the identity matrix and $\mathbb{E}_{sq}[\vartheta]$ is $\mathbb{E}[\vartheta]$ with squared entries.

We use Equations (4.3) and (4.4) to study how the distribution of the expected value ϑ scales for increasing values of N and $|C_s|$. As the results presented in Figure 4.4c show, $\mathbb{E}[\vartheta]$ is characterized by a logarithmic growth indicating high scalability for increasing values of N. The same trend applies to the variance of ϑ: given a certain value for $|C_s|$, $\mathbb{V}[\vartheta]$ grows logarithmically for increasing N. The results also show that as $|C_s|$ increases, the variance of ϑ decreases considerably increasing the reliability of the expectation value as an aggregated indicator of the

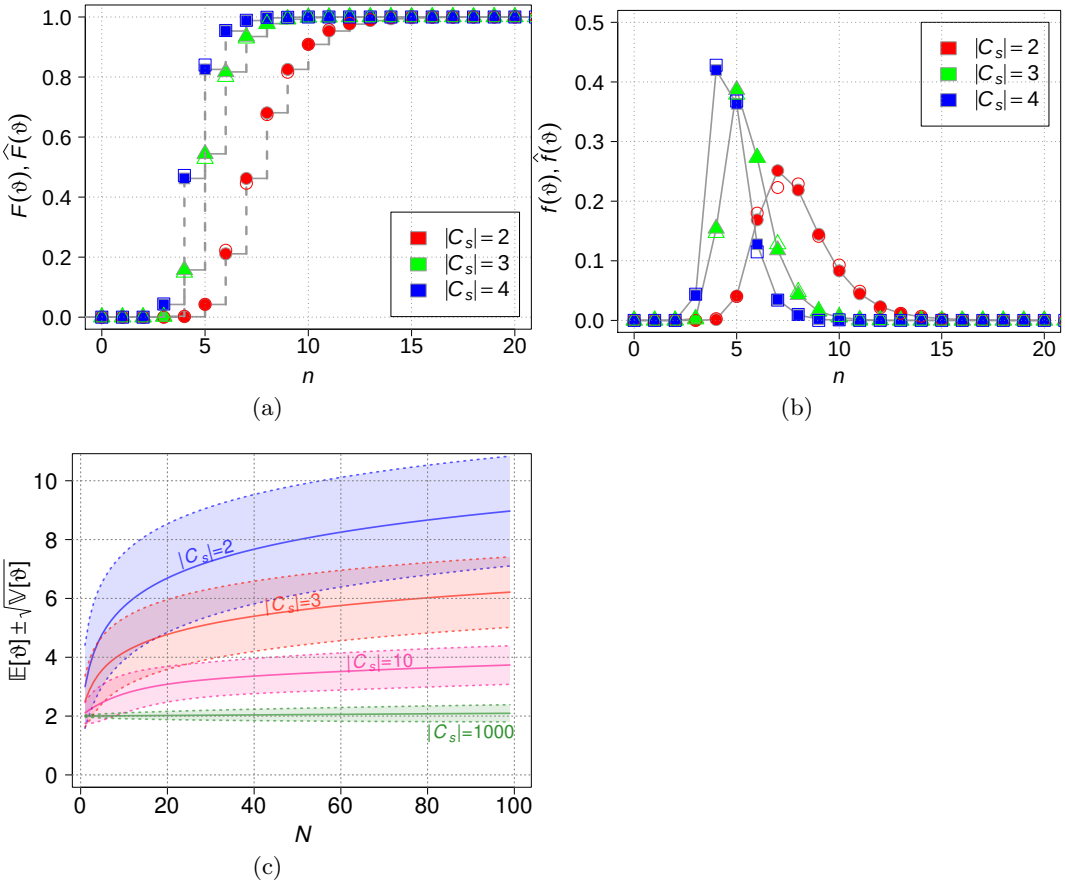


Figure 4.4: Model validation and scalability prediction. (a) Cumulative distribution function $F(\vartheta)$ and (b) probability mass function $f(\vartheta)$ and their empirical counterparts $\hat{F}(\vartheta)$ and $\hat{f}(\vartheta)$, respectively, plotted against the number of required iterations n before termination. Full symbols represent model predictions, empty symbols provide the outcome of 1000 physics-based simulation runs where the number of potential recipient robots N was set to 50. (c) Plot showing the scalability of the iterative elimination process as predicted by the model for large N and $|C_s|$. The shaded areas represent the variance of the expected value: $\mathbb{E}[\vartheta] \pm \sqrt{\mathbb{V}[\vartheta]}$.

performance of the process (e.g., compare the width of the shaded areas for $N = 80$ between $|C_s| = 2$ and $|C_s| = 1000$ in Figure 4.4c). Followingly, in the limit of an infinite number of available signals, $|C_s| \to \infty$, the probability for a potential recipient robot to randomly choose the same color as the target robot tends to be zero. The variance of ϑ disappears, and as a consequence, the iterative elimination process approaches a deterministic behavior that has exactly 2 iterations: at the first iteration all potential recipient robots except the target robot are eliminated, while in the second iteration the target robot performs the final handshake with the initiator robot. At the other end of the spectrum, in an application scenario where the number of available signals is limited, the elimination process still ends in finite time as guaranteed by the absorbing state of the Markov chain. The number of iterations necessary to complete the process decreases as the number of available signals increases.

The presented Markov chain model theoretically confirms all three tendencies

we identified using the empirical data collected from preliminary experimentation described in Section 4.1.1. The model suggests high, logarithmic scalability of the studied process. Also, an initiator robot can detect faults in the process by comparing actual number of iterations occurred in a process to its probability value returned by the model.

4.1.3 Experiments and results

We first ran a series of experiments on a homogeneous robot system composed of s-bots only. Figure 4.5 shows snapshots from an experiment using 5 s-bots. We let the s-bot in the bottom row assume the role of the initiator robot while keeping all robots stationary during the experiments. We implemented two behavior-based controllers based on the FSMs presented in Section 4.1.1, one for the initiator robot and another one for the remaining robots. Each s-bot is placed such that each RGB-based color signal emitted by an s-bot can be perceived by all other s-bots. The timer threshold τ is set to 20 control steps, which is equivalent to 2 seconds. Note that the optimal value of τ should be chosen such that it leaves the robots sufficient time to perceive, process, and reply to the signals sent by other robots. Hence, the optimal value of τ is dependent on the properties of the underlying communication hardware used by the robots. In the experiment shown in Figure 4.5, the elimination process is iterated four times before a one-to-one STC communication link is successfully established between the initiator s-bot and another s-bot within 9 seconds. We replicated the experiment 10 times using the same setup. On average, 3 iterations were required for the termination of the elimination process. The video footage all experiments can be found online at `http://iridia.ulb.ac.be/supp/IridiaSupp2009-006/`.

We also carried out extensive experiments on a heterogeneous robot system composed of one AR.Drone and up to ten foot-bots. Due to hardware limitations (the AR.Drone are not equipped with controllable LEDs) and the very agile nature of the aerial robot, we augmented the communication speed and bandwidth between the aerial and the ground-based robots by using wireless Ethernet for signal transmission as illustrated in Figure 4.6a. For safety reasons, we used a colorless, transparent plexiglass platform installed at 40 cm height from the ground to shield the foot-bots from potential emergency landings of the AR.Drone (see Figure 4.6b). We placed the AR.Drone on the plexiglass platform and varied the number for foot-bots $N \in \{2, 4, 6, 8, 10\}$ distributed in a 1 m x 1.5 m arena with $|C_S| = 2$ colors. For each value of N, we execute 30 experimental runs resulting in a total of 150runs. In each run, the foot-bots were placed in the arena with random orientations and positions together with a light source embodying the point-of-interest (hereafter referred to as POI) to the AR.Drone. The AR.Drone uses the POI to determine the target robot by identifying the closest foot-bot to the POI. While the foot-bots execute the identical behavior-based controller as the s-bots and remain static, the AR.Drone executes the following behavior: i) it elevates to a height of approximately 1.4 m above the plexiglass platform (i.e., ca. 1.8 m above ground level) such that all foot-bots part of the experiment is in the field of view of its downward pointing camera, ii) hovers above the POI by executing a manually tuned PID controller that continuously minimizes the distance between the POI and the center of the image received from the downward-pointing camera, and iii) establishes a one-to-one STC link to the foot-bot closest to the POI before landing on the plexiglass platform. While hovering, we occasionally observed lateral drifts of the AR.Drone caused by impeded airflow in the arena. Therefore, we introduce an additional signal for our

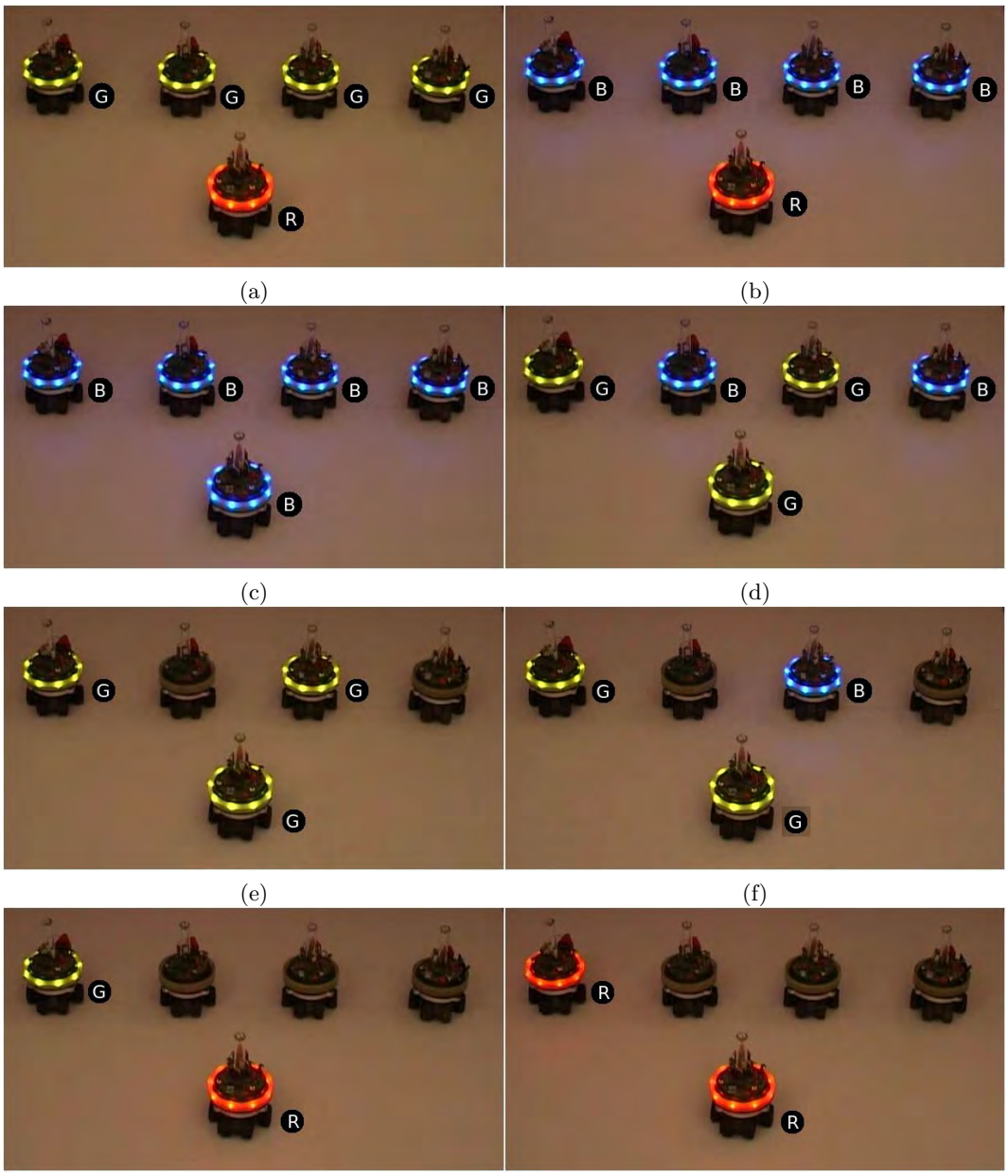
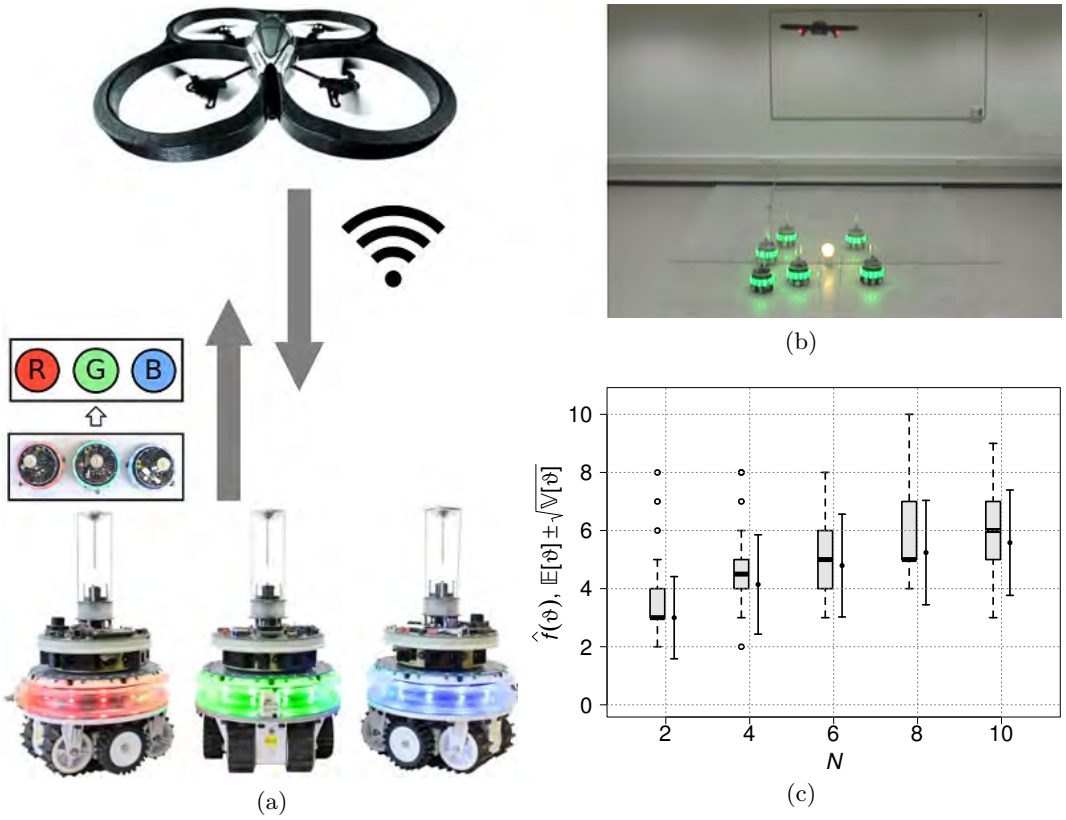


Figure 4.5: Snapshots of an experiment showing the establishment of a one-to-one STC link in a homogeneous system composed of s-bots only. The s-bot in the bottom row assumes the role of the initiator robot. This predesignated s-bot establishes a link with the s-bot on the top-left. The letters next to the s-bots represent the current color displayed: R=red, G=green and B=blue. (a) A `STA` signal is sent by the initiator s-bot. (b) The `STA` signal is acknowledged by all other 4 s-bots. (c) The initiator s-bot initiates the iterative elimination process `EP`. (d) All 4 s-bots remaining in the `EP`, (e) `EP` includes 2 remaining s-bots, (f) `EP` includes 2 remaining s-bots, (g) only the desired s-bot is in the `EP`. The initiator s-bot confirm the termination of the process using the color red and (h) the establishment is confirmed by the selected s-bot. The video footage of this experiment can be found online at `http://iridia.ulb.ac.be/supp/IridiaSupp2009-006/`.



(a) (b) (c)

Figure 4.6: (a) Communication in the heterogeneous robot platform composed of one AR.Drone and multiple foot-bots considered in this chapter. Foot-bots use RGB colors to transmit signals that can be detected by the AR.Drone using its downward-pointing camera. Signal transmission to the foot-bots occurs over wireless Ethernet. (b) The experimental setup showing the airborne AR.Drone at around 1.8 m above ground, six foot-bots, the plexiglass platform, and the light source representing the point of interest (POI). (c) Empirical probability mass function $\hat{f}(\vartheta)$ computed for varying sizes of N and $|C_s| = 2$. Box-plots depict 30 runs, errorbars correspond to predictions of the model ($\mathbb{E}[\vartheta] \pm \sqrt{\mathbb{V}[\vartheta]}$). (d,e,f) Three frames extracted from an application scenario where a one-to-one STC link is established by the AR.Drone to the foot-bot closest to the POI. The STC link is used to send tailored messages to the target foot-bot which uses instructions received to grow an arrow-like morphology.

real robot experiments. The AR.Drone issues a *freeze* signal if not all foot-bots in the experiment are in its field of view due to drifting. The foot-bots respond to this signal by pausing their internal timers as long as the *freeze* signal is perceivable. Hence, the *freeze* signal avoids the fixed threshold τ (set to 200 ms) to be reached when the AR.Drone is unfavorably positioned to continue the elimination process. For each value of N, a video montage composed of 10 runs can be seen online at `http://iridia.ulb.ac.be/supp/IridiaSupp2013-005/`.

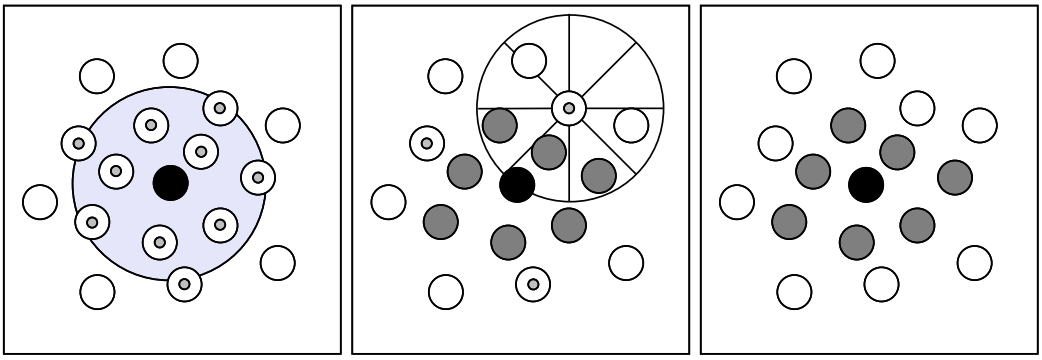
Figure 4.6c shows the observed number of iterations plotted as box-plots together with predictions of the model plotted as errorbars. They show the expected value $\mathbb{E}[\vartheta]$ and the variance $\mathbb{V}[\vartheta]$ of the number of iterations necessary to establish the one-to-one STC link. We compare the empirical distribution of the result with the predictions of the Markov chain model defined above. As the results show, the agreement between the empirical observations and the theoretical predictions is rather

high. In particular, the median values obtained from the real robot experiments (shown in box-plots) approximate well the theoretical expectation of ϑ. However, in the case of $N \in \{6, 8, 10\}$, the empirical observations have a larger variance than the theoretical predictions. This discrepancy can be explained by the limited number of observations gathered from the experiments on the heterogeneous system, and thus is negligible. While the size of N in this set of experiments was increased by a factor of 5 (i.e., form 2 to 10 foot-bots), the average number of iterations only experienced an increase by a factor of 1.6 (i.e., from 3.8 for 2 foot-bots to 6.1 iterations for 10 foot-bots). These results confirm the high scalability of the iterative elimination process predicted by the Markov chain model.

4.2 Establishing a one-to-many communication link

Spatial coordination in multirobot systems often require messages to be transmitted to co-located robots simultaneously. Here we present a protocol extension that allows an already established one-to-one STC link to be expanded to become a one-to-many STC link between an initiator robot and a group co-located robots. Our approach works by iteratively growing a group of robots around a target robot with which a one-to-one STC link has already been established. Note that we are not interested in which individual robots are selected, but only in how many are selected in the spatial cohesiveness of the target group.

An already established one-to-one STC link can be used to create a one-to-many STC link through an iterative growth process using a set $C := \{c_1, c_2, c_3\}$ of situated signals. Before the initiator robot initiates the iterative group process, the target robot is the only initial member. Target group members signal their membership using the predefined signal c_1. At each iteration, the initiator robot may emit the signal c_3 to request a growth of the group. Only those robot that perceive this signal and a target group member will process the request by responding with signal c_2. We refer to these robots as *candidate robots*. In a subsequent step, each candidate robot executes an exclusion mechanism that allows only the *closest candidate robots* to remain in the growth process and hence become a potential group member. As visualized in Figure 4.7, this exclusion mechanism enables candidate robots ensure spatial cohesiveness of the final target group by eliminating those candidate robots from the growth process that have another robot located between itself and the closest target group member. The remaining set of closest candidate robots communicate their candidacies to the initiator robot by emitting c_3. If no signals from candidate robots can be perceived, the initiator robot may finally grant group membership to the closest candidate robots by signaling c_1. Alternatively, if the *potential target group size* (i.e., the number of the closest candidate robots plus the number of current group members) exceeds the required target group size, the initiator robot may issue the signal c_2 to request closest candidate robots to relinquish their candidacies probabilistically. All closest candidate robots are granted group membership if potential target group size is lower than or equal to the target group size desired by the initiator robot. The iterative growth process is repeated until the desired target group size is reached.



(a) (b) (c)

Figure 4.7: The exclusion mechanism executed by candidate robots to achieve cohesive target groups. ● = target group member, ◎ = candidate robot, ○ = hibernating robot, and ● = closest candidate robot. (a) Candidate robots within the communication range of the target robot. (b) Division of perception range into eight equally sized sections visualized on a candidate robot. Three candidate robots exclude themselves and do not become closest candidate robots, as they detect another candidate robot that is in the same section as the closest group member and closer. (c) The target robot and its remaining closest candidate robots from which any robot can become a target group member without breaking the group cohesion.

4.2.1 The iterative growth process

We present details of the iterative growth process using two behavior-based robot controllers we developed, one for the initiator robot and one for the candidate robots. Both controllers are implemented using the FSMs shown in Figure 4.8 and operate under the assumption that $C := \{red, blue, green\}$.

Figure 4.8a shows the FSM executed by the initiator robot. The FSM consists of three states: `HAL`, to halt the growth process; `GRO` to request growth of target group size; and `LEA`, to request closest candidate robots to leave the process probabilistically. Initially, the initiator robot enters the `HAL` state and signals the color *red*. If the current target group size matches the desired size, transition `ta1` is triggered and the initiator robot quits the growth process and stops executing the controller. By contrast, if the current target group size is smaller than the desired target group size, transition `ta2` is taken and the initiator robot enters the `GRO` state and sends the signal *green* to request candidate robots to execute the exclusion mechanism in order to grow a cohesive target group. From the state `GRO`, the initiator robot can move to two states as needed as soon as all candidate robots have finished executing the exclusion mechanism (i.e., when only signals sent by closest candidate robots are perceivable) . If the potential target group size is equal or less than the required target group size, the initiator robot simultaneously grants all closest candidate robot group membership and halts the growth process at an intermediate group size by taking the transition `ta2` and by returning to state `HAL`. Alternatively, if the target group size may be exceeded if all closest candidate robots are granted group membership, the initiator robot requests closest candidate robots to probabilistically relinquish their candidacies by moving to state `LEA` through transition `ta3` and by signaling the color *blue*. The initiator robot iterates in the `LEA` state by taking transition `ta4` until the potential target group size is equal or less than

the required target group size. Finally, when this condition is satisfied, the initiator robot triggers the transition `ta5` to the `HAL` state to grant group membership to remaining closest candidate members.

Figure 4.8b shows the FSM executed by the candidate robots. The FSM is composed of four states: `HIB`, hibernate; `CAN`, candidate robot; `CCR`, closest candidate robot; and `MEM`, member of the target group. No signal is transmitted while in state `HIB`, while the *blue* color is illuminated in state `CAN`, *green* when in state `CCR`, and *red* when in state `MEM`. The controller is initiated in the `HIB` state. If the initiator robot indicates an excess of closest candidate robots by signaling using color *blue*, the transition `tm5` is taken to leave the growth process permanently. The transition `tm1` is taken to enter the `CAN` state if the signal *green* is received from the initiator robot requesting target group growth. In this state, the following procedure is executed to maintain the group's spatial cohesion. As shown in the example in Figure 4.7b, each candidate robot divides the 360 degree perception of the environment into 8 equally sized sectors with an angular range of 45 degrees each. Subsequently, transition `tm1` is taken to leave the growth process if another candidate robot is perceived closer than the nearest target group member within a single sector. Or else, transition `tm2` is triggered to enter state `CCR`. When in state `CCR`, each robot responds to a received *blue* signal (indicating too many closest candidate robots) from the initiator robot by considering the outcome of an independent Bernoulli trial with success probability $p = 0.5$. In case of success, the robot stays in the state `CCR`, otherwise it takes the transition `tm3` to enter the `HIB` state. A single bit of memory is used in state `HIB` to store a binary value that represents the state from which the state `HIB` was reached: `CAN` or `CCR`. This allows closest candidate robots from previous iterations to re-enter the state `CCR` by taking the transition `tm3` while avoiding the `CAN` state entirely.[1] Finally, transition `tm4` is triggered when signal *red* is received from the initiator robot granting membership to the target group.

In what follows, we detail three underlying concepts of the iterative growth process essential to the modeling process that follows in Section 4.2.2: iterations, interaction graph, and growth phases. We hereafter assume the robots to be distributed in a square lattice. For clarity, we refer to the exemplary growth process illustrated in Figure 4.9 in which each robot is at least able to perceive the signals from the robots in its Moore neighborhood. We later assess the impact of the robot distribution assumption in Section 4.2.2 and discuss possible relaxations.

Iterations: we define an iteration such that it can be used to represent time n required to expand a one-to-one STC link to a one-to-many STC link. An iteration is an occurrence of one of the four following sequences of state transitions in the initiator robot's FSM:

(i) `HAL`→`GRO`→`HAL`: The initiator robot requests a target group size growth; the potential target group size does not exceed the required target group size, and thus, the initiator robot grants membership to all closest candidate robots (e.g., at iteration I in Figure 4.9).

[1] When the intermediate group size is close to the target group size, the initiator robot may need to repeatedly request the same set of robots to candidate and withdraw candidacies probabilistically until the target group size is reached. Storing the prior state allows robots in state `HIB` that were previously in state `CCR` to respond to such a request by avoiding the re-execution of the exclusion mechanism and thus reduces the wall clock time of an iteration.

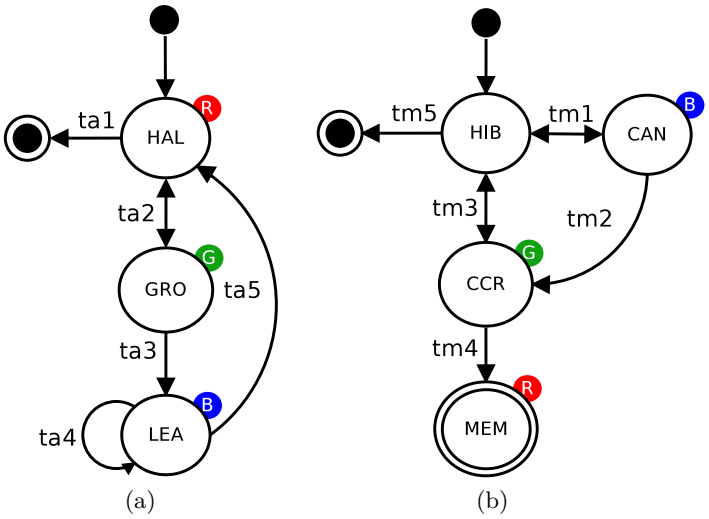


Figure 4.8: The FSMs executed by (a) the initiator and (b) the candidate robots that enable the establishment of one-to-many STC links. The RGB color-based signal emitted in each state is indicated by R, G, or B. HAL: halt, GRO: grow, LEA: leave, HIB: hibernate, CCR: closest candidate robot, CAN: candidate, MEM: member.

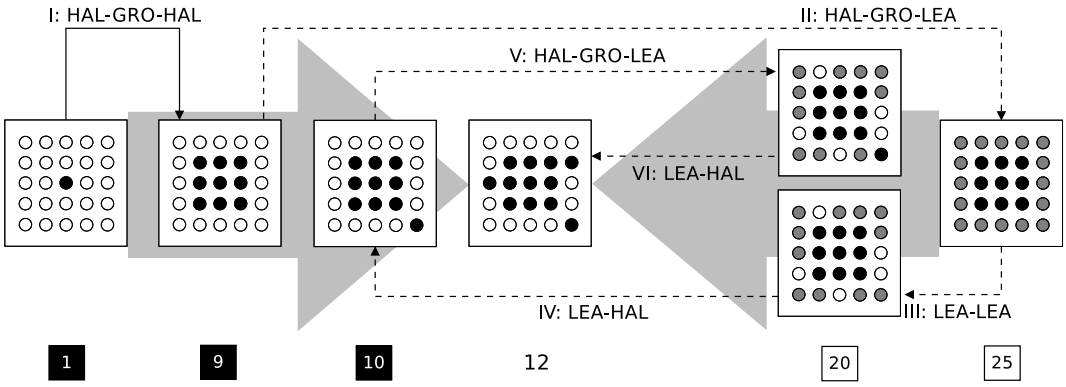


Figure 4.9: A schematic of an iterative growth process around a target robot located in the center of square lattice where the total number of robots $N = 25$ and the target group size $D = 12$. The process requires six iterations (I–VI) before reaching target group size. The sole iteration in the deterministic phase is indicated by a solid arrow whereas the iterations in the stochastic phase are indicated by dashed arrows. Intermediate group sizes (filled boxes) and potential target group sizes (empty boxes) are shown in the bottom. The opposing arrows in the background depict how the iterative growth process constantly pushes the intermediate group size and the total number of participating robots in the growth process towards the target group size after each iteration. ● = `MEM`, ◍ = `CCN`, ○ = `HIB`.

(ii) `HAL`→`GRO`→`LEA`: The initiator robot requests a target group size growth; the potential group size exceeds the target group size and therefore the initiator enters the `LEA` state (e.g., at iteration II in Figure 4.9) requesting closest candidate robots to relinquish their candidacies probabilistically.

(iii) `LEA`→`LEA`: The number of closest candidate robots that relinquish candidacy is too low; the initiator robot loops in `LEA` state and transmits a further signal requesting a further reduction (e.g., at iteration III in Figure 4.9 at which the potential target group size was reduced from 25 to 21).

(iv) `LEA`→`HAL`: Some closest candidate members relinquished their candidacies leaving potential target group size equal to or lower than the target group size; the initiator grants membership to all closest candidate robots (e.g., at iteration IV in Figure 4.9, the number of closest candidates robots is reduced from 11 to 1 upon which the sole closest candidate robot was granted group membership allowing the intermediate group size to be increased from 9 to 10).

We define this particular set of state transition sequences due to three important properties. Firstly, because it is complete. That is, any possible run of the growth process (i.e., for any N and D) can be decomposed into multiple iterations of which each corresponds to either one of these state transition sequences. For instance, the growth process shown in Figure 4.9 requires six iterations to reach target group size. Secondly, when a LEDs and camera-based communication modality is in use, all four state transition sequences have the same time and computation costs making an analysis of iterations based on them meaningful. To see why the cost is constant, note that each state transition sequence ends with a transition that is triggered when the number of closest candidate robots changes in the system. Consequently, the cost of detecting an iteration by an initiator robot can be kept constant by applying a color-based similarity matching technique [72] of constant cost on two subsequently retrieved images such that each change in the number of closest candidate robots can be detected. Thirdly, robots in state `CAN`, i.e., candidate robots, are completely disregarded when defining iterations using this particular set of state transition sequences. This leads to a much smaller state space in the model presented in Section 4.2.2 as candidate robots do not have to be considered at all.

Iteration graph: is an abstraction mechanism we consider to derive a non-spatial characterization of the growth process. Two connected nodes in such an interaction graph correspond to a target group member and one of its closest candidate robots. Using such an interaction graph, we consider a sequence $\{a_0, a_1, \dots\}$ of which each element corresponds to the total number of closest candidate robots available at each intermediate group size. For the intermediate group sizes 1, 9, and 10 shown in Figure 4.9, the sequence corresponds to $\{8, 16, 15\}$. For instance, at intermediate group size 1 the total number of closest candidate robots is 8. This is a result of the exclusion mechanism shown in Figure 4.7 that limits a group member's closest candidate robots to its Moore neighborhood regardless of a robot's communication range if at least the robot's Moore neighborhood is covered. At intermediate group size 9, closest candidate robots are all 16 outmost robots.

Growth phases: in a square lattice distribution of robots, the iterative growth process can be distinguished into two distinct phases: a deterministic and a stochastic

phase. The deterministic phase is composed of a series of HAL→ GRO→HAL iterations that occur repeatedly until the potential target group size is greater than or equal to D. We refer to these iterations as deterministic, as at each iteration, the number of new members entering the group is a priori predictable as a function of the relative positioning of the potential recipient robots in the lattice. For instance, in the example shown in Figure 4.9, iteration I will always result in 8 newly added group members. If the potential target group size equals D, the growth process terminates without entering the stochastic phase. On the other hand, if the potential target group size exceeds D as at iteration II of the growth process shown in Figure 4.9, then the deterministic phase ends and the growth process enters the second stochastic phase. Each iteration in this phase is an occurrence of one of the three remaining iteration types (HAL→GRO→LEA, LEA→LEA, LEA→HAL). These iterations are stochastic because each of them contains a probabilistic response to a leave request (LEA). The stochastic phase continues until the target group size is reached (see iterations II–VI in Figure 4.9). In the stochastic phase, each iteration of type (iii) and (iv) allows the potential target group size to act as a monotonically decreasing upper bound on the number of robots involved in the growth process. After each iteration of type (i) and (ii), i.e., also during the deterministic phase, the intermediate group size representing a lower bound increases monotonically. These two bounds meet at the target group size as showing using the opposing arrows in the background of Figure 4.9.

4.2.2 Square lattice distribution-based Markov chain model

Given a number of N potential recipient robots, we are interested in modeling the time required for the initiator robot to grow a one-to-one link into a one-to-many STC link to a target group of size D. In our model, each time step n corresponds to one iteration as defined earlier. For the reasons presented in Section 4.1.2, we consider time to be discrete. We assume the potential recipient robots to be distributed in a square lattice with each robot able to perceive at least the robots in its Moore neighborhood. We then assess the impact of such an assumption on model predictions and discuss possible relaxations.

The distinction of growth phases allows us to build a model of the iterative growth process such that each phase can be modeled separately. In the following section, we present separate models for each growth phase such that the estimation of $D = D_{det} + D_{sto}$, where D_{det} is the number of robots added to the intermediate group during the deterministic phase, and D_{sto} is the number of robots added to the intermediate group during the stochastic phase.

4.2.2.1 Deterministic phase model

In the interaction graph we defined, any target robot that is not located at the edges of a square lattice distribution can be represented using a node connected to 8 neighbors (i.e., the target robot has 8 closest candidate robots). The sequence of closest candidate robots for intermediate group sizes of any iterative growth process that starts in such a condition is thus given by $\{8k, \forall k \in \mathbb{N} > 0\}$ in an infinite lattice while the number of iterations required to exhaust the deterministic phase is

$$\arg\max_{k>0} \left(1 + \sum_{i=1}^{k} 8i < D\right). \tag{4.5}$$

The target group size at the end of a deterministic phase is defined by $D = D_{det} = 1 + \sum_{i=1}^{k} 8i$.

4.2.2.2 Stochastic phase model

We define a Markov chain $\{Y_n, n \in \mathbb{N}\}$ with the random variable $Y \in \Omega := \{1, \ldots, w\}$ that represents the state of the growth process and Ω that represents the state space with $w = 2(N - D)(D - D_{det}) + 1$ states. Each state $Y = y$ is characterized by:

- η_y^m, current number of target group members;
- η_y^c, current number of closest candidate robots;
- $\eta_y^{c'}$, number of closest candidate robots at the previous iteration;
- η_y^h, number of hibernating robots.

Although the growth process uses a single bit of memory, we apply the formalism of time-homogeneous Markov chains using a chain of order 2. The memory is modeled using the variable $\eta_y^{c'}$: when the memory is necessary, $\eta_y^{c'}$ equals the number of closest candidate robots at the previous step of the chain, when it is not necessary, $\eta_y^{c'} = -1$. All states are transient with the exception of $Y = y$ such that $\eta_y^m = D, \eta_y^c = 0, \eta_y^h = N - D, \eta_y^{c'} = -1$ which is the only absorbing state.

At step n, the process on the chain will move from the current state $Y_n = i$ to the next state $Y_{n+1} = j$ with probability π_{ij}. We define the stochastic transition matrix $\Pi_Y := (\pi_{ij} : i, j \in \Omega)$ of the chain $\{Y_n, n \in \mathbb{N}\}$ as

$$\begin{cases} \pi_{ij} = C_r^q p^q (1-p)^r, & \text{if } c_{1.1} \vee c_{1.2} \vee c_{1.3}, \quad (4.6a) \\ \pi_{ij} = 1, & \text{if } c_2 \vee c_3, \quad (4.6b) \\ \pi_{ij} = 0, & \text{otherwise}, \quad (4.6c) \end{cases}$$

where the adjacency conditions $c_{1.1}$, $c_{1.2}$, $c_{1.3}$, c_2, c_3 between states i and j are summarized in Table 4.2. Case 4.6a corresponds to a binomial distribution where $r = \eta_i^c$, i.e., the number of closest candidate robots withdrawing candidacies probabilistically (in state $Y_n = i$) with success probability p, and $q = \eta_j^c + \eta_i^m - \eta_j^m$, i.e., the number of closest candidate robots retaining candidacies (in state $Y_{n+1} = j$). The adjacency conditions $c_{1.1}$, $c_{1.2}$, $c_{1.3}$ define all possible states i and j such that the transition $i \to j$ corresponds to an iteration `LEA` $\to$ `LEA` or `LEA` $\to$ `HAL`. Case 4.6b defines the transition probability between adjacency conditions c_2 and c_3 to 1. The adjacency condition c_2 covers all possible pairs of states (i, j) such that the transition $i \to j$ corresponds to an iteration `HAL` $\to$ `GRO` $\to$ `LEA`. The adjacency condition c_3 defines the absorbing state of the system. Finally, Case 4.6c defines that transitions not covered by the adjacency conditions will never occur. Given the interaction graph, note that this Markov chain model can be applied to any spatial distribution of the robots by solely changing the adjacency conditions.

4.2.2.3 Analyzing model predictions and scalability

We first validate the reliability of the Markov chain model by comparing model predictions against data collected from simulation-based studies. Second, we assess how a square lattice distribution of robots assuming a Moore neighborhood-based interaction graph compare to the model predictions using data acquired from

Table 4.2: The adjacency conditions between states i and j of the stochastic transition matrix Π_Y

	η_j^m	η_j^c	η_j^h	$\eta_j^{c'}$
$c_{1.1}$	$= \eta_i^m$	$= 0$	$= \eta_i^c + \eta_i^h$	$= \eta_i^c$
$c_{1.2}$	$= \eta_i^m$	$\leqslant \eta_i^c$	$= \eta_i^c + \eta_i^h - \eta_j^c$	$= -1$
$c_{1.3}$	$> \eta_i^m$	$= 0$	—	$= (\eta_i^c + \eta_i^m - \eta_j^m) \vee -1$
c_2	$= \eta_i^m$	$\leqslant \eta_i^c$	$= \eta_i^h - \eta_i^c$	$= -1$
c_3	$= D$	$= 0$	$= N - D$	$= -1$

simulation-based experiments in which we vary the spatial distribution of the robots from a square lattice to a random distribution. Third, we present the results of our scalability studies of the growth process.

We perform a canonical decomposition of Π_Y in the transition probability matrix Γ_Y and the fundamental matrix Σ_Y. The number of iterations necessary to reach a target group size D is defined as the random variable $\varphi = \varphi_{det} + \varphi_{sto}$, where φ_{det} is the number of iterations during the deterministic phase and φ_{sto} is the number of iterations during the stochastic phase. φ_{det} is given by Equation (4.5), while φ_{sto} is derived from the Markov chain model that provides the number of state transitions before the process reaches its only absorbing state. The cumulative distribution function $F(\varphi)$ is then given by Equation (4.2) and the variance $\mathbb{V}[\varphi]$ is given by Equation (4.4), while the expected value is given by

$$\mathbb{E}[\varphi] = \arg\max_{k>0} \left(1 + \sum_{i=1}^{k} 8i < D \right) + \xi \Sigma_Y, \tag{4.7}$$

where ξ is a column vector filled exclusively with the value 1.

The reliability of the Markov chain model is validated using multiple probability-probability (P-P) plots. We use them to compare the cumulative distribution function $F(\varphi)$ with the empirical distribution $\hat{F}(\varphi)$ retrieved from simulation for varying sizes of potential recipient robots N and target group size D. The results are presented in a 3×3 matrix from which each cell is a different P-P plot. Plots with identical N values are in the same row. Plots in the same column have target group size $D = D_{det} + d \cdot (N - D_{det})$, where d is a constant fraction of robots remaining in the stochastic phase $(N - D_{det})$. For each combination of N and D, we run 1000 independent simulation runs and register the number of iterations required to reach D. In a P-P plot, the distributions under examination agree when the points in the plot lie on the diagonal of the square due to the confinement of P-P plots to unit squares. As the results show in all nine plots shown in Figure 4.10, the function $(F(\varphi), \hat{F}(\varphi))$ returns data points in the close proximity of the diagonal of the respective cell. This indicates a good agreement between theoretical and empirical distributions for each combination of parameters considered. Based on this observation, we conclude our model to be reliable.

We further compare model predictions with results acquired from simulation to investigate the difference between the duration of the iterative growth process between a random distribution and a square lattice alignment of the potential recipient robots. Figure 4.11 shows the results for increasing $N \in \{9, 25, 49, 81, 121\}$ plotted against target group size $D = D_{det} + 50\% \cdot (N - D_{det})$ set to a constant fraction of

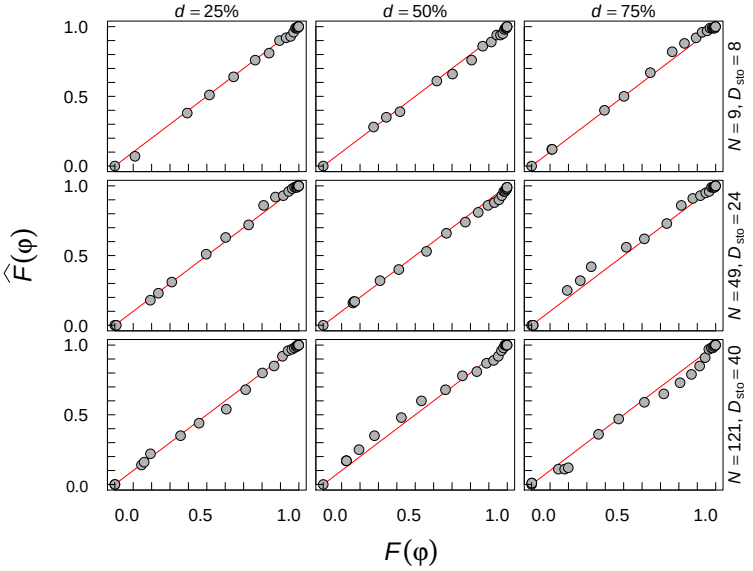


Figure 4.10: P-P plots comparing theoretical and empirical cumulative distribution functions of φ with $N \in \{9, 49, 121\}$ and $D = D_{det} + d \cdot (N - D_{det})$ with $d \in \{25\%, 50\%, 75\%\}$. Each data point (plotted as grey circles) is an average of 1000 simulation runs.

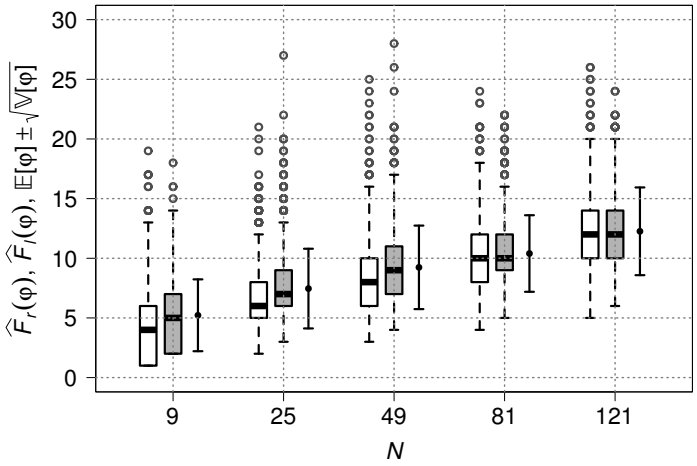


Figure 4.11: Empirical cumulative distribution functions computed for $N \in \{9, 25, 49, 81, 121\}$ and $D = D_{det} + 50\% \cdot (N - D_{det})$ with 1000 simulation runs each. $\hat{F}_r(\varphi)$ represents randomly distributed robots $\hat{F}_l(\varphi)$ represents square lattice distribution, and $\mathbb{E}[\varphi] \pm \sqrt{\mathbb{V}[\varphi]}$ is the theoretical expectation.

the stochastic phase. For empirical values resulting from simulation runs, i.e., for $\hat{F}_r(\varphi)$ representing randomly distributed robots and for $\hat{F}_l(\varphi)$ representing robots distributed in a square lattice, box-plots are presented. Each of them represent 1000 simulation runs we executed to mitigate the effect of random fluctuations. Model predictions $\mathbb{E}[\varphi] \pm \sqrt{\mathbb{V}[\varphi]}$ are plotted using error bars. As the results show, the growth process for smaller $N \in \{9, 25, 49\}$ terminates the fastest when robots are randomly distributed. For larger $N \in \{81, 121\}$, the model predictions correspond well with the results from simulation independent of the spatial distribution of the potential recipient robots. The discrepancy between the model predictions and empirical data can be explained by the increased probability of a robot to perceive at least one neighbor in each of its eight sections of perception for larger N. The average degree of connectivity in the resulting interaction graph (for larger N) may hence resemble that of a square lattice with eight nodes – the spatial assumption underlying the Markov chain model. The model therefore provides a reliable and quantitative approximation of the duration of the growth process given the potential recipient robots are situated in a densely packed group.

Once an initial one-to-one link has been established, two methods can be chosen to expand the link to a target group: (i) by repeating the elimination process presented in Section 4.1 multiple times such that one robot is added to the target

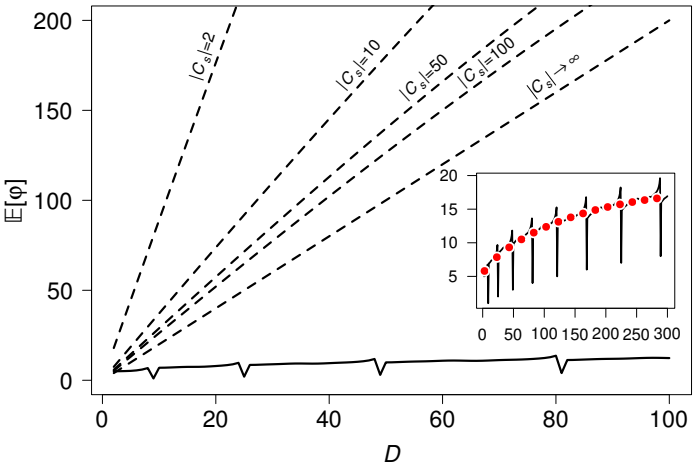


Figure 4.12: Scalability of target group selection. $\mathbb{E}[\varphi]$ for $N = 121$ in a square lattice distribution. Dashed lines show results of repeating the elimination process with varying $|C_s|$. Solid line represents the growth process. Inset shows the performance of the growth process for larger values of D. Red dots are fitted values.

group after each repetition, or (ii) by following up the one-to-one link by the growth process described in Section 4.2. We study the difference in the scalability properties between both these approaches. For this purpose, we compute the expectation value of φ for $N = 121$ using the Markov chain model and plot it against $D \in \{2, \ldots, 100\}$.

The results of the scalability studies are plotted in Figure 4.12. We vary the number of distinct signals C_s available to the iterative elimination process and plot them using dashed lines while the results obtained from the iterative growth process is plotted using a solid line. As the results show, the iterative growth process clearly outperforms the iterative elimination process regardless of the size of $|C_s|$. Even when the theoretical case is considered in which $|C_s|$ tends to infinity, the dominance of the iterative growth process over the iterative elimination process still holds. We also observed that the iterative growth process scales approximately as $D \cdot log(D)$ for larger values of D. We plot this result in the inset in Figure 4.12 in which fitted values are shown in red circular markers. The spikes in the inset are caused by faster termination of the iterative growth process for values of D that can be reached exclusively through iterations in the deterministic phase only (e.g., $D \in \{9, 25, 49, \ldots\}$). In summary, establishing an STC link to a target group is faster through the iterative growth process in all cases considered. However, there may also be advantages in repeating the iterative elimination process in particular application scenarios. For instance, when communication links are required to particular robots rather than to a cohesive group of a specific size.

4.2.3 Experiments and results

We first validate the establishment of one-to-many STC links using a homogeneous robot team composed of 5 s-bots. Second, we use a heterogeneous team composed of up to 9 foot-bots and one AR.Drone to assess how a square lattice distribution of robots assuming a Moore neighborhood-based interaction graph effects the model predictions contrasted against data acquired from real robots experiments in which we vary the spatial distribution of the robots from a square lattice to a random distribution (see Figure 4.13). The behavioral controller for the robots are implemented based on the FSMs presented in Figure 4.8. In the case of the robots in the homogeneous team, the behavioral controller is initiated with the logic of the FSMs presented in Figure 4.2.

Figure 4.14 shows snapshots of an experiment we ran using s-bots, where $N = 4$ and $D = 2$. We placed 4 s-bots assuming the role of potential recipient robots in

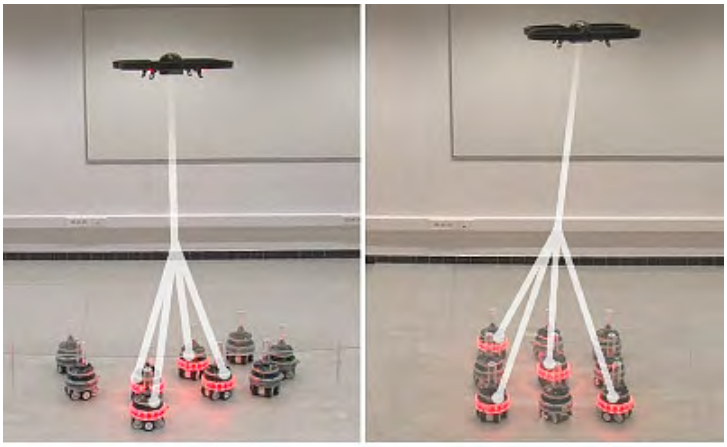


Figure 4.13: Snapshots of experiments with the heterogeneous team where $N = 9$ and $D = 4$: (left) random distribution and (right) square lattice distribution. The overlays illustrate one-to-many STC links established between the AR.Drone and the respective target groups.

the shape of an arch around a predesignated s-bot, which assumes the role of the initiator robot. All robots are in the vision range and therefore in the communication range of all other robots. The timer threshold τ is set to 20 control steps (equivalent to 2 seconds). In the experiment, the elimination process is iterated 5 times and lasted 10 seconds before a one-to-one STC link is established. The expansion to one-to-many link to include one further s-bot lasted another 8 seconds and included the iteration `HAL`→`GRO`→`HAL`. We replicated the experiment 30 times. 10 times for a different each value of $D \in \{2, 3, 4\}$. On average, the STC links to the target groups were established within 16, 20, and 22 seconds, respectively. The video footage of the experiment presented in Figure 4.14 can be found online together with others at `http://iridia.ulb.ac.be/supp/IridiaSupp2009-006/`.

Figure 4.13 shows snapshots from experiments we carried out using the heterogeneous team. We considered two spatial distributions of the potential recipient robots: random (Figure 4.13 left) and a square lattice with a side length of 0.4 m (Figure 4.13 right). We study how randomly distributed potential recipient robots may affect the duration of the growth process with respect to the model predictions that assume a square lattice distribution. To mitigate the stochasticity underlying the iterative elimination process, we assume an already established one-to-one STC link between the AR.Drone (i.e., the initiator robot) and one foot-bot. For each value of $D \in \{2, 4, 8\}$ and spatial distribution, we execute 30 runs resulting in a total of 180 runs. The experimental setup considered is similar to that presented for the heterogeneous team in Section 4.1.3. However, in the setup used here, no light source is considered. Therefore, the AR.Drone hovers above the foot-bots by relying on a PID controller that minimizes the distance between the center of the bounding box that includes all nine foot-bots and the center of the image returned by the camera. When all nine foot-bots are the in the field of view of the AR.Drone, the control logic of the initiator robot shown in Figure 4.8 is executed. Otherwise, when fewer than nine foot-bots are in its field of view, the AR.Drone issues the *freeze* signal to stop the the timer running on the foot-bots. The fixed threshold τ is set to 200 ms which corresponds to two control cycles of a foot-bot – sufficient to receive multiple messages via wireless Ethernet from the AR.Drone.

Figure 4.15 shows results obtained using random distribution and square lattice distribution together with the model predictions. It can be noted that the median value of the square lattice distribution and the expectation value of the model predictions for $D = 2$ and$D = 8$ do not vary significantly. The amplitude of standard

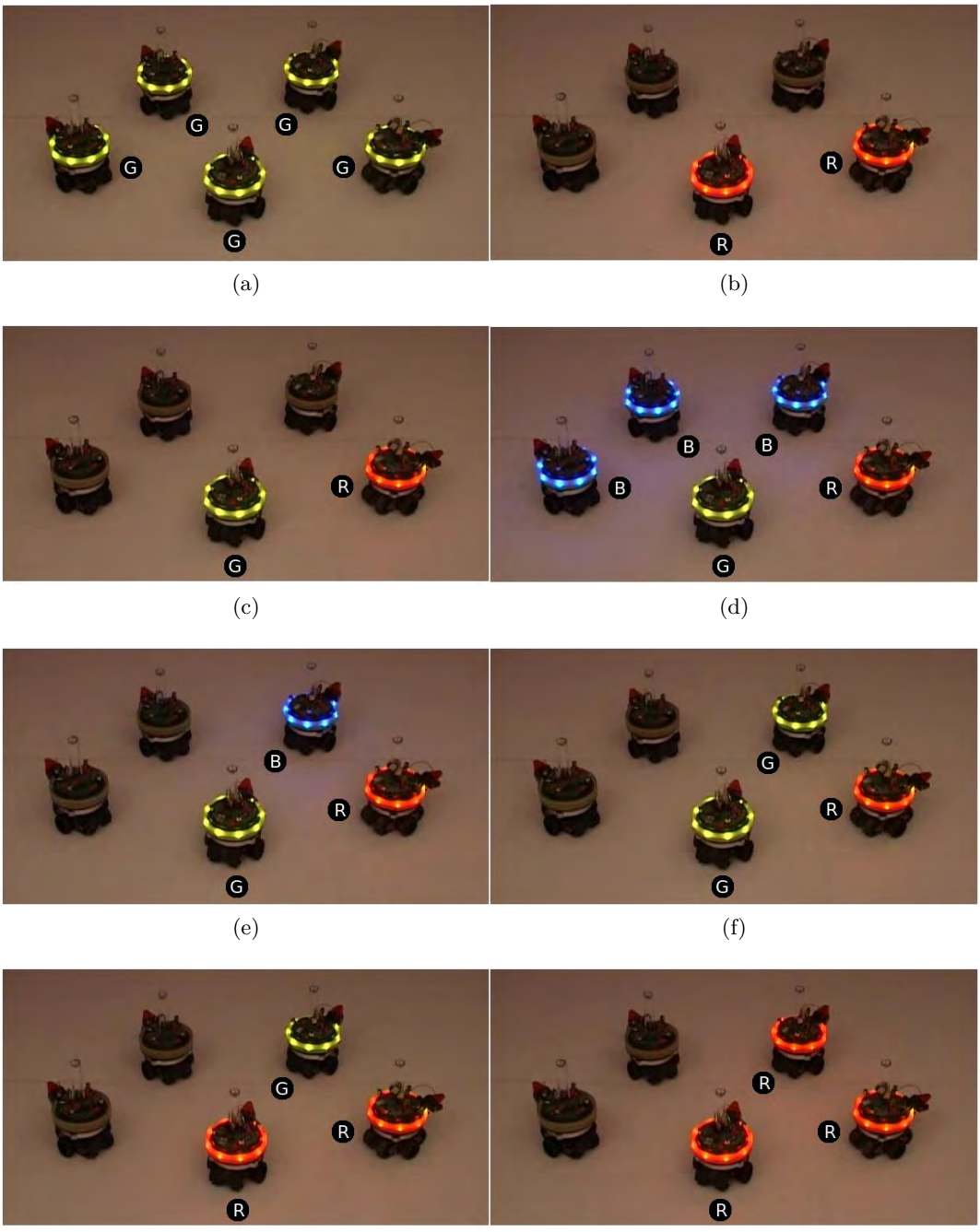


Figure 4.14: Snapshots of an experiment showing the establishment of a one-to-many in a group of 5 s-bots. The initiator robot is the s-bot in the center and target group size $D = 2$. The signals emitted are illustrated using the letters shown next to the emitting robot: R=red, G=green and B=blue. (a) The initialization of the experiment . (b) The one-to-one communication link is established to the robot on the right. The selection of this particular robot is entirely random. (c) The initiator robot requests for the growth of the group. (d) All three remaining s-bots candidate by illuminating blue. (e) The closest candidate robot is determined. (f) The closest candidate robot signals its candidacy by illuminating green. (g) The initiator robot grants group membership by emitting the red signal. (h) The new target group member confirms the membership by illuminating the LEDs in red.

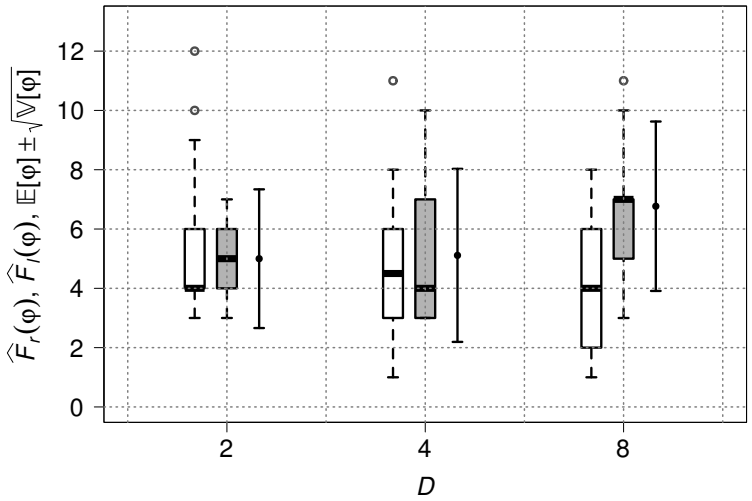


Figure 4.15: Empirical cumulative distribution functions of real robot experiments with $\hat{F}_r(\varphi)$ for randomly distributed foot-bots, $\hat{F}_l(\varphi)$ for foot-bots distributed in a square lattice plotted against theoretical expectation $\mathbb{E}[\varphi] \pm \sqrt{\mathbb{V}[\varphi]}$ for $N = 9$ and $D \in \{2, 4, 8\}$. Each box-plot represents 30 runs. Empty boxes are for $\hat{F}_r(\varphi)$ and filled boxes are for $\hat{F}_l(\varphi)$. The error bars represent the model predictions.

deviation is also well approximated. However, in the case of $D = 4$, the data collected from our experiments do not correspond well to the model predictions. One reason for this outcome could be limited sample of size of 30 runs causing random fluctuations of the estimated distributions. In the case of randomly distributed robots, we observe that the growth process terminates faster than the expectation value of the model in all cases. We consider this as a direct consequence of the sizes chosen for D that requires stochastic phase iterations in all cases. For instance, in the square lattice distribution, all eight neighbors of the target robot are closest candidate robots and hence no deterministic iteration of the type `HAL`→`GRO`→`HAL` can occur for $D \in \{2, 4, 8\}$. On the other hand, when distributed randomly, some foot-bots may occlude others and render intermediate group sizes such as 2,4, or 8 more probable. As a consequence, the AR.Drone may avoid stochastic phase iterations entirely and reach the target group size $D \in \{2, 4, 8\}$ with deterministic phase iterations only. We confirm this observation by visually validating the video footage of the experiment runs with random distribution of the foot-bots. In conclusion, for small values of N, the model predictions based on square lattice-based interaction graph overestimates the duration of the growth process as random distributions often do not result in the dense, structured neighborhoods underlying the Markov chain model.

4.3 Discussion

We address three potential issues with the approach described in this chapter to enable spatially targeted communication in multirobot systems and present possible solutions:

1. Asymmetrical communication between initiator and potential recipient robots

2. Parallelism in heterogeneous teams

3. Noisy or wrongly detected signals

Both protocols described in this chapter assumes that all potential recipient robots (i.e., robots in the communication range of the initiator robot) are able to communicate with the initiator robot (i.e., the initiator robot is in the communication range of each potential recipient robot). If this assumption does not hold true in a certain scenario, the issue of asymmetrical communication arises which causes the initiator robot to be ignorant of response messages transmitted by potential recipient robots. As a consequence, initiator robots may terminate iterative elimination processes assuming that a one-to-one STC link can be established while multiple robots still remaining the process. Similarly, excess potential recipient robots can become members of a target group without the initiator's knowledge. We avoided the asymmetrical communication problem by positioning each robot in the communication range of all others in order to enable global communication. An alternative to solving the symmetric communication problem by using global communication would be for the initiator robot to calculate the probability of only one target robot remaining in the elimination process by providing the number of iterations already conducted and an estimate of potential recipient robots to the Markov chain model presented in Section 4.1.2. To be on the safe side, the estimates can be rounded up by an order of magnitude for a very low cost in terms of additional iterations. This is ensured by the logarithmic relationship between the number of iterations and the number of potential recipient robots (see Figure 4.4c). In the one-to-many case, however, this solution is not suitable and therefore requires a behavioral solution from initiator robots. The key is to prevent robots outside of the communication range from becoming part of a target group. For instance, this can be achieved by the AR.Drone by executing the iterative growth process in such a way that the boundaries of the group never expand beyond its own field of vision.

In the heterogeneous team considered in this chapter, we use the AR.Drone robot as the initiator robot. The fact that is uses non-situated broadcast communication (wireless Ethernet) to communicate with the foot-bots make it impossible for multiple STC links to be established in parallel. This is because, foot-bots would not be able to distinguish between signals from the different initiator robots. However, this is no longer an issue as the initiator uses situated communication such as colors displayed using LEDs. In the next chapter, we present heterogeneous teams composed of multiple foot-bots and an eye-bot demonstrating this type of bidirectional situated communication. In such a setup, multiple initiator robots can establish STC links in parallel as potential recipient robots will be able to use the inherent properties of situated communication to distinguish between signals from the different initiator robots (e.g., based on distance) and therefore can choose which STC process to participate in.

Most situated communication technologies (e.g., ultrasound, infrared, or LED light) are subject to inherent noise that may lead to wrongly detected signals. The protocols presented in this chapter can cope with such signals resulting from the underlying hardware. A wrongly detected signal during the elimination process can cause the target robot to eliminate itself. In this case, the initiator robot simply needs to stop transmitting signals and restart the process. On the other hand, if a signal is wrongly detected by any of the robots during the growth process, the process still can be corrected by looping through a few additional iterations. However, both these corrective behaviors have time costs we try to limit by using filtering techniques that reduce the number of wrongly detected signals. Each signal

receiving robot processes multiple camera frames before a decision is made about what signal has been detected. This simple technique proved to be sufficient to completely eliminate signal detection errors in our experiments.

4.4 Related work

Established spatially targeted communication links can enable spatial coordination in a multirobot system as communication partners can be chosen based on their location. Many robots systems consider specialized hardware to achieve this kind of coordination. For instance, an infrared-based technology was presented in [73] for simultaneous localization and communication (hence situated communication) between neighboring robots. This pioneering work has led to the development of multiple derivations [74, 33] targeting a variety of robot platforms. While these devices are not explicitly been designed for spatially targeted communication, they still can be used by robots for the same purpose by first broadcasting robot IDs among peers to then follow up with communication data marked with IDs of the targeted communication partners. In fact, this method is used by the extending robot in the guidance algorithm (see Algorithm 2 presented in Section 3.1.2) to guide the recruit to the desired extension point. Researchers have also presented devices to enable relative localization based on ultrasound [75, 76] without the exchange of message payload. Peer robots using these devices can be located but no messages can be exchanged using the same device. In general, however, not many robot systems have access to specialized hardware and the features they provide. Although less precise, an alternative and more general method to enable relative localization in multirobot systems could be the probabilistic framework presented in [77].

A more often used technology to achieve spatial coordination in multirobot systems is wireless Ethernet (IEEE 802.11). Systems that employ wireless Ethernet often employ a technique referred to as "tightly coupled" coordination [78, 79, 80, 81] in the literature in which broadcast messages from message senders are channeled to message receivers (either directly or via intermediate stations). In tightly coupled systems, perceptual or computational capabilities are shared by a message sender while also revealing its identity and its capabilities through the messages transmitted. The internal state of each message receiving robot then determines whether a received message should be processed. Alternatively, the information regarding which robot to process what kind of message is predetermined. Wireless Ethernet has been used [82, 83] in this manner to achieve coordination between aerial and ground-based robots by also combining GPS and environment maps or georeferenced coordinates. The ZigBee wireless technology (IEEE 802.15.4) has been applied in [84] to coordinate a team of micro quadrotors flying formations in 3D space. In this work, an externally installed motion capture system delivers positional data about each micro quadrotor to a central base such that the station can compute and control each team member's position. Decentralized solutions based on Bluetooth [85, 86] have also been proposed to enable simultaneous communication between pairs of robots. As localization information is absent when using this technology, a priori knowledge is used by the robots as the are pre-programmed to localize themselves based on landmarks distributed in the environment. While a priori information and pre-determined communication channels may be feasible for robot teams composed of few robots, these solutions tend not to scale in systems that consider a large teams [25, 87].

Another often used technique is to combine vision with unique IDs such as markers [80, 81] or barcodes [88] that reveal the identity of the robot on which they are mounted. An aerial robot was presented in [89] in which an on-board camera is used to locate and identify peer robots on the ground in order to compute and provide their relative location with respect to one another. While visual identification may not always be feasible or reliable in air-ground teams due to occlusion, low illumination, etc., teams that consider robot IDs and support simultaneous relative localization (as it is mostly the case in vision-based systems) have the potential to engage in spatially targeted communication. Robots in such systems can employ any wireless communication technology to address interestingly located peers directly by broadcasting messages marked with addressee ID. A further vision-based alternative is to use a combination of cameras and LEDs to detect neighboring robots (without detecting their identities) and their relative locations. Simultaneous communication can then occur by displaying different LED colors by the message sender. Such low-bandwidth, vision-based communication similar to the one used in this chapter has been successfully applied to study a series of tasks that require spatial coordination such as path formation [90], collective transport [23], and morphology control [22].

4.5 Summary

In this chapter, we identified spatially targeted communication as an essential form of communication in multirobot systems and presented a novel approach to establish STC links. We presented two protocols that enable robots in a decentralized system to establish ad-hoc communication links to particular robots based on their location in the environment. We have shown using extensive experiments that our approach can be applied to both purely ground-based robot systems and to heterogeneous robot teams composed of aerial and ground-based robots. Our approach is highly portable as it can be implemented using standard hardware such as cameras and LEDs found on many existing robotic platforms. Furthermore, we presented formal descriptions of both protocols and showed that the protocol used to establish one-to-one STC links scales logarithmically with respect to the number of potential recipient robots while the establishment of one-to-many STC links scales logarithmically with respect to the target group size. This is particularly important as multirobot systems are starting to consist of large numbers of individual robots requiring scalable communication solutions.

In the next chapter, we combine the research work presented both in this chapter and in previous Chapter 3 to present a cooperation methodology for coordination between aerial robots and ground-based robots. We demonstrate application scenarios related to self-assembly and show how established STC links can be followed up by aerial robots to send useful information to self-assembling robots such that they can operate on the ground without requiring a priori knowledge of the task or environment.

CHAPTER 5

Supervised morphogenesis

In this chapter, we combine the work presented in the previous two chapters and introduce supervised morphogenesis – a morphology control methodology enabling cooperation between aerial and self-assembling robots. Supervised morphogenesis exploits the perspective of the environment available to aerial robots from elevated positions and assigns all decision-making authority to these robots regarding when to trigger self-assembly and what target morphologies to form on the ground. That is, self-assembling robots rely on aerial robots to monitor the environment and provide the guidance required to form new morphologies as a function of task or environment.

Heterogeneous multirobot systems composed of aerial and ground-based robots have been the focus of many studies [91]. The interest in such systems is mainly motivated by the fact that such systems have the potential to solve tasks that require capabilities that go beyond those of a single robot type. For instance, while aerial robots can explore large areas rapidly, ground-based robot can carry higher payloads and manipulate objects on the ground. In recent years, a surge in technology has led to the development of aerial robots [92] able to maneuver through previously unreachable environments, such as inside obstacle-filled buildings and densely loaded warehouses [93, 94, 95]. Researchers have also proposed innovative designs [96, 97] that allow aerial robots to operate in highly cluttered environments and even be resilient towards potential collisions. Despite these advances, almost no research effort has been dedicated to study how aerial robots may use their capabilities to assist self-assembling robots overcome some of their inherent limitations. Here, for the first time, we put aerial robots and self-assembling robots on a convergence path and demonstrate how such a cooperation can increase overall system autonomy and extend the range of tasks that can be solved by the robots on the ground.

Simultaneous deployment of cooperating robots with different sets of sensors and actuators calls for the resolution of multiple challenges related to perception, decision and action [98]. This is particularly true when robots cannot rely on the availability of a common frame of reference (such as GPS) or when they do not have access to global information such as maps of the environment to facilitate cooperation. In Section 5.1, we present the underlying control methodology of supervised morphogenesis and present a solution regarding perception, decision and action for both aerial and self-assembling robots. We present a first case study in Section 5.2 in which we use a robot team composed of one AR.Drone and multiple foot-bots. In Section 5.3, we present a second and more comprehensive case study in which we consider a hill-crossing task using a team composed of an eye-bot and multiple foot-bots. Then, we quantify the performance benefits (in terms of task completion

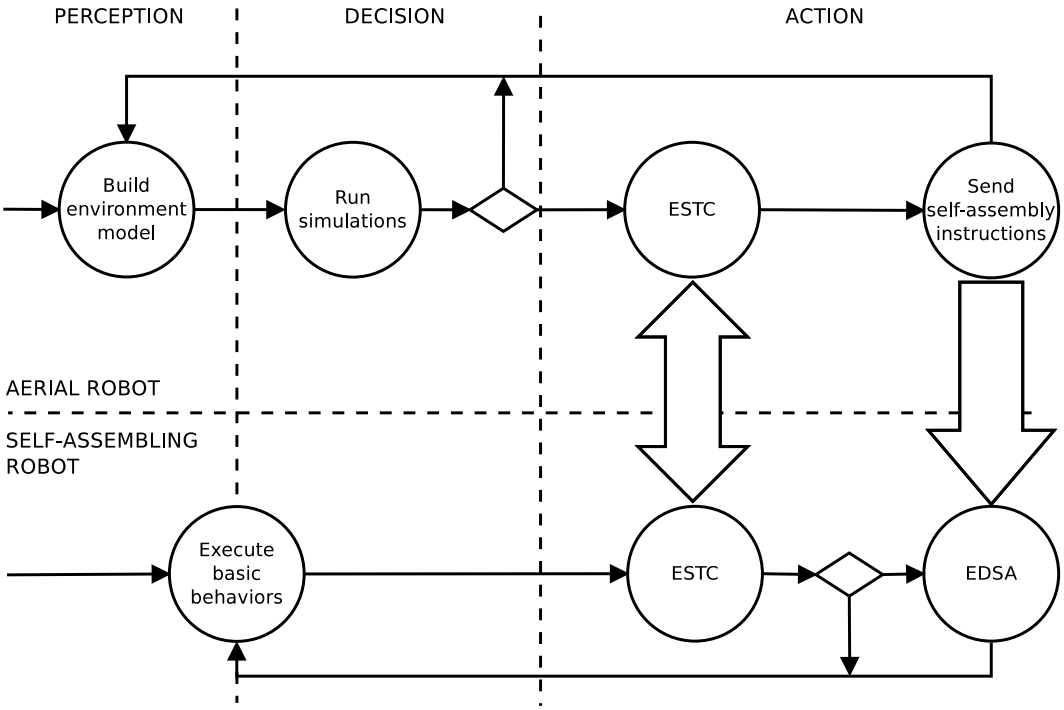


Figure 5.1: Overview of supervised morphogenesis. Control states (shown in circles) provide a solution related to perception, decision or action to both robot types. Basic behaviors (such as phototaxis and obstacle avoidance) executed by self-assembling robots are task-dependent and may include both perception and decision-related activities. Interactions occurring between the two robot types and their directionalities are shown using according arrows. ESTC: establish spatially targeted communication link, EDSA: enhanced directional self-assembly.

times) that can be achieved by a robot team implementing supervised morphogenesis in the context of a simulated gap-crossing task. The results are presented in Section 5.4. In Section 5.5, we review existing air-ground robot teams and the tasks they solve. We summarize the contributions of this chapter in Section 5.6.

5.1 Control methodology

We present the methodology underlying supervised morphogenesis. See Figure 5.1 for an overview. As shown in the figure, supervised morphogenesis is composed of multiple control states (shown in circles) for each robot type. We assume that only one aerial robot has been allocated [99] to provide assistance to a particular group of ground-based robots. In the following, we describe each state starting with those of the aerial robot.

5.1.1 Aerial robot

Aerial robot exploits its agility to follow the group of robots it is allocated to supervise. Simultaneously, it uses a standard downward-pointing camera — a sensor available on most aerial platforms — to build an internal model of the environment underneath. Depending on the task, the dimensionality of the built model may vary between 2D or 3D. This ability to choose the dimensionality of the model allows the

aerial robot to avoid unnecessary data collection and computation. For example, the solution of certain tasks only require the relative positions of the robots and the objects on the ground to be represented using a 2D model (see our first case study) while other tasks depend on a detailed three-dimensional model of the environment (see our second case study).

Subsequently, in a further control state, we let the aerial robot use this internal model of the environment to perform on-board simulations to determine whether self-assembly is required on the ground to maneuver through the environment or to solve the task at hand. Such on-board simulations allow the aerial robot to assess the adequacy of different target morphologies in a particular environment prior to their costly (in terms of energy and time) and physical realization on the ground. The aerial robot continues with the modeling of the environment if simulation outcome requires no intervention.

On the other hand, if supervision is required, the action taken by the aerial robot is a combination of different techniques. First, a one-to-one spatially targeted communication link is established to a robot that can initiate self-assembly. The selection (and therefore the location) of this robot may depend on the task and/or the environment. I required, the established link is expanded using the iterative growth process to allocate further robots to join the self-assembly process. In this manner, resources are allocated optimally as non-selected robots become free to purse other tasks. Second, the aerial robot follows up an established STC link in a separate state in which it selects the target morphology and broadcasts its description using a SWARMORPH-script. Before transmitting the SWARMORPH-script, it is first translated into a binary string allowing it to be transmitted via wireless Ethernet or via the low-bandwidth communication modality based on LEDs and cameras. Once the transmission is complete, the control of the aerial robot returns to the environment model building state.

5.1.2 Self-assembling robots

In most tasks, the control of a self-assembling robot is initiated with a behavior that does not require immediate supervision from an aerial robot. We refer to these behaviors as basic behaviors that may include both perceptual and decision-making processes. For instance, in a basic obstacle avoidance behavior, data need to be acquired from proximity sensors and processed before a decision can be made on the robot's direction of movement. Further examples of basic behaviors include phototaxis and EDSA presented in Section 3.1. The total number of basic behaviors executed, the parameters used to configure them, and the manner in which they are combined may depend on the actual task at hand.

Basic behaviors are executed by self-assembling robots until a `STA` signal is perceived from the aerial robot trying to establish an STC link (as described in Chapter 4) and therefore highlighting the need for supervision. Self-assembling robots able to perceive the signal come to a halt and acknowledge participation in the iterative elimination process by responding with the `ACK` signal. Messages from the aerial robot can either be transmitted via wireless Ethernet or via LED colors that can be detected using ceiling cameras. The robots on the ground are required to use a situated communication modality to communicate with the aerial robots in order to facilitate spatially targeted communication. In case no STC link has been established, a self-assembling robot continues with the execution of basic behaviors under the assumption that it is not required for the formation of the target morphology

selected by the aerial robot.

Robots selected by the aerial robot wait for the reception of self-assembly instructions from the aerial robot. These instructions are expected in form of a SWARMORPH-script that lead to the formation of a morphology most suited for the task. Once received, the connections described in the SWARMORPH-script are formed using EDSA — a basic behavior based on recruitment and guidance algorithms that do not depend on further assistance from the aerial robot for execution.

5.2 Case study number 1

We present a preliminary case study of supervised morphogenesis. We show how established STC links can be followed up with actual information relevant to the task at hand. We demonstrate how different states in the control methodology function seamlessly with each other and across robot platforms.

5.2.1 Task and experimental setup

In the task we consider, ground-based robots depend on aerial assistance to be successful. A successfully solved task requires the formation of a target morphology at a distance between 60 cm and 70 cm from a point-of-interest (POI) in the environment. The POI is a light source placed on the ground that can be detected by an aerial robot while it remains undetectable to the ground-based robots due to their vantage point and the intense ambient lighting. While the shape of the target morphology can be freely chosen by the aerial robot from a pre-compiled morphology library consisting of many morphologies of different shapes and sizes, its size is task-dependent. That is, the task requires the formation of a morphology that leaves at least three ground-based robot unconnected at all times and therefore free to pursue other, more important, tasks. The total number of deployed robots is not made available to the robot team. The ground-based robots are given no information about the task. Therefore, in order for the robot team to be successful, the aerial robot needs to acquire following three bits of information:

1. the location of the POI and all ground-based robots
2. the relative distances between the POI and the robots on the ground
3. the total number of ground-based robots deployed.

We set up an experiment in which we consider an heterogeneous team composed of one AR.Drone and six foot-bots. We deploy the robots on a mission towards a task represented by the POI. The AR.Drone flies in advance to the POI and hovers until the foot-bots follow. Simultaneously, it builds a two-dimensional model of the environment using its downward-pointing camera. For safety reasons, we used a colorless, transparent plexiglass platform installed at 40 cm height from the ground to shield the foot-bots from potential emergency landings of the AR.Drone (see Figure 5.2a). The AR.Drone analyzes the images returned by its downward-pointing camera to locate the POI by detecting the point with the highest light intensity above the thresholds that can be reached by the foot-bot LEDs. To detect the foot-bots, standard circle detection algorithms are executed. Using these information, the AR.Drone builds a two-dimensional environment model containing X/Y coordinates of the POI and the foot-bots in its field of view. Subsequently

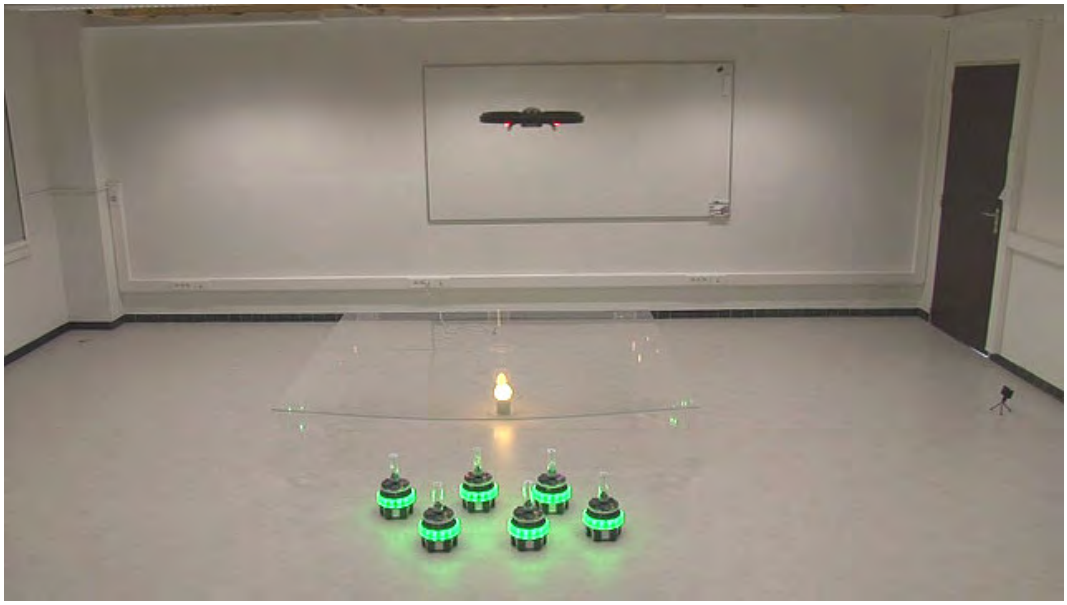
executed on-board simulations continuously compute the relative distances between the POI and the foot-bots. All communication from the AR.Drone to the foot-bots occurs over wireless Ethernet. Foot-bots, in turn, use their LEDs illuminated in RGB colors to send different messages. Other parameters used during the STC link establishment process are identical to those described in Section 4.1.3.

5.2.2 Results

We carried out 10 experimental runs using the heterogeneous team composed of one AR.Drone and six foot-bots. In 7 of the 10 runs, the team managed to successfully complete the task by forming target morphologies of the appropriate size in the distance range required. In the remaining three runs, however, task conditions were not met as only fewer than three robots remained unconnected after self-assembly. This is an immediate result of the low-resolution downward-pointing camera (176x144) available to the AR.Drone that led to the incorrect detection of more than six deployed ground-based robots. One solution to avoid the occurrence of such faulty runs may be to fine tune the image analysis parameters prior to each run considering the battery charge of the robot (we observed that a lower battery charge had a negative effect on the quality of the image processing routines). Alternatively, one could consider replacing the AR.Drone camera with a camera capable of capturing images with higher resolution or deploying another aerial robot that already comes with a higher resolution camera (see Section 5.3). Video footage of one of the successful s can be seen online at `http://iridia.ulb.ac.be/supp/IridiaSupp2017-007/`.

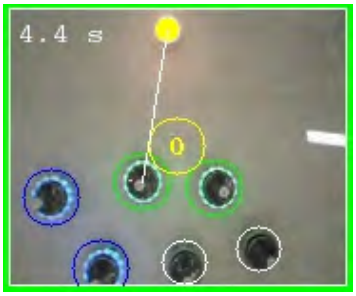
Here we describe a successful run and present snapshots from the experiment in Figures 5.2b to 5.2e. Initially, the foot-bots are placed facing the POI with their LEDs illuminating the color green. On deployment, the foot-bots are instructed to drive forward while the AR.Drone flies over them at an altitude of approximately 1.8 m while modeling the environment underneath. Once the POI has entered its field of view, the AR.Drone runs an hovering behavior (see Section 4.1.3) to remain above the POI and starts running on-board simulations. As soon as the outcome of the simulations confirm a relative distance between any one of the foot-bots and the POI between 60 cm and 70 cm, the AR.Drone broadcasts a SWARMORPH-script with a *Stop* command causing all foot-bots to come to a halt (see Figure 5.2b). Then, to estimate the total number of foot-bots in the group, the AR.Drone scans the vicinity around the already detected foot-bots. In this manner, the size of the target morphology (i.e., total number of foot-bots minus three) can be determined. In the control state that follows, the AR.Drone initiates the iterative elimination process to establish an STC link to the closest foot-bot (indicated using a straight line in Figure 5.2c) to the POI. Once the one-to-one STC link is established, as shown in Figure 5.2d, a SWARMORPH-script of the correct size is (in this case of size three) is chosen from the morphology library and then transmitted to the selected foot-bot using wireless Ethernet broadcast. Although all foot-bots are able to receive the SWARMORPH-script, only the selected foot-bot executes the self-assembly instructions and form and arrow-like morphology of size three using the recruitment and guidance algorithms introduced as part of EDSA (see Figure 5.2e). In this manner, the foot-bots were able to build the target morphology of appropriate size without any prior knowledge about the task.

Note that a SWARMORPH-script is first converted into a binary string before transmission. A robot that receives such a binary string can, in turn, translate it back to a SWARMORPH-script and execute the received control logic. By



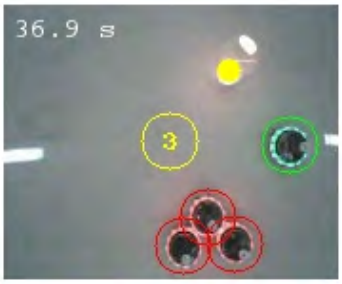
(a)

(b)



(c)

(d)



(e)

Figure 5.2: Supervised morphogenesis case study number 1. (a) The experimental setup showing the heterogeneous robot team composed of one AR.Drone and six foot-bots. The light source is placed on the ground covered by a transparent plexiglass platform. Figures (b-d) show images taken from the downward-pointing AR.Drone camera after processing. The border colors represent the signals transmitted during the iterative elimination process. The number in the center of each frame shows the size of the target morphology determined using on-board simulations run by the AR.Drone. Figure (e) shows the target morphology formed using EDSA. The annotated time duration in Figures (b-e) corresponds to the clock time starting the execution of ESTC.

converting to binary strings, SWARMORPH-scripts can also be transmitted using low-bandwidth communication modalities such as LEDs and cameras. The aerial robot was flown manually in this preliminary case study. However, the control (environment modeling, on-board simulations) and communication algorithms (ESTC, transmission of self-assembly instructions) were executed entirely autonomous by the AR.Drone. The foot-bots remained autonomous throughout the course of all experiments. Further examples of cooperation between an AR.Drone and foot-bots are presented in [100].

5.3 Case study number 2

In the second case study, we consider a more challenging hill-crossing task in which we test the adaptivity of supervised morphogenesis to changing environmental conditions. Contrary to the task considered in Section 5.2, a successful task solution requires the aerial robot to build a three-dimensional model to characterize the environment. Additionally, more sophisticated simulations are required to detect the obstacle. The obstacle is an elevated surface, hereafter referred to as *hill obstacle*, that needs to be crossed by a group of five foot-bots. We further test the feasibility of our approach under minimal conditions by limiting the communication between the robot types to low-bandwidth LEDs and camera-based communication. The solution of the task also requires the formation of multiple target morphologies – rather than a single one as in the previous case study – and the selection of separate foot-bot groups for their formation. We use the eye-bot as our aerial robot.

5.3.1 Task and experimental setup

The heterogeneous robot team consisting of an eye-bot and five foot-bots is given the task to maneuver towards a light source from a deployment area where they are initially placed (see Figure 5.3a–c). The foot-bots use their light sensors to detect the light source and drive in an almost straight path to reach their destination. As shown in Figure 5.3d, we place a steep hill obstacle in their path of which we vary the steepness between $0\,^\circ$ (i.e., the obstacle is absent) and $30\,^\circ$. A foot-bot can drive over a hill obstacle of up to $25\,^\circ$ inclination without toppling over. If the steepness exceeds this limit, the foot-bot has to self-assemble into bigger morphologies that provide the physical stability required to cross the hill without toppling over. Neither the presence of the hill obstacle nor its steepness can be detected by the foot-bots. They also do not know the total size of the group in order to determine the shape and size of target morphologies. Therefore, the foot-bots depend on the eye-bot to provide the supervision necessary to successfully solve the task. A successful task completion in this scenario requires the eye-bot to detect the hazardous environment and compute the total number of target morphologies and supervise their formation such that the safe-crossing of all foot-bots is guaranteed. The foot-bots are preloaded with a morphology library composed of several SWARMORPH-scripts from which particular scripts (hence particular target morphologies) can be chosen by the eye-bot using a pre-defined signal. The eye-bot is assumed to have flown in advance and positioned itself (by attaching to the ceiling in indoor environments or otherwise by hovering) above the hazardous area at a height of 2.96 m immediately over the hill obstacle.

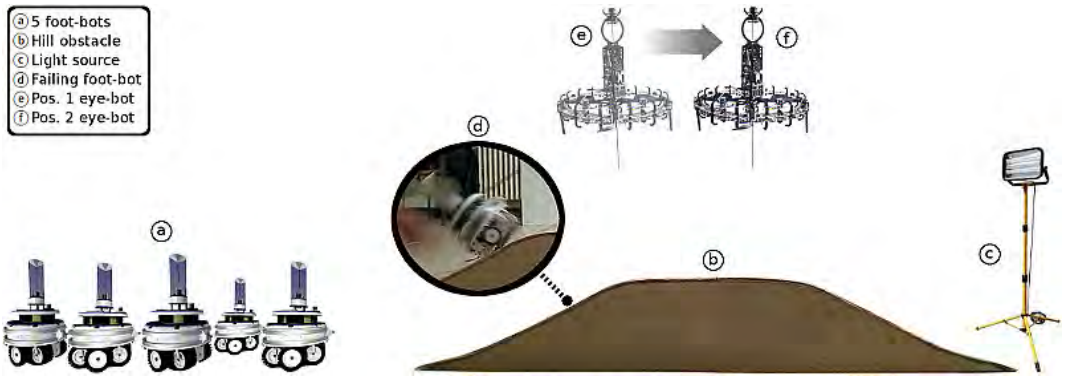


Figure 5.3: The experimental setup of the hill-crossing task. Five foot-bots are shown in the deployment area and one light source in the destination area. A hill obstacle of 30 ° is shown between both areas. Visualized are also two positions above the hill obstacle the eye-bot considers when using its monocular vision system to build three-dimensional environment models.

5.3.2 3D environment modeling using heightmaps

The task considered in this case study requires the aerial robot to have a three-dimensional model of the surface on which the foot-bots operate. The eye-bot is given no maps or prior information about the environment. Therefore, it computes heightmaps (sometimes referred to as depth maps) of the surface and the objects underneath it during task execution. A heightmap is a three-dimensional representation of the environment (in a sensor's field of view) based on a matrix data structure containing surface elevation data in cm. We considered two different methods for the aerial robot to obtain heightmaps; one using a comparatively lightweight method available to most aerial platforms, and another one based on a dedicated sensing hardware for aerial robots not subject to limited payload capacities. In Appendix A, we present details of how the eye-bot computes heightmaps based on stereo images retrieved from two different positions (as shown in the experimental setup in Figure 5.3) using its downward-pointing monocular camera. We also use Microsoft Kinect sensor to acquire precomputed heightmaps and provide a quantitative and a qualitative comparison between the two methods.

5.3.3 Decision-making based on height profiles

Independent of the method used to compute heightmaps, we assume that the eye-bot has access to a matrix of size 640×480 with the surface elevation in cm for the surface underneath. This information about the environment is used by the eye-bot to perform on-board simulations and determine whether individual foot-bots will be able to drive towards the light source safely. If not, decision will be made to initiate self-assembly on the ground and assist the formation of suitable morphologies (depending on the size of the group) that offer the physical stability and a safe passage to the foot-bots. Here we describe how this decision is made based on height profiles retrieved from heightmaps.

A height profile represents the elevation along the estimated trajectory of a foot-bot to the light source. For each newly acquired heightmap, the eye-bot extracts the height profile for each foot-bot by reading out the cell values along the foot-bot's estimated trajectory in the heightmap. The estimated trajectory of each foot-bot is

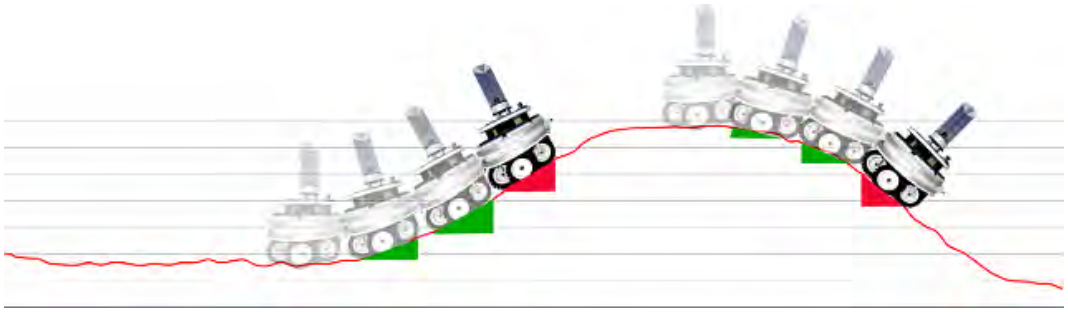


Figure 5.4: Visualization of an on-board simulation run of a foot-bot along its estimated trajectory (from left to right). The height profile is extracted from a heightmap computed using stereo images retrieved from the eye-bot camera. The foot-bot is shown in eight different segments along its estimated trajectory. In two segments, a too steep inclination is detected (marked in red) rendering the whole trajectory too dangerous for individual foot-bots. Inclinations lower than 25 ° is marked in green.

assumed to be a straight line connecting the foot-bot's current position with the light source. Note that while each foot-bot's position is estimated using circle detection algorithms executed on the image returned by its downward-pointing camera, the light source is assumed to be in the center of the area in the image with the highest light intensity.

Based on retrieved height profiles, the eye-bot simulates virtual navigation of each foot-bot along its estimated trajectory. Each foot-bot is virtually driven towards the light source by moving it pixel-by-pixel allowing the eye-bot to estimate the stability a foot-bot may experience on its path. More precisely, the foot-bot is placed at its currently detected position on the height profile and its inclination at this particular trajectory segment is calculated before it is moved by a pixel. That is, the inclination of the surface underneath the front and the rear of the simulated foot-bot's chassis is calculated. A simulation run ends when all foot-bot's chassis reach the light source or when a calculated inclination is higher than 25 °, the angle an individual foot-bot can endure without toppling over. When a too dangerous inclination is detected in simulation, the eye-bot stops running on-board simulations and issues the `STA` signal using the color red displayed on its LEDs to warn the foot-bots underneath and to supervise the formation of target morphologies. Visualization of a simulation run is shown in Figure 5.4 including a height profile extracted from a heightmap calculated using stereo images.

5.3.4 Results

We conducted three sets of experiments each with 10 experimental runs. First, we considered no obstacle. Second, we considered a hill obstacle with a gentle slope of 12 ° that can be crossed by individual foot-bots without toppling over. Third, we considered a steep hill obstacle too dangerous for individual foot-bots to cross. From the resulting 30 experimental runs, the robot team successfully solved the task in 27 runs. That is, in the first and second sets up experiments, the eye-bot correctly estimated the surface to be safe for foot-bots and did not intervene. As a result, all foot-bots reached their destination. In the third set where self-assembly is required to solve the task, only 7 out of 10 runs were successful. From the remaining 3 runs,

2 failed because of hardware damages in the docking mechanism caused by too high torque when driving over the hill obstacle. These damages led to the breaking of physical connections causing individual foot-bots to topple over. A third run failed because of low battery charge of a foot-bot. Note that these failed runs did not result from flaws in our presented approach but rather are related to hardware issues.

In Figure 5.5, we present screenshots of a successful experimental run from the third set and describe details. We manually moved the eye-bot between two positions 30 cm away from each other above the hill obstacle. Ten images were taken from each position. Resulting 20 images used to compute 10 different height maps of the environment. All computed heightmaps are considered in the simulations executed by the eye-bots, or as in this particular experimental run, simulations were stopped when the eye-bot's belief β of an hazardous environment is exceeds beyond 90%. After the virtual navigation of each foot-bot in its field of view, the eye-bot updates β using the following filtering method: $\beta = (1-c) \cdot \beta_{h-1} + c \cdot \beta_h$, where β_h is a binary value representing the outcome of the simulation (where 0=no danger and 1=danger) and c, $0 \leq c \leq 1$, is the confidence level of the eye-bot in the precision of the underlying heightmap. Based on empirical evidence, we determined $c = 0.85$ to be appropriate for heightmaps computed using stereo images and $c = 0.9$ for heightmaps retrieved from the Kinect sensor. This filtering method makes simulations less vulnerable to extreme outliers and has a smoothening effect of the ground surface. In this experimental run, the average value for the maximum inclination computed was 29.12 ° with a standard deviation of 2.88°. The hazardous environment was detected on the basis of ten simulated drives in total (i.e., two for each foot-bot).

While individual height profiles may contain incorrect height estimates for individual pixels, the results of our experiments show that, on average, environment models based on heightmaps closely corresponds to reality (where the maximum inclination was 30 °) and hence can be used for decision-making. The simulations were able to reliably identify the too steep hill obstacle as hazardous for foot-bots, as the low standard deviation indicates. However, the decision-making process may need to assume a more defensive threshold inclination such as 20 ° (as opposed to the 25 °) an individual foot-bot can withstand without toppling over. Such a defensive threshold may, on the one hand, allow the aerial robot to be more tolerant with the spatial assumptions and, on the other hand, may help avoid undetected hazardous environments. A drawback of this defensive approach,however, is the fact that self-assembly may be unnecessarily initiated when they were not actually required in reality.

The foot-bot controllers are initiated (see Figure 5.5a) with a phototaxis behavior which uses the data retrieved from their light sensors to determine the movement direction. Simultaneously, they process the images returned by the upward-pointing camera to detect potential warning signals (i.e., the red color) issued by the eye-bot. The eye-bot is stationary above the hill obstacle running on-board simulations. As the foot-bot group reaches the hill obstacle (see Figure 5.5b) the eye-bot warns the foot-bots using the `STA` signal (i.e., the red color. All foot-bots halt. This is either because all foot-bots were able to perceive the signal sent by the eye-bot or because at least one foot-bot has detected the signal and has relayed it to other foot-bots using the mxRAB communication device. A static foot-bot responds to the eye-bot using the color blue indicating availability to be supervised. Subsequently, an STC link is established within 16 s to the foot-bot situated in the center (relative

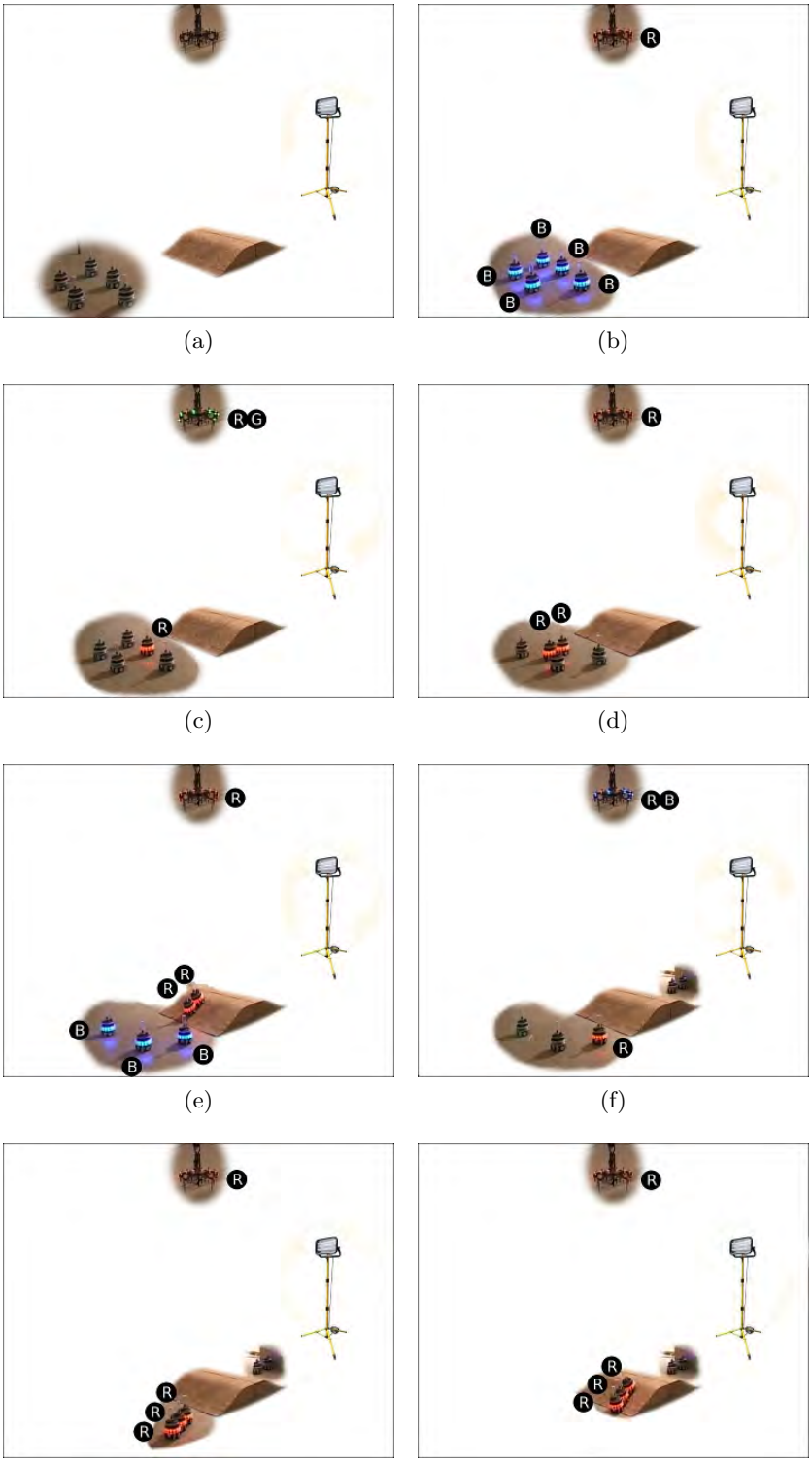


Figure 5.5: Snapshots of an hill-crossing experiment. For the sake of better visibility, background clutter has been edited out. The signals transmitted by each robot type during different mission phases is indicated as follows: R=RED, G=GREEN, B=BLUE, RG=RED-GREEN, and RB=RED-BLUE. See text for details.

to the hill) and is closest to the light source. By choosing this particular foot-bot, the eye-bot makes sure that the morphology is formed away from the two edges of the hill obstacle while also allowing completed target morphologies to drive over the hill obstacle without having to avoid collisions with free foot-bots along the way. The STC link is followed up by the eye-bot by sending the RED-BLUE signal (see Figure 5.5c) to initiate the formation of a chain morphology of size two. This target morphology was formed within 6 s (see Figure 5.5d) and managed to cross the hill obstacle successfully with the next 12 s. In the meantime, the eye-bot establishes another STC link to a second foot-bot (see Figure 5.5d and f) and issues a RED-GREEN signal to invoke the formation of a chain morphology of size three. The formation of this target morphology lasted 15 s (see Figure 5.5g) and its successful crossing took another 11 s (see Figure 5.5h). The total duration of the experiment was 70 s. Note that the target morphologies were autonomously chosen by the eye-bot. An alternative solution would have been to form one chain morphology of size five. This option was not considered by the eye-bot as long structures cause hardware damages to the docking mechanism when different foot-bots part of the same morphology simultaneously experience opposing inclinations on the hill obstacle. The decision regarding how many target morphologies are required to be formed depends on the total number of foot-bots that are deployed and have acknowledged the issued warning. The eye-bot monitors morphology growth progress by counting the number of foot-bots illuminating the color red (meaning they are attached and part of a collective morphology). Target morphologies are iteratively formed under the eye-bot's supervision until no foot-bot is visible in the eye-bot's field of view. Hence, this control logic can be applied to foot-bot groups of any size. Video footage of the experiment can be found online `http://iridia.ulb.ac.be/supp/IridiaSupp2017-007/`.

5.4 Quantifying performance benefits

We quantify the performance benefits (w.r.t task completion time) that can be achieved by a group of self-assembling robots when cooperating with an aerial robot through supervised morphogenesis. We define a task that can be solved by ground-based robots with or without aerial supervision. Besides the control methodology presented in Section 5.1, we developed two further control methodologies to study both the benefits of aerial supervision compared to a solution without supervision and then isolate the benefits of location-based supervision – a core feature of the approach presented in this thesis. We first present the task and experimental setup followed by a description of all three control methodologies. Then, we present the results we obtained from simulation-based studies [15]. We considered an heterogeneous team composed of an eye-bot and multiple foot-bots. Videos of these experiments are available online at `http://iridia.ulb.ac.be/supp/IridiaSupp2017-007/`.

5.4.1 Task and experimental setup

We consider a gap-crossing task (with variable gap widths) for foot-bots. The experimental setup shown in Figure 5.6. The heterogeneous robot team is deployed in an environment consisting of a start zone, a target zone, a gap that separates the two zones, and a light source placed in the target zone. At the beginning of each

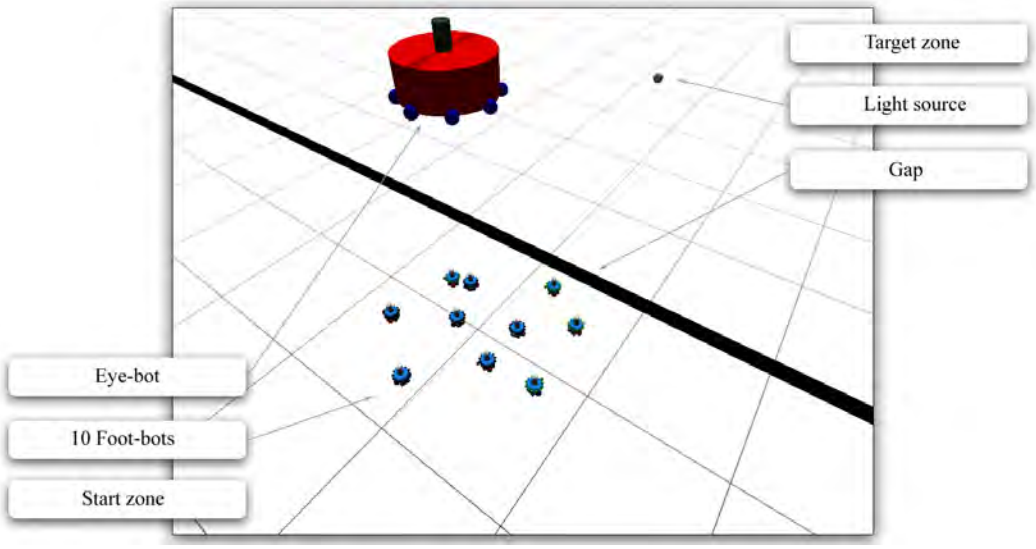


Figure 5.6: A snapshot from the simulation showing the heterogeneous team and the environment it operates in. The gap separates the environment into a start zone and a target zone with a light source. An eye-bot and ten foot-bots are visible in the start zone.

experiment, 10 foot-bots are randomly placed in the start zone and are required to perform a phototaxis behavior. They use their light sensors to determine the direction in which the need to drive and their ground sensors to detect possible gaps and avoid falling into it. The foot-bots are deployed within a square area of 2 m $\times$ 2 m – an area that can be entirely captured in the field of view of an eye-bot hovering above the foot-bots at 2.2 m altitude. The eye-bot uses its pan-and-tilt camera and the on-board image processing software to estimate the width of the gap. To reach the target zone, the foot-bots may need to connect to each other to form a chain morphology, depending on the width of the gap. Note that the minimal length of such a chain morphology (i.e., the number of foot-bots in the chain) that guarantees a safe crossing of the gap depends on the width of the gap (if a morphology with an inappropriate size attempts to cross an overly wide gap, it will fall in and fail). We use gaps of widths between 5 cm, 10 cm, 15 cm and 25 cm. In case of a 5 cm wide gap, individual foot-bots are able to drive over without cooperating with the eye-bot or other foot-bots. For all other widths, the foot-bots are required to form a chain morphology of 2, 3 and 4 foot-bots respectively to be successful. The task is considered to be completed when the final foot-bot of the first chain morphology has crossed the gap and has successfully disconnected from the chain. Foot-bots that do not get selected by the eye-bot continue with another basic behavior, in this case an anti-phototaxis behavior.

5.4.2 Control methodologies

Figure 5.7 visualizes the main control states and the order in which they are executed in the three control methodologies we developed to study the performance benefits of supervised morphogenesis. Communication between the two robot types are handled via the LEDs and cameras-based communication modality which provides the user with a visual feedback of ongoing activities. Below, we provide a detailed description

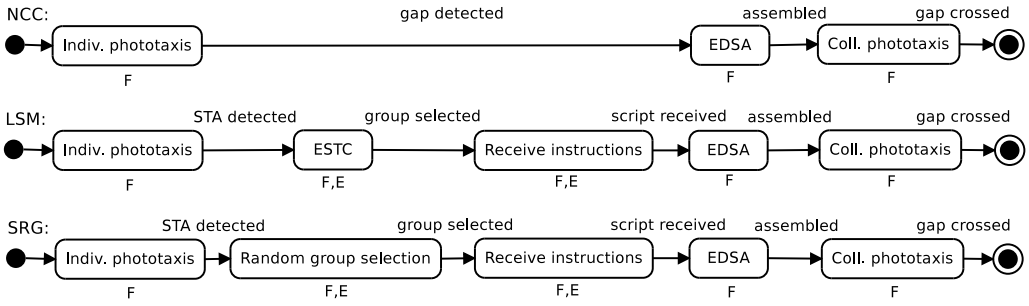


Figure 5.7: Decomposition of foot-bot control methodologies into control states. States only involving foot-bots are marked 'F', states involving foot-bot eye-bot cooperation are marked 'F.E'. i) NCC: non-cooperative control, ii) LSM: location-based supervised morphogenesis and iii) SRG: supervision based on random group. NB "Indiv. phototaxis" = "individual phototaxis", "Coll. phototaxis" = collective phototaxis, "ESTC" = "establishing spatially targeted communication", "EDSA" = "enhanced directional self-assembly". Both phototaxis behaviors are "basic behaviors" shown in Figure 5.1.

of each control methodology.

5.4.2.1 Non-cooperative control (NCC)

When executing this control methodology, the foot-bots do not seek for supervision from the eye-bot to solve the task. This methodology is an implementation of the work presented in [101]. The foot-bots are pre-loaded with a SWARMORPH-script that they use to form a four foot-bot chain morphology when a gap (regardless of its actual width) is encountered. They initially execute a phototaxis behavior to drive towards the light source until one of the foot-bots detects the gap using its ground sensors. This foot-bot retreats approximately 40 cm from the gap and warns its peers via a message (*Stop* command) sent over the mxRAB device causing the all other foot-bots to come a halt (while remaining available for recruitment). Note that halted foot-bots still execute an obstacle behavior in order avoid collisions with a retreating foot-bot. Subsequently, the foot-bot that had detected the gap initiates self-assembly and forms a chain morphology composed of four foot-bots. Once the chain of four foot-bots is formed, the morphology executes a collective phototaxis behavior towards the light source to cross the gap. NCC allows us to isolate the performance benefits of aerial supervision.

5.4.2.2 Location-based supervised morphogenesis (LSM)

LSM is the implementation of supervised morphogenesis as presented in this thesis. The foot-bots execute a phototaxis behavior while looking for messages (and therefore supervision) from the eye-bot. They are given no knowledge of the environment or the task. The eye-bot sends a `STA` message to establish a one-to-one STC link if a gap wider than 5 cm is detected. The foot-bot to which the link is established will be located approximately 40 cm away from the gap and is intended to initiate

self-assembly. A foot-bot that detects a message from the eye-bot, warns its peers via a message (*Stop* command) sent over the mxRAB device. Subsequently, the one-to-one STC link is expanded to include a certain number of immediate neighbors required for the formation of the target morphology. The selected foot-bots receive self-assembly instructions in the form of a SWARMORPH-script (a binary string thereof) from the eye-bot. The foot-bots follow the instructions in the script to self-assemble into the target morphology most adequate to cross the gap. Once the morphology is formed, the foot-bots execute a collective phototaxis behavior to cross the gap. Non-selected foot-bots move away from the light source. LSM represents the baseline against which the performance of the other two methodologies are measured.

5.4.2.3 Supervision based on random group (SRG)

This control methodology allows us to isolate the performance benefits of selecting the robots based on their location and their mutual proximity to each other to form the target morphology. SRG is identical to LSM, except for the selection of foot-bots to form the target morphology. After establishing a one-to-one STC link to a randomly located foot-bot, the eye-bot avoids the execution of iterative growth process that would have selected immediate neighbors of the self-assembly initiating foot-bots. Instead, the eye-bot repeats the iterative elimination process to select further foot-bots from the group independent of their location in the environment. This approach mirrors potential solutions that could be applied in systems that consider unique IDs of the robots. In such systems, for instance, the aerial robots may broadcast self-assembly instructions together with robots IDs that should form the target morphology while ignoring their location in the environment entirely.

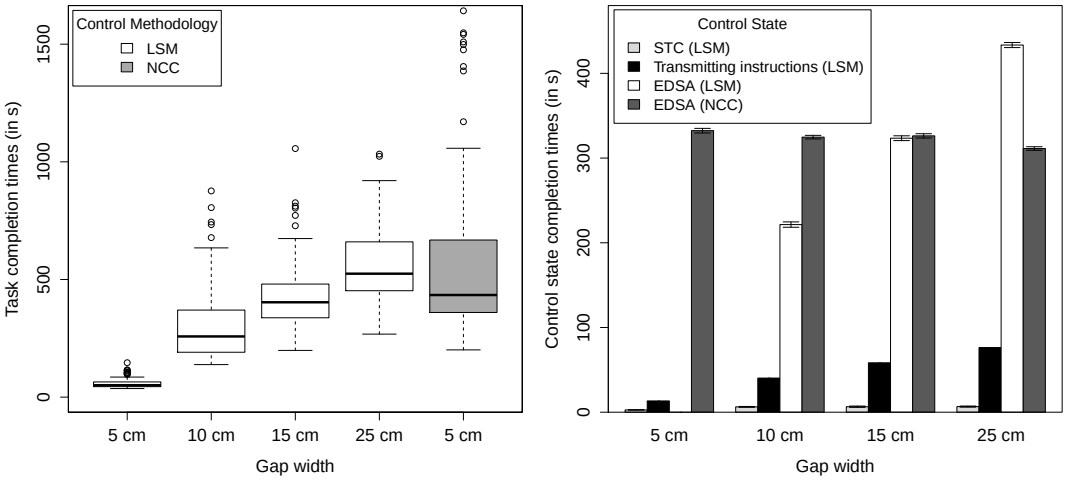
5.4.3 Experiments and results

We ran 100 simulation runs for each combination of gap width and control methodology (i.e., $4\times3\times100=1200$ runs in total). We first study the benefits of supervision by the eye-bot and then isolate the benefits of location-based foot-bot selection.

5.4.3.1 NCC vs. LSM

To analyze the performance benefits the foot-bots can achieve through cooperating with an eye-bot, we compare the task execution times of NCC with those of LSM. The results are shown in Figure 5.8a. We have only plotted the results of the narrowest gap of 5 cm for NCC, as the task completion times between the various gap widths did not prove to be significantly different. This is a direct consequence of the fact the foot-bots executing this methodology formed chain morphologies of the same size regardless of the gap width they encountered.

In all the experiments, the foot-bots solved the task. According to the results in Figure 5.8a, the median task completion times of LSM are 51, 259 and 403 seconds for the widths 5 cm, 10 cm and 15 cm, respectively. Compared to the median task completion time of NCC (434 seconds), the mean completion times for LSM were respectively 88%, 40% and 7% lower in environments with gaps that can be crossed by an individual or chains composed of two or three foot-bots. This is due to the fact that in LSM, the length of the chain is chosen based on the gap width. The supervision provided by the eye-bot avoids the inclusion of excess foot-bots in the



(a) (b)

Figure 5.8: (a) Box-and-whisker plot showing task completion times (in seconds) of LSM and NCC for four different gap widths. For the NCC methodology, results of only 5 cm are plotted as the results do not differ significantly from other gap widths. (b) Bar-plot showing a breakdown of the time spent by the foot-bots executing different control states. Bars are decorated with the standard deviation, except for the control state "transmitting instructions" that is bound to constant time dependent only of the transmitted SWARMORPH-script (i.e., the gap width).

morphology requiring additional time for the formation of the target morphology. In the case of the widest gap (i.e., 25 cm) that can only be crossed by four or more physically connected foot-bots, NCC is, in general, faster than LSM. Intuitively, this could have been expected given that both control methodologies (i.e., LSM and NCC) form a chain of four foot-bots close to the gap, but in the case of LSM, self-assembly instructions need to be first transmitted from the eye-bot to the foot-bots before the self-assembly process can start. However, the NCC methodology has several outlier runs that take very long to complete. This is because in NCC, the foot-bots that become part of the target morphology are not pre-selected by the eye-bot before they reach the gap. Hence, non-connected foot-bots can cause (sometimes severe) physical interference with ongoing self-assembly processes or with moving target morphologies. Both interferences delay task completion times.

In Figure 5.8b, we present a breakdown of how much time is spent by the foot-bots executing each control state. As the results show, the wider the gap, the more time is spent by the robots transmitting self-assembly instructions. This is due to the fact that the length of the SWARMORPH-script describing the target morphology is linearly proportional to the size of the target morphology. However, this communication overhead part of LSM would become negligible if a communication modality with higher bandwidth (such as wireless Ethernet) is used for communication. The results also show that when a target morphology composed of four foot-bots is formed, the self-assembly process in LSM requires on average 39% more time than that of NCC. This can be explained by the fact that in NCC all foot-bots are available for forming a connection during the morphology growth process which increases the chances of a foot-bot being located close to where a connection is required causing connections to be formed faster. On the other hand, LSM allocates

resources optimally by selecting precisely the number of foot-bots needed for the target morphology and freeing up the rest of the team for other tasks. The decision involving this trade-off between faster target morphology formation times and optimal resource allocation may depend on the task and mission priorities.

5.4.3.2 SRG vs. LSM

We study the difference in target group selection time between both control methodologies by considering various values of N and D. For N, we considered $N = 10$ and three further values $N := \{20, 30, 40\}$ to detect potential performance differences in larger groups. Furthermore, we considered four relative sizes for $D := \{20\%, 40\%, 60\%, 80\%\}$. For each combination of N and D, we executed 100 simulations runs and recorded the average number of iterations that was required to select the target group. The results are plotted using fixed grids in Figures 5.9a and 5.9b, for SRG and LSM respectively. As the results show, the average number of iterations increases in SRG by a factor of 5 as N increases from 10 to 40, independent of the D value. Simultaneously, for increasing D, the average number of iterations increases by a factor of 3.5. In the worst case, i.e., when comparing $N = 10$ and $D = 20\%$ with $N = 40$ and $D = 80\%$, the value increases by a factor of 19. By contrast, LSM yields lower averages independent of the values considered for N and D indicating a much faster and scalable target group selection process. In the worst case, the average for LSM increases from 19 to 28, representing an increase by a factor of 1.5. These empirical observations correspond closely to the model predictions analyzed in Section 4.2.2.3 and plotted in Figure 4.12.

To further study performance differences in self-assembly, we discount the time spent for target group selection and plot task completion times in Figure 5.9c. As the results show, LSM is on average faster than SRG in all cases studied independent of the width of the gap. The explanation for these results is that a target morphology formed next to the gap by involving nearby foot-bots in most cases requires less time to finish the formation, and then reaches and crosses the gap faster than a morphology formed at a random place with peer foot-bots joining from random places in the environment. We expect that this difference in terms of task completion time would be even greater for larger start zones as the distances between randomly selected foot-bots and the gap may be larger.

We further studied the difference in task completion times between LSM and SRG by conducting additional experiments in a walled environment where we surrounded the foot-bots in the start zone by walls on three sides to adjoin the gap on the fourth side. We simulated an indoor environment similar to factory halls that contain static objects, goods or static machinery. We observed that the presence of the walls had no significant impact on the task completion time in LSM. For SRG, on the other hand, the presence of walls had a significant negative impact on performance. When the randomly selected self-assembly initiating robot happened to be located close to one of the walls, it was difficult, and at times even impossible, for foot-bots to physically connect to connection seeking foot-bots. As a result, successful task completion only occurred in 13% (chain size 2), 29% (chain size 3) and 34% (chain size 4) of the experiments using the SRG methodology.

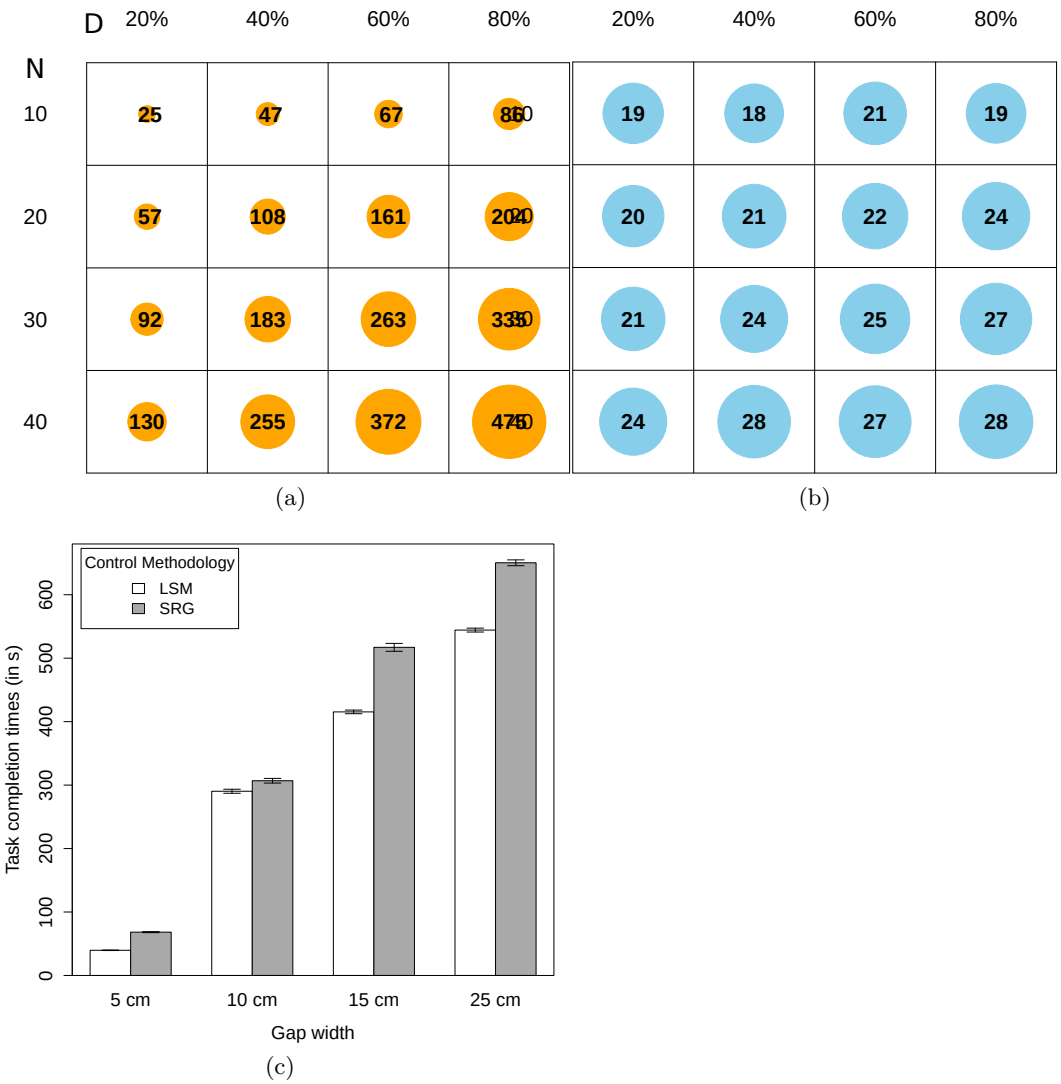


Figure 5.9: Contrasting performance benefits of control methodologies SRG and LSM. (a) Average number of iterations required for target group selection only based on iterative elimination process as implemented in SRG. (b) Average number of iterations required for target group selection based on iterative elimination and growth process as implemented in LSM. (a,b) The values in the grids are average values resulting from 100 simulation runs and are also visualized using colored circles. The diameter of each circle is relative to other values within the particular grid and represents a visual comparison aid. (c) Task completion times (in seconds) of LSM and SRG minus the time taken for target group selection. Bars are appended with the standard deviation.

5.5 Related work

In this section, we review existing (i.e., not conceptual or simulated) heterogeneous robotic systems with a focus on air-ground teams and the tasks they solve. We also present control frameworks developed particularly for the control and coordination of such heterogeneous teams. Robotic systems that use technologies or solve tasks similar to those considered in this chapter are also presented.

Air-ground teams are often limited to one terrestrial and one aerial robot. This is mainly because they have relatively low overhead (in terms of coordination and communication) due to the limited number of team members. Such two-robot teams have nevertheless been successfully applied to solve various tasks including the building of indoor maps [102, 103], the mapping of obstacles in large areas [104], and cooperative tracking of moving targets [105]. Aerial images have also been used by ground-based robots for semantic mapping [106] and navigation among moving obstacles [107]. In another system [108], the ground-based robot provides the aerial robot with the visual aid and platform necessary for take off, following a moving target, and landing. A localization method for ground-based robots based on elevation maps is presented in [109]. The method does not dependent on GPS or on sensors. Instead, the terrestrial robot finds its relative position and orientation within a reference map provided by the aerial robot.

Heterogeneous robot teams composed of more than two members have also been applied to various application scenarios [25, 110, 111]. In [25], a heterogeneous team is proposed that considers a climbing robot besides aerial and ground-based robots. This team is therefore able to explore vertical surfaces and manipulate objects unreachable for both terrestrial and aerial robots. Twenty robots are shown to solve a search-and-retrieve task in which the team combines the capabilities of three robotic platforms to explore an environment to locate and then retrieve a book situated in book shelf. Further search-and-rescue tasks are presented in [110]. In [111], two aerial robots combine their video streams to provide stereo vision to a robot situated on the ground. This robot then computes a heightmap of the ground that also includes the obstacles situated in the environment. The heightmap is used for path planning. This system, however, may not scale for systems that consider more than one robot on the ground. While the teams presented in [25, 110, 111] do not depend on GPS-based solutions and can be deployed in indoor environments, the team presented in [112] relies on GPS for localization and operates in outdoor environments. The team is composed of one car-sized robot and two quadrocopter drones. The terrestrial robot is equipped with the Kinect sensor and relies on GPS for localization. Through a control station that maintains communication contact to all three vehicles at all times, a human operator can issue reconnaissance and surveillance tasks at the team level.

In this chapter, we presented an aerial robot able to compute heightmaps based on images retrieved from a standard monocular camera. Similar techniques have already been used by flying platforms in previous research. For instance, in the pioneering work presented in [113], a tethered blimp flying at an altitude between 10 and 40 m is deployed in an outdoor environment to retrieve stereo images. These images are then used to compute the heightmap of an area covering several thousands of square meters. In [108], an aerial robot is presented that uses two different monocular vision streams to compute heightmaps at 1 Hz in both outdoor and indoor environments. The heightmap is used by the robot to detect safe landing-spots. Despite its high payload, we also showed in this chapter how a Kinect sensor can be used by an aerial platform to directly retrieve heightmaps that can be included in the decision-making relevant for the actions taken by the robots on the ground. Another such platform is presented in [102] in which the heightmap is used for navigation in indoor environments. Contrary to the work presented in this chapter, note that the heightmaps available to aerial robots in [102, 108, 113] do not represent the basis for decisions or actions made by terrestrial robots. Instead, the heightmaps

are solely included in the decisional processes relevant to aerial robots.

Cross-platform control and coordination in heterogeneous air-ground robot teams require solutions to a variety of problems. In recent years, multiple frameworks that include such solutions have been presented. For instance, a customizable framework is presented in [114] to enable collaboration between aerial drones and ground-based vehicles and other smart-objects. A real-world search-and-find task is used to validate the framework. During task execution, teams members are able to detect each others presence, select leaders of a team, and assign tasks to particular members in the team. Other researchers proposed a control scheme [115] that allows an air-ground team to coordinate and control its members in a leader-follower scenario. The followers are shown to be able to maintain a particular formation through the course of a mission. Numerous search-and-rescue scenarios are used to validate the scheme both in simulation and using real robots. A decentralized architecture enabling interaction within air-ground teams in the context of an area-inspection scenario is presented in [116]. The architecture is able to monitor an environment by combining the "global" coverage provided by the aerial robot with the "local" coverage provided by the ground-based robots. While these cooperation frameworks have been validated in a variety of application scenarios, note that none of them have been studied in the context of self-assembling robots.

For further examples on existing air-ground robot systems, we invite the reader to refer to the following two reviews [91, 117]. Detailed discussions on the advantages and disadvantages of heterogeneous systems are presented in [92, 98]. To the best of our knowledge, the work presented in this thesis represents the first and only implementation of a robotic system that enables self-assembling robots to cooperate with aerial robots.

In terms of the tasks considered in this chapter, the study presented in [101] represents the most related work. The study considers a homogeneous group of self-assembling robots that solves a gap-crossing task by executing a pre-programmed response behavior. The sensory equipment available to the robots do not permit them to estimate the gap width. Therefore, when a gap is detected, the response behavior forms chain morphologies of a pre-programmed size independent of the gap width. This response behavior is similar to the NCC control methodology presented in Section 5.4.2.1.

5.6 Summary

In this chapter, we demonstrated the first self-assembling robot system able to cooperate with aerial robots. We described supervised morphogenesis — a novel control methodology that increases the autonomy of self-assembling robots by allowing them to exploit the advantages provided by aerial robots. The presented approach does not require global information, a priori information, or GPS information.

We presented results of two case studies in which we studied supervised morphogenesis using two different air-ground robot teams. The case studies confirm the applicability and high portability of the presented approach even when different communication modalities and aerial platforms are considered for experimentation. They also showed that aerial assistance can be used to dynamically form task or environment dependent morphologies and increase the overall adaptivity of self-assembling robots. We showed how input from standard monocular cameras can be used to build three-dimensional models of environments such that aerial robots

can perform on-board simulations to evaluate the adequacy of different morphologies prior to their physical realization. In simulation-based studies, we showed that our approach can provide performance benefits by allocating resources optimally (in terms of number and location) for self-assembly.

The research presented in this chapter combines the work presented in both previous chapters and introduces a self-assembling robot system that – in many ways – goes beyond the state of the art. For instance, the robots do not depend on environmental cues neither to trigger self-assembly nor to choose target morphologies. Also, by allocating all decision-making authority to the aerial robot with a better view of the environment, self-assembling robots are able to operate in unknown environments and solve previously unknown tasks by following the ad-hoc guidance instructions provided by the aerial robot. However, supervised morphogenesis as presented so far has certain limitations that need to be overcome in further research before the overall system reaches genuine autonomy. In particular, the task-morphology mapping problem has not been entirely solved by the system as target morphologies of different shapes and sizes have been pre-compiled into morphology libraries and made available to the robots. The aerial robots were only required to find a morphology of the right size from this library and trigger formation. Furthermore, we only considered robot groups with a single aerial robot (i.e., either an eye-bot or an AR.Drone) responsible for supervising morphology formation on the ground. While this form of authority delegation may resemble the principles underlying those of centralized systems, this chapter merely outlines the control components of one specific robot in a multirobot system that, in general, could be decentralized and composed of many air-ground robot teams operating in parallel. However, more experimentation is required to identify potential challenges and confirm applicability in heterogeneous multirobot systems that consider multiple aerial robots (with potentially distinct capabilities) simultaneously providing assistance to robots on the ground. In the next chapter, we discuss potential directions that may be considered in future research to overcome these limitations and to achieve fully autonomous and scalable self-assembling robot systems able to function without any prior information.

CHAPTER 6

Conclusions

In this chapter, we conclude the research presented in this thesis by summarizing our contributions. We then give a brief outlook to future research directions.

Our main contribution has been to advance the state of the art in self-assembling robots through the development of novel control, communication, and cooperation techniques. First, we proposed mergeable nervous systems as the foundation for a paradigm shift in the sensorimotor coordination of physically connected robots. We showed that our proposal can provide self-assembling robots with unprecedented sensorimotor coordination capabilities. Second, we identified spatially targeted communication as an essential communication methodology to achieve spatial coordination in multirobot systems. We presented a scalable mechanism to achieve this form of communication in decentralized multirobot systems using standard hardware. Third, we introduced supervised morphogenesis to increase the level of autonomy in self-assembling robots by enabling cooperation with aerial robots.

The thesis presented here addresses some of the scientific challenges raised in the Ph.D. theses presented by Roderich Groß and Rehan O'Grady. They, among others, pioneered self-assembling robot research using the s-bots platform. More precisely, we provide concrete answers to multiple future research suggestions made by both researchers in their respective theses. For instance, in one of the final paragraphs of his thesis [118], Roderich Groß suggests the following when designing future self-assembling systems:

> Innovative designs can also be observed in nature. ... Certainly, many more self-assembly processes can be found in nature, and might inspire next generation designs. (page 176)

In Chapter 3, we presented MNS robots that are inspired by the unique self-assembling mechanism found in certain slime molds species [9, 8] and the motor control schemes observed in octopuses [48, 119]. Unicellular slime molds are aggregation organisms with unusual bodies. They can function independently, but when food resources are scarce, they are also able to aggregate and operate as a single body. However, these bodies lack a central nervous system and are therefore limited to (relatively) rudimentary behaviors. In an entirely different biological organism, researchers have found indications [48] that the octopus brain is able to send high-level commands to control arm movements. These commands were then shown to activate motor control programs that seem to be embedded locally in the arm itself. Researchers have also shown [119] that body parts of the octopus can sense and respond to changing environmental conditions without exchanging signals with the

brain – suggesting the existence of computational and decision-making faculties outside of the brain. By combining aspects of these phenomena observed in fundamentally different natural systems, we were able to design and develop a self-assembling robotic system that is able to exhibit capabilities that go beyond the state of the art. We first presented a fast and precise basic connection forming mechanism for individual robots. We then employed this mechanism in the autonomous formation of larger, composite robot bodies with a brain unit each. Subsequently, a novel control paradigm was introduced to allow composite robot bodies to be controlled as a single body independent of size and shape. A core idea we presented is the control of composite robot bodies based on brain units issuing high-level commands that are then translated by individual robotic units into (a sequence of) locally meaningful actuator commands. Based on this form of control, we demonstrated previously unseen features in any artificial or natural system including the ability to perform self-healing as a reaction to faulty units.

In Chapter 3, we presented an approach to establish spatially targeted communication links between robots that operate in a decentralized multirobot system. Spatially targeted communication allows robots in a multirobot system to choose their communication partners based on location — an important feature that can enable spatial coordination in a robot team. We first presented *iterative elimination process* – a protocol used to establish one-to-one spatially targeted communication links. Then, we presented *iterative growth process* that we developed to create one-to-many spatially targeted communication links by expanding existing one-to-one links. We presented formal descriptions of both protocols using Markov chain models and showed that the elimination process scales logarithmically with respect to the number of robots in the communication range while the growth process scales logarithmically with respect to the number of desired communication partners. This scalability aspect is vital for application scenarios considered in today's multirobot systems that tend to consist of large numbers of robots. We showed that off-the-shelf components such as LEDs and cameras available to most robotic platforms are sufficient to establish spatially targeted communication links both in homogeneous and heterogeneous robots teams (i.e., teams composed of robots that are fundamentally different in terms of their physical capabilities and the environments in which they operate).

In his pioneering thesis [120], Rehan O'Grady studied self-assembly as a response mechanism to changing environmental conditions. In the future work section of his thesis, he makes the following suggestion to increase the autonomy levels of future self-assembling robot systems:

> [In our experiments] ... we were forced to place explicit cues in the arena to allow the robots to distinguish between the two types of obstacle.
>
> For future systems to overcome such limitations, one possibility would be to give each agent in the system better sensing hardware. However, the simplicity of individual agents is often cited as a key strength of distributed self-assembling systems. An alternative might be to introduce heterogeneity into the system, and have some robots specialized in sensing that could then communicate relevant information to the other robots. In such a heterogeneous swarm, supervisory robots with more sophisticated sensory equipment could direct the morphogenesis activities of self-assembling robots. (page 137)

In Chapter 5, we presented supervised morphogenesis as a control methodology for heterogeneous teams composed of self-assembling robots and aerial robots. Supervised morphogenesis is based on the control and communication techniques presented in Chapters 3 and 4, respectively. We presented results of two case studies we carried out using a different heterogeneous team each. In both studies, the aerial robots were able to exploit their elevated position in the environment to monitor and control morphogenetic processes on the ground. Given this kind of ad-hoc supervision, we showed that self-assembling robots can be deployed in a priori unknown environments and solve previously unknown tasks. In simulation, we then quantified the performance benefits attainable to a heterogeneous team by analyzing the task completion times in a gap-crossing task.

In future work, one interesting line of research is to study how the advantages provided by an aerial robot can be shared by different groups of robots such that a variety of different morphologies can be formed on the ground in parallel. Alternatively, it may also be interesting to study how multiple aerial robots with complementing sensory capabilities may simultaneously provide different guidance instructions to the same group of robots on the ground. A further idea is to augment the computing capabilities available to the aerial robot by providing on-demand access to a cloud-based infrastructure [121] that can allocate the compute power necessary to run physics-based simulations. Physics-based simulations can be used to determine target morphologies for tasks that require high-fidelity solutions or for tasks that have low levels of fault tolerance. A further step is to couple such a cloud-based simulation framework with state-of-the-art machine-learning techniques [122] enabling aerial robots to use monocular images to learn about task-morphology mappings that then can be shared between aerial robots operating in distinct environments. Such a system can feature the ability to instantly produce the most appropriate morphology for potentially any type of physically plausible and previously unseen task and environment.

Another promising direction for future work is to study how the principles underlying mergeable nervous systems can be extended to modular systems that operate in three-dimensional environments [46, 123, 124] or consider other – such as legged – locomotion types [125]. The next frontier, however, may then be defined by the investigation on how self-assembling robots can determine which morphology is appropriate to which task or environment without any assistance from human operators or aerial robots. This direction of research is becoming even more important as the state-space of possible morphologies is growing rapidly due to the continuous miniaturization of individual robots allowing them to operate in new environments such as inside animal [126] and human bodies [64, 65]. Nature solves this problem of finding the appropriate morphology to a particular environment through the rather lengthy process of evolution. Similarly, evolutionary computation techniques [127, 128, 129, 130] may hold the key in providing the solution to the task-morphology mapping problem. The ever increasing computing power may also be helpful in solving the problem in close to real-time. In the future, an MNS robot able to run millions of evolutionary steps and explore large state-spaces within a short period of time may be able to autonomously evolve and build the robots of the required size and morphology. In such a future, when robots build other robots of required sizes and shapes without any external assistance, the true potential of self-assembling robots will be fulfilled as robots for particular tasks no longer need to be designed and built by humans.

Appendices

APPENDIX A

3D environment modeling using heightmaps

We considered two different methods for the aerial robot to acquire three-dimensional models of the surface underneath. First, we use stereo images retrieved from a standard monocular camera to compute heightmaps. Second, we use a dedicated sensor – the Microsoft Kinect – to have direct access to precomputed heightmap data. In the following, we describe details of both methods and present a quantitative and qualitative comparison between the two methods. We use an eye-bot as the aerial robot for both methods.

A.1 Stereo images retrieved from a monocular camera

The eye-bot retrieves two images taken from two different positions (see Fig 5.3e–f) using its downward-facing camera. Based on these two images, the eye-bot computes the elevation of the surface and the objects in the scene. When acquiring the images, we make the following spatial assumptions: (i) we assume that the aerial robot is able to remain (by hovering or by attaching to a ceiling) at a fixed height and that the image plane is parallel to the ground, (ii) we assume that an altitude sensor can provide reliable altitude measurements to the aerial robot, (iii) we assume that the relative distance between the two positions from which the images were taken is known (for instance through the eye-bot's 3D relative positioning sensor), and (iv) we assume the focal length of the camera has been previously obtained through a prior calibration step [131]. Through empirical experimentation, we have determined that for the experimental setup considered in this thesis, images taken at a distance of 30 cm from each other and from an altitude of 2.42 cm (measured from the ground to the tip of the camera) yield the most precise data for the experimental setup considered.

The eye-bot can retrieve images at a maximum resolution of 640×480 pixels – see Figure A.1a for an example. From two such images each taken from different positions, the eye-bot first computes a so-called *disparity map*. For a pair of stereo image, a disparity map contains the distance (in pixel) by which each point in the first image has moved in the second image. For instance, the displacement of points closer to the camera is higher than that of the points further away from the camera. See Figure A.1b for a representation of a disparity map. In a second step, the eye-bot calculates the height of each point in real-world distances based on the disparity of each point, the elevation (in cm) of the eye-bot, the displacement between the two images, and the focal length properties of the camera.

We summarize the individual steps required to compute a heightmap based on a

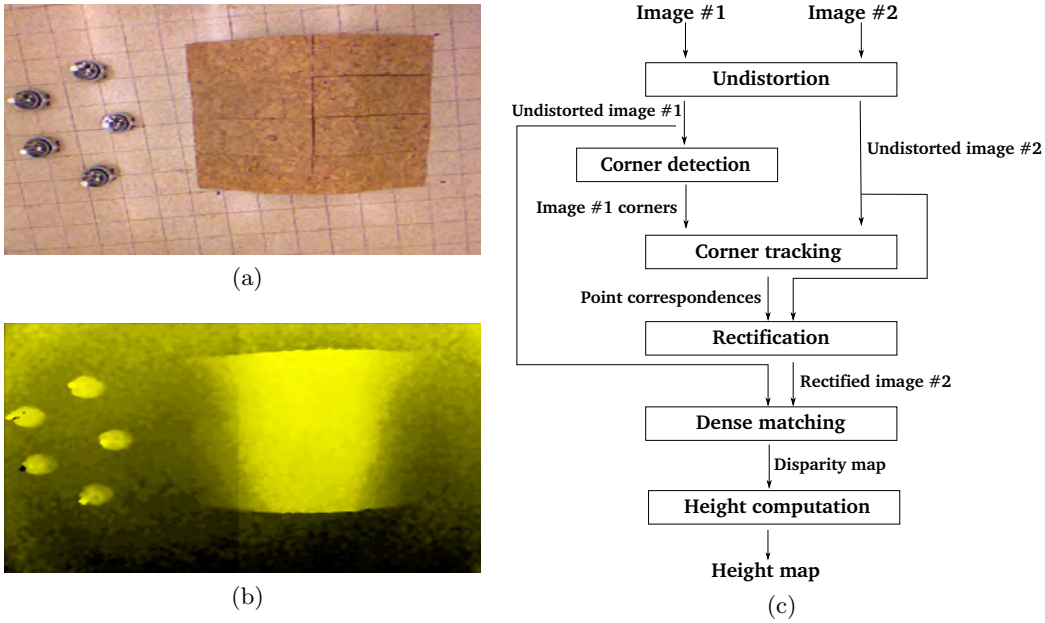


Figure A.1: Generation of heightmaps using stereo images. (a) A 640×480 image acquired by the eye-bot. (b) A representation of the computed disparity map in which brighter pixels denote greater motion and lesser distance to the camera. (c) A flowchart showing the computation of heightmaps based on a pair of stereo image. The result is a two-dimensional matrix with the elevation in cm for each pixel.

pair of stereo image using the flowchart shown in Figure A.1c. In an initial *Undistortion* step, the tangential and radial distortion is compensated. The distortion is introduced in the lens and modeled by the coefficients found by the calibration step. In the next three steps, the images are transformed such that the search for a point's corresponding point in the second image can be reduced to a horizontal scan. In this process called *Rectification*, first the *Corner Detection* step finds interesting feature points in the first image as described in [132]. In the *Corner Tracking* step that follows, these interesting feature points are tracked in the second image using the iterative Lucas-Kanade method [133] which outputs a set of sparsely matched points. This set of correspondences is then used in the *Rectification* step to find the transformation matrix that can be used to rectify the second image and to vertically align it to the first. A subsequent *Dense Matching* step is then applied to derive the disparity of each point in the image. In a final *Height Computation* step, stereo triangulation is applied to each disparity value in order to compute the heightmap – a two-dimensional matrix of size 640×480 that contains the elevation in cm for each point visible in both images. This method to extract three-dimensional information based on stereo images has been thoroughly studied by the computer vision community [134].

A.2 The Microsoft Kinect sensor

The eye-bot is able to take flight with a rather high payload of up to 2 kg. Hence, we mounted an additional dedicated sensor for the retrieval of heightmaps – the Microsoft Kinect sensor. The Kinect weighs ca. 1.4 kg and is a commercially available multi-purpose motion detection sensor that also features a depth sensor. The

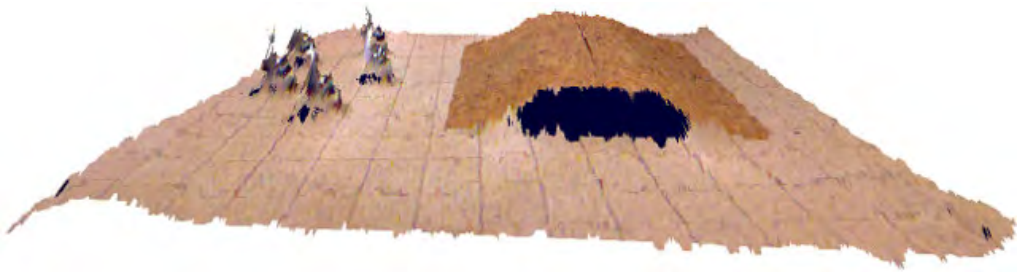


Figure A.2: A heightmap converted into a 3D point cloud showing five foot-bots and the hill obstacle. The heightmap was acquired by a Microsoft Kinect sensor mounted on an eye-bot hovering above the hill obstacle, as shown in Fig 5.3f.

depth sensor is based on an infrared laser projector used in combination with a CMOS sensor. Many third-party tools have been developed to exploit the sensor's rich feature set. For example, the MATLAB Image Acquisition Toolbox allows real time visualization of heightmaps using point clouds. A point cloud is a user-friendly representation of a heightmap that can be used to visualize all points in the field of view and is not limited to the particular viewpoint from which the heightmap has been generated. See Figure A.2 for an example that provides a human operator with a more intuitive representation of the data collected by the eye-bot. Furthermore, Kinect is shipped with libraries that can detect human gestures in real time. This feature has been also used by researchers in the robotics community to design human-swarm-interaction (HSI) tools allowing human operators to communicate with and control robot teams using gestures [135]. This inherent feature may become important as more robot teams (including aerial robots) are finding applications in warehouses and production lines and are required to be controlled by human operators using as few intermediate devices as possible. Despite the obvious advantages the Kinect sensor can provide, the sensor weight reduces flight autonomy significantly for most application scenarios including aerial robots. Therefore, the sensor has only been applied by limited number of aerial robots.

A.3 Quantitative and qualitative analysis

In Figure A.3, we present a quantitative analysis of the results obtained with the two methods considered to acquire heightmaps. We have plotted in red the mean elevations of each foot-bot's (shown in Figure A.1a) estimated trajectory, i.e., its height profile we acquired using stereo images. Mean elevations are based on values we extracted from 10 different heightmaps. The eye-bot assumes a foot-bot trajectory to be a straight line connecting the foot-bot's location in the deployment area and the light source. We have also plotted in blue the equivalent data acquired from 10 different heightmaps retrieved from the Kinect sensor. In this case, however, we have only plotted the height profile of the longest foot-bot trajectory along with error bars representing the standard deviation as the values are too close to each other to be plotted in a clearly comprehensible manner. The standard deviation for the elevation computed using stereo images is 2.91 cm (not shown in the figure) as opposed to the 2.14 cm for the elevation retrieved from the Kinect sensor indicating a slightly more reliable data source. However, as the figure shows, both methods deliver relatively precise data. Although we have observed that in our setup the

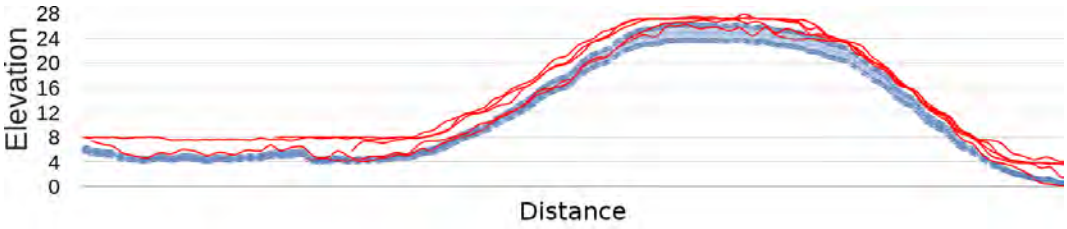


Figure A.3: Mean elevation of height profiles acquired by the eye-bot from 10 different heightmaps. Graphs in red result from computation based on stereo images while data retrieved from the Kinect sensor is shown in blue.

Table A.1: A qualitative comparison between the two methods considered to compute heightmaps. Thumbs up/down represent the desirability of a quality. An ideal sensor would checkmark all desirable properties and x-mark others (i.e., checkmark the first four and x-mark the later four).

	Quality	Monocular Camera	Kinect
👍	Availability on aerial platforms	✓	✗
👍	Data precision	✓	✓
👍	3rd party tools available	✗	✓
👍	HSI potential	✗	✓
👎	Extreme outliers	✓	✗
👎	Spatial assumptions	✓	✗
👎	Payload	✗	✓
👎	Ambient light dependency	✓	✗

absolute values of the surface elevation computed using stereo images constantly resulted in real-world values above those acquired from the Kinect sensor, the relative differences between any two points is almost identical for the two methods. That is, we observed that the inclination computed between any two points in a height profile resulted in almost identical values independent of the underlying method. However, note that the absolute values returned by neither method matches the ground truth (not presented) which remained between 3 and 4 cm under the values returned by the Kinect sensor. This is clearly visible in Figure A.3 for the flat surface area to the left of the hill obstacle. One explanation may be the fact that the eye-bot was slightly tilt (and hence not parallel to the ground) when the data was retrieved.

Further, we have also observed that there are fundamental qualitative differences between the methods. For instance, heightmaps computed from images retrieved from the monocular camera are extremely dependent on ambient light conditions and may also include extreme outliers caused by dust on the camera lens. In Table A.1, we have identified and listed a number of qualities that need to be considered before a decision can be made regarding which method to be adopted for the generation of heightmaps by aerial robots. The decision may not only depend on the aerial robot platform itself, but may also need to consider task/mission properties such as required flight time and capabilities of the aerial robot. Also refer to [32] for a detailed study on the trade-off between the autonomy of aerial robots and the payload they can carry.

List of figures

List of tables

Bibliography

[1] N. Mathews. Ph.D. thesis supplementary website. `http://iridia.ulb.ac.be/~mathews/PhD/supp`, 2017.

[2] W. Liu and A. Winfield. Autonomous morphogenesis in self-assembling robots using ir-based sensing and local communications. In *Proceedings of the 7th International Conference on Swarm Intelligence*, volume 6234 of *LNCS*, pages 107–118. Springer, Berlin-Heidelberg, Germany, 2010.

[3] A. L. Christensen, R. O'Grady, and M. Dorigo. SWARMORPH-script: a language for arbitrary morphology generation in self-assembling robots. *Swarm Intelligence*, 2(2–4):143–165, 2008.

[4] E. Klavins, R. Ghrist, and D. Lipsky. A grammatical approach to self-organizing robotic systems. *IEEE Transactions on Automatic and Control*, 51(6):949–962, 2006.

[5] M. Rubenstein, A. Cornejo, and R. Nagpal. Programmable self-assembly in a thousand-robot swarm. *Science*, 345(6198):795–799, 2014.

[6] R. O'Grady, A. L. Christensen, C. Pinciroli, and M. Dorigo. Robots autonomously self-assemble into dedicated morphologies to solve different tasks. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1517–1518. IFAAMAS, Richland, SC, 2010.

[7] H. Wei, Y. Chen, M. Liu, Y. Cai, and T. Wang. Swarm robots: From self-assembly to locomotion. *The Computer Journal*, 54(9):1465–1474, 2011.

[8] M. Ito, R. Okamoto, and A. Takamatsu. Characterization of adaptation by morphology in a planar biological network of plasmodial slime mold. *Journal of the Physical Society of Japan*, 80(7):074801, 2011.

[9] A. Takamatsu, E. Takaba, and G. Takizawa. Environment-dependent morphology in plasmodium of true slime mold physarum polycephalum and a network growth model. *Journal of Theoretical Biology*, 256(1):29–44, 2009.

[10] A. Lioni, C. Sauwens, G. Theraulaz, and J.-L. Deneubourg. Chain formation in œcophylla longinoda. *Journal of Insect Behavior*, 14(5):679–696, 2001.

[11] A. Lioni and J.-L. Deneubourg. Collective decision through self-assembling. *Naturwissenschaften*, 91(5):237–241, 2004.

[12] C. Anderson, G. Theraulaz, and J.-L. Deneubourg. Self-assemblages in insect societies. *Insectes Sociaux*, 49(2):99–110, 2002.

[13] K. Støy. Using situated communication in distributed autonomous mobile robotics. In *Proceedings of the Seventh Scandinavian Conference on Artificial Intelligence (SCAI)*, pages 44–52. IOS Press, Amsterdam, The Netherlands, 2001.

[14] A. L. Christensen. Efficient neuro-evolution of hole-avoidance and phototaxis for a swarm-bot. In *Mémoire de DEA, Technical report TR/IRIDIA/2005-14*. IRIDIA, Université Libre de Bruxelles, Brussels, Belgium, 2005.

[15] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, and M. Dorigo. ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4):271–295, 2012.

[16] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. A. Di Caro, F. Ducatelle, T. Stirling, A. Gutiérrez, L. M. Gambardella, and M. Dorigo. ARGoS: A modular, multi-engine simulator for heterogeneous swarm robotics. In *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5027–5034, Los Alamitos, CA, 2011. IEEE Computer Society Press.

[17] T. Krajník, V. Vonásek, D. Fišer, and J. Faigl. AR-Drone as a platform for robotic research and education. In *Research and Education in Robotics (EUROBOT)*, volume 161, pages 172–186. Springer, Berlin-Heidelberg, Germany, 2011.

[18] F. Mondada, L. M. Gambardella, D. Floreano, S. Nolfi, J.-L. Deneubourg, and M. Dorigo. The cooperation of swarm-bots: Physical interactions in collective robotics. *IEEE Robotics & Automation Magazine*, 12(2):21–28, 2005.

[19] M. Dorigo, E. Tuci, V. Trianni, R. Gross, S. Nouyan, C. Ampatzis, T. H. Labella, R. O'Grady, M. Bonani, and F. Mondada. SWARM-BOT: Design and implementation of colonies of self-assembling robots. In *Computational Intelligence: Principles and Practice*, chapter 6, pages 103–135. IEEE Computational Intelligence Society, Piscataway, NJ, 2006.

[20] R. Groß, M. Bonani, F. Mondada, and M. Dorigo. Autonomous self-assembly in swarm-bots. *IEEE Transactions on Robotics*, 22(6):1115–1130, 2006.

[21] S. Nouyan, R. Groß, M. Dorigo, M. Bonani, and F. Mondada. Group transport along a robot chain in a self-organised robot colony. In *Intelligent Autonomous Systems 9 – IAS 9*, pages 433–442. IOS Press, Amsterdam, The Netherlands, 2006.

[22] A. L. Christensen, R. O'Grady, and M. Dorigo. Morphology control in a self-assembling multi-robot system. *IEEE Robotics & Automation Magazine*, 14(4):18–25, 2007.

[23] A. Campo, S. Nouyan, M. Birattari, R. Groß, and M. Dorigo. Negotiation of goal direction for cooperative transport. In *Ant Colony Optimization and Swarm Intelligence: 5th International Workshop, ANTS 2006*, volume 4150 of *Lecture Notes in Computer Science*, pages 191–202. Springer-Verlag, Berlin, Germany, 2006.

[24] M. Bonani, V. Longchamp, S. Magnenat, P. Rétornaz, D. Burnier, G. Roulet, F. Vaussard, H. Bleuler, and F. Mondada. The MarXbot, a Miniature Mobile Robot Opening new Perspectives for the Collective-robotic Research. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4187–4193. IEEE Press, Piscataway, NJ, 2010.

[25] M. Dorigo, D. Floreano, L. M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy, D. Burnier, A. Campo, A. L. Christensen, A. Decugnière, G. Di Caro, F. Ducatelle, E. Ferrante, A. Förster, J. Guzzi, V. Longchamp, S. Magnenat, J. Martinez Gonzales, N. Mathews, M. Montes de Oca, R. O'Grady, C. Pinciroli, G. Pini, P. Rétornaz, J. Roberts, V. Sperati, T. Stirling, A. Stranieri, T. Stützle, V. Trianni, E. Tuci, A. E. Turgut, and F. Vaussard. Swarmanoid: A novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics & Automation Magazine*, 20(4):60–71, 2013.

[26] J.F. Roberts, T.S. Stirling, J-C. Zufferey, and D. Floreano. 2.5D infrared range and bearing system for collective robotics. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3659–3664. IEEE Press, Piscataway, NJ, 2009.

[27] S. Magnenat, P. Rétornaz, M. Bonani, V. Longchamp, and F. Mondada. Aseba: A modular architecture for event-based control of complex robots. *Mechatronics, IEEE/ASME Transactions on*, 16(2):321–329, 2011.

[28] E. Ferrante, A. E. Turgut, C. Huepe, A. Stranieri, C. Pinciroli, and M. Dorigo. Self-organized flocking with a mobile robot swarm: a novel motion control method. *Adaptive Behavior*, 20(6):460–477, 2012.

[29] A. Scheidler, A. Brutschy, E. Ferrante, and M. Dorigo. The k-unanimity rule for self-organized decision-making in swarms of robots. *Cybernetics, IEEE Transactions on*, PP(99):1–14, 2015.

[30] G. Pini, A. Brutschy, A. Scheidler, M. Dorigo, and M. Birattari. Task partitioning in a robot swarm: Retrieving objects by transferring them directly between sequential sub-tasks. *Artificial Life*, 20(3):291–317, 2014.

[31] S. Magnenat, R. Philippsen, and F. Mondada. Autonomous construction using scarce resources in unknown environments. *Autonomous Robots*, 33(4):467–.485, 2012.

[32] J.F. Roberts. *Enabling Collective Operation of Indoor Flying Robots*. PhD thesis, École polytechnique fédérale de Lausanne, Lausanne, Switzerland, 2011.

[33] J.F. Roberts, T. Stirling, J.-C. Zufferey, and D. Floreano. 3-D relative positioning sensor for indoor flying robots. *Autonomous Robots*, 33(1-2):5–20, 2012.

[34] P.-J. Bristeau, F. Callou, D. Vissière, and N. Petit. The navigation and control technology inside the AR.Drone micro UAV. In *Proceedings of the 18th IFAC World Congress*, pages 1477–1484. IFAC-PapersOnLine, Centerville, OH, 2011.

[35] R. Pfeifer, M. Lungarella, and F. Iida. Self-organization, embodiment, and biologically inspired robotics. *Science*, 318(5853):1088–1093, 2007.

[36] A. Westphal, N.F. Rulkov, J. Ayers, D. Brady, and M. Hunt. Controlling a lamprey-based robot with an electronic nervous system. *Smart Structures and Systems*, 8(12):39–52, 2011.

[37] K. Støy, D. Brandt, and D.J. Christensen. *Self-reconfigurable robots*. MIT Press, Cambridge, MA, 2010.

[38] G. Amato, D. Bacciu, M. Broxvall, S. Chessa, S. Coleman, M. Di Rocco, M. Dragone, C. Gallicchio, C. Gennaro, H. Lozano, T.M. McGinnity, A. Micheli, A.K. Ray, A. Renteria, A. Saffiotti, D. Swords, C. Vairo, and P. Vance. Robotic ubiquitous cognitive ecology for smart homes. *Journal of Intelligent & Robotic Systems*, Online First:1–25, 2015.

[39] L. Wang, M. Liu, and M. Q.-H. Meng. Real-time multisensor data retrieval for cloud robotic systems. *IEEE Transactions on Automation Science and Engineering*, 12(2):507–518, April 2015.

[40] A. L. Christensen, R. O'Grady, and M. Dorigo. SWARMORPH-script: A language for arbitrary morphology generation in self-assembling robots. *Swarm Intelligence*, 2(2-4):143–165, 2008.

[41] R. Fitch and Z. J. Butler. Million module march: Scalable locomotion for large self-reconfiguring robots. *International Journal of Robotics Research*, 27(3-4):331–343, 2008.

[42] R. O'Grady, A. L. Christensen, and M. Dorigo. SWARMORPH: Multi-robot morphogenesis using directional self-assembly. *IEEE Transactions on Robotics*, 25(3):738–743, 2009.

[43] M. Bonani, S. Magnenat, P. Rétornaz, and F. Mondada. The Hand-bot, a robot design for simultaneous climbing and manipulation. In *Proceedings of the 2nd International Conference on intelligent robotics and applications (ICIRA)*, pages 11–22. Springer, Berlin-Heidelberg, Germany, 2009.

[44] A.L. Christensen, R. O'Grady, M. Birattari, and M. Dorigo. Fault detection in autonomous robots based on fault injection and learning. *Autonomous Robots*, 24(1):49–67, 2008.

[45] M. Yim, W. m. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian. Modular self-reconfigurable robot systems [grand challenges of robotics]. *IEEE Robotics & Automation Magazine*, 14(1):43–52, 2007.

[46] J. Bongard, V. Zykov, and H. Lipson. Resilient machines through continuous self-modeling. *Science*, 314(5802):1118–1121, November 2006.

[47] W.-M. Shen, B. Salemi, and P. Will. Hormone-inspired adaptive communication and distributed control for CONRO self-reconfigurable robots. *IEEE Transactions on Robotics and Automation*, 18(5):700–712, 2002.

[48] G. Sumbre, Y. Gutfreund, G. Fiorito, T. Flash, and B. Hochner. Control of octopus arm extension by a peripheral motor program. *Science*, 293:1845–1849, 2001.

[49] M. G. Gouda and T. M. McGuire. Accelerated heartbeat protocols. In *Proceedings of the 18th International Conference on Distributed Computing Systems*, pages 202–209, IEEE Press, Piscataway, NJ, 1998.

[50] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret. Robots that can adapt like animals. *Nature*, 521:503–507, 2015.

[51] M. J. Abrams, T. Basinger, W. Yuan, C.-L. Guo, and L. Goentoro. Self-repairing symmetry in jellyfish through mechanically driven reorganization. *Proceedings of the National Academy of Sciences*, 112(26):E3365–E3373, 2015.

[52] R. Groß and M. Dorigo. Self-assembly at the macroscopic scale. *Proceedings of the IEEE*, 96(9):1490–1508, 2008.

[53] P. R. Wurman, R. D'Andrea, and M. Mountz. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. In *Proceedings of the 19th National Conference on Innovative Applications of Artificial Intelligence - Volume 2*, IAAI'07, pages 1752–1759. AAAI Press, 2007.

[54] H. Li, T. Wang, H. Wei, and C. Meng. Response strategy to environmental cues for modular robots with self-assembly from swarm to articulated robots. *Journal of Intelligent & Robotic Systems*, 81(3):359–376, 2016.

[55] L. Murray, J. Timmis, and A. Tyrrell. Self-reconfigurable modular e-pucks. In *Proceedings of the 8th International Conference on Swarm Intelligence*, ANTS'12, pages 133–144. Springer-Verlag, Berlin-Heidelberg, Germany, 2012.

[56] H.Wei, N. Li, M. Liu, and J. Tan. A novel autonomous self-assembly distributed swarm flying robot. *Chinese Journal of Aeronautics*, 26(3):791–800, 2013.

[57] R. Oung, F. Bourgault, M. Donovan, and R. D'Andrea. The distributed flight array. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA)*, pages 601–607. IEEE Press, Piscataway, NJ, 2010.

[58] S. Miyashita, M. Lungarella, and R. Pfeifer. *Tribolon: Water-Based Self-Assembly Robots*, pages 161–184. Springer London, London, UK, 2009.

[59] M. T. Tolley, M. Kalontarov, J. Neubert, D. Erickson, and H. Lipson. Stochastic modular robotic systems: A study of fluidic assembly strategies. *IEEE Transactions on Robotics*, 26(3):518–530, 2010.

[60] M. T. Tolley and H. Lipson. On-line assembly planning for stochastically reconfigurable systems. *International Journal of Robotics Research*, 30(13):1566–1584, 2011.

[61] J.A. Escalera, M. Doyle, F. Mondada, and R. Groß. Evo-bots: A Simple, Stochastic Approach to Self-Assembling Artificial Organisms. In *Proceedings of the International Symposium on Distributed Autonomous Robotic Systems*. Springer-Verlag, Berlin-Heidelberg, Germany, 2016.

[62] L. Brodbeck, S. Hauser, and F. Iida. Morphological evolution of physical robots through model-free phenotype development. *PLOS ONE*, 10(6):1–17, 2015.

[63] M. Mastrangeli, S. Abbasi, C. Varel, C. Van Hoof, J.-P. Celis, and K. F. Bóhringer. Self-assembly from milli- to nanoscales: methods and applications. *Journal of Micromechanics and Microengineering*, 19(8):083001, 2009.

[64] Z. Nagy, R. Oung, J. J. Abbott, and B. J. Nelson. Experimental investigation of magnetic self-assembly for swallowable modular robots. In *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1915–1920. IEEE Press, Piscataway, NJ, 2008.

[65] A. T. Becker, O. Felfoul, and P. E. Dupont. Toward tissue penetration by MRI-powered millirobots using a self-assembled gauss gun. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2015)*, pages 1184–1189, 2015.

[66] J. Nam, N. Won, H. Jin, H. Chung, and S. Kim. ph-induced aggregation of gold nanoparticles for photothermal cancer therapy. *Journal of the American Chemical Society*, 131(38):13639–13645, 2009.

[67] S. D. Perrault and W. C. W. Chan. In vivo assembly of nanoparticle components to improve targeted cancer imaging. *Proceedings of the National Academy of Sciences*, 107(25):11194–11199, 2010.

[68] M. Grzelczak, J. Vermant, E. M. Furst, and L. M. Liz-Marzán. Directed self-assembly of nanoparticles. *ACS Nano*, 4(7):3591–3605, 2010.

[69] A. Martinoli, K. Easton, and W. Agassounon. Modeling swarm robotic systems: a case study in collaborative distributed manipulation. *The International Journal of Robotics Research*, 23(4-5):415–436, 2004.

[70] John G. Kemeny and J. Laurie Snell. *Finite markov chains*. Springer-Verlag, New York, NY, 1976.

[71] Isermann R. Model-based fault-detection and diagnosis - status and applications. *Annual Reviews in Control*, 29(1):71–85, 2005.

[72] A.A. Goshtasby. Similarity and dissimilarity measures. In *Image Registration*, Advances in Computer Vision and Pattern Recognition, pages 7–66. Springer, London, UK, 2012.

[73] J. Pugh, X. Raemy, C. Favre, R. Falconi, and A. Martinoli. A fast onboard relative positioning module for multirobot systems. *Mechatronics, IEEE/ASME Transactions on*, 14(2):151–162, April 2009.

[74] A. Gutiérrez, A. Campo, M. Dorigo, D. Amor, L. Magdalena, and F. Monasterio-Huelin. An open localization and local communication embodied sensor. *Sensors*, 8(11), 2008.

[75] R. Grabowski and P. Khosla. Localization techniques for a team of small robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1067–1072. IEEE Press, Piscataway, NJ, 2001.

[76] F. Rivard, J. Bisson, F. Michaud, and D. Létourneau. Ultrasonic relative positioning for multi-robot systems. In *Proceedings of the IEEE International*

Conference on Robotics and Automation, pages 323–328. IEEE Press, Piscataway, NJ, 2008.

[77] A. Franchi, G. Oriolo, and P. Stegagno. Probabilistic mutual localization in multi-agent systems from anonymous position measures. In *49th IEEE Conference on Decision and Control*, pages 6534–6540, Atlanta, GA, 2010.

[78] L. Chaimowicz, T. Sugar, V. Kumar, and M. F M Campos. An architecture for tightly coupled multi-robot cooperation. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation (ICRA 2001)*, volume 3, pages 2992–2997. IEEE Press, Piscataway, NJ, 2001.

[79] B. P. Gerkey and M. J. Mataric. Sold!: auction methods for multirobot coordination. *IEEE Transactions on Robotics and Automation*, 18(5):758–768, 2002.

[80] L.E. Parker, B. Kannan, F. Tang, and M. Bailey. Tightly-coupled navigation assistance in heterogeneous multi-robot teams. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1016–1022. IEEE Press, Piscataway, NJ, 2004.

[81] Y. Zhang and L.E. Parker. IQ-ASyMTRe: Forming executable coalitions for tightly coupled multirobot tasks. *IEEE Transactions on Robotics*, 29(2):400–416, 2013.

[82] A. Stentz, A. Kelly, H. Herman, P. Rander, O. Amidi, and R. Mandelbaum. Integrated air/ground vehicle system for semi-autonomous off-road navigation. In *Proceedings of AUVSI Unmanned Systems Symposium*, 2002.

[83] R. T. Vaughan, G. S. Sukhatme, F. J. Mesa-Martinez, and J. F. Montgomery. Fly spy: Lightweight localization and target tracking for cooperating air and ground robots. In *Proceedings of the International Symposium on Distributed Autonomous Robot Systems*, pages 315–324. Springer-Verlag, Berlin, Germany, 2000.

[84] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar. Towards a swarm of agile micro quadrotors. *Autonomous Robots*, 35(4):287–300, 2013.

[85] D. H. Barnhard, J. T. McClain, B. J. Wimpey, and W. D. Potter. Odin and Hodur: Using bluetooth communication for coordinated robotic search. In *Proceedings of the International Conference on Artificial Intelligence (IC-AI)*, pages 365–371. CSREA Press, Athens, GA, 2004.

[86] J. T. McClain, B. J. Wimpey, D. H. Barnhard, and W. D. Potter. Distributed robotic target acquisition using bluetooth communication. In *Proceedings of the 42nd annual Southeast regional conference*, pages 291–296. ACM, New York, NY, 2004.

[87] M. Dorigo, M. Birattari, and M. Brambilla. Swarm robotics. *Scholarpedia*, 9(1):1463, 2014.

[88] K. Bolla, Z. Istenes, T. Kovacs, and G. Fazekas. A fast image processing based robot identification method for Surveyor SRV-1 robots. In *Proceedings of 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2011)*, pages 1003–1009. ASME, New York, NY, 2011.

[89] P. Stegagno, M. Cognetti, L. Rosa, P. Peliti, and G. Oriolo. Relative localization and identification in a heterogeneous multi-robot system. In *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA 2013)*, pages 1857–1864. IEEE Press, Piscataway, NJ, 2013.

[90] S. Nouyan, A. Campo, and M. Dorigo. Path formation in a robot swarm: Self-organized strategies to find your way home. *Swarm Intelligence*, 2(1):1–23, 2008.

[91] S. L. Waslander. *Autonomous Control Systems and Vehicles: Intelligent Unmanned Systems*, chapter Unmanned Aerial and Ground Vehicle Teams: Recent Work and Open Problems, pages 21–36. Springer Japan, Tokyo, Japan, 2013.

[92] V. Kumar and N. Michael. Opportunities and challenges with autonomous micro aerial vehicles. *International Journal of Robotics Research*, 31(11):1279–1291, 2012.

[93] C. Bills, J. Chen, and A. Saxena. Autonomous mav flight in indoor environments using single image perspective cues. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 5776–5783. IEEE Press, Piscataway, NJ, 2011.

[94] S. Zingg, D. Scaramuzza, S. Weiss, and R. Siegwart. MAV navigation through indoor corridors using optical flow. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3361–3368. IEEE Press, Piscataway, NJ, 2010.

[95] S. Shen, N. Michael, and V. Kumar. Autonomous multi-floor indoor navigation with a computationally constrained MAV. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 20–25. IEEE Press, Piscataway, NJ, 2011.

[96] A. Briod, P. Kornatowski, J.-C. Zufferey, and D. Floreano. A collision-resilient flying robot. *Journal of Field Robotics*, 31(4):496–509, 2014.

[97] A. Briod, A. Klaptocz, J. C. Zufferey, and D. Floreano. The Airburr: A flying robot that can exploit collisions. In *ICME International Conference on Complex Medical Engineering (CME)*, pages 569–574. IEEE Press, Piscataway, NJ, 2012.

[98] S. Lacroix and G. Besnerais. Issues in cooperative air/ground robotic systems. In *Robotics Research*, volume 66 of *Springer Tracts in Advanced Robotics*, pages 421–432. Springer, Berlin-Heidelberg, Germany, 2011.

[99] L. Chaimowicz and V. Kumar. *Distributed Autonomous Robotic Systems 6*, chapter Aerial Shepherds: Coordination among UAVs and Swarms of Robots, pages 243–252. Springer Japan, Tokyo, 2007.

[100] N. Mathews, A. L. Christensen, R. O'Grady, and M. Dorigo. Spatially targeted communication and self-assembly. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pages 2678–2679. IEEE Computer Society Press, Los Alamitos, CA, 2012.

[101] R. O'Grady, A. L. Christensen, C. Pinciroli, and M. Dorigo. Robots autonomously self-assemble into dedicated morphologies to solve different tasks (extended abstract). In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. In press.

[102] N. Michael, S. Shen, K. Mohta, Y. Mulgaonkar, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, K. Ohno, E. Takeuchi, and S. Tadokoro. Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *Journal of Field Robotics*, 29(5):832–841, 2012.

[103] A. Dewan, A. Mahendran, N. Soni, and K. M. Krishna. Heterogeneous ugv-mav exploration using integer programming. In *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5742–5749. IEEE Piscataway, NJ, 2013.

[104] M. Garzón, J. Valente, D. Zapata, and A. Barrientos. An aerial-ground robotic system for navigation and obstacle mapping in large outdoor areas. *Sensors*, 13:1247–1267, 2013.

[105] M. B. Mosely, B. P. Grocholsky, C. Cheung, and S. Singh. Integrated long-range uav/ugv collaborative target tracking. In *SPIE, Unmanned Systems Technology XI Conference*, volume 7332, April 2009.

[106] M. Persson, T. Duckett, and A. J. Lilienthal. Fusion of aerial images and sensor data from a ground vehicle for improved semantic mapping. *Robotics and Autonomous Systems*, 56(6):483–492, 2008.

[107] E. Mueggler, M. Faessler, F. Fontana, and D. Scaramuzza. Aerial-guided navigation of a ground robot among movable obstacles. In *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics*, pages 1–8. IEEE Press, Piscataway, NJ, 2014.

[108] C. Forster, M. Faessler, F. Fontana, M. Werlberger, and D. Scaramuzza. Continuous on-board monocular-vision-based elevation mapping applied to autonomous landing of micro aerial vehicles. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 111–118. IEEE Press, Piscataway, NJ, 2015.

[109] R. Käslin, E. Fankhauser, P.and Stumm, Z. Taylor, E. Mueggler, J. Delmerico, D. Scaramuzza, R. Siegwart, and M. Hutter. Collaborative localization of aerial and ground robots through elevation maps. In *Proceedings of the International Symposium on Safety, Security, and Rescue Robotics (SSSR)*, 2016.

[110] C. Luo, A. P. Espinosa, D. Pranantha, and A. De Gloria. Multi-robot search and rescue team. In *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 296–301. IEEE Press, Piscataway, NJ, 2011.

[111] J. H. Kim, J. w. Kwon, and J. Seo. Multi-uav-based stereo vision system without gps for ground obstacle mapping to assist path planning of ugv. *Electronics Letters*, 50(20):1431–1432, 2014.

[112] M. Langerwisch, M. Ax, S. Thamke, T. Remmersmann, A. Tiderko, K.-D. Kuhnert, and B. Wagner. Realization of an autonomous team of unmanned ground and aerial vehicles. In *Proceedings of the 5th International Conference on Intelligent Robotics and Applications (ICIRA)*, pages 302–312. Springer, Berlin, Germany, 2012.

[113] S. Lacroix, I.-K. Jung, and A. Mallet. Digital elevation map building from low altitude stereo imagery. *Robotics and Autonomous Systems*, 41(2–3):119–127, 2002.

[114] P. Pace, G. Aloi, G. Caliciuri, and G. Fortino. A mission-oriented coordination framework for teams of mobile aerial and terrestrial smart objects. *Mobile Networks and Applications*, 21(4):708–725, 2016.

[115] M. Saska, V. Vonásek, T. Krajník, and L. Přeučil. Coordination and navigation of heterogeneous mav-ugv formations localized by a 'hawk-eye'-like approach under a model predictive control scheme. *International Journal of Robotics Research*, 33(10):1393–1412, 2014.

[116] E. H. C. Harik, F. Guinand, H. Pelvillain, F. Guérin, and J. F. Brethé. A decentralized interactive architecture for aerial and ground mobile robots cooperation. In *Proceedings of the International Conference on Control, Automation and Robotics*, pages 37–43. IEEE Press, Piscataway, NJ, 2015.

[117] H. Duan and S. Liu. Unmanned air/ground vehicles heterogeneous cooperative techniques: Current status and prospects. *Science China Technological Sciences*, 53(5):1349–1355, 2010.

[118] R. Groß. *Self-assembling robots*. PhD thesis, Université Libre de Bruxelles, Brussels, Belgium, 2007.

[119] M. D. Ramirez and T. H. Oakley. Eye-independent, light-activated chromatophore expansion (lace) and expression of phototransduction genes in the skin of octopus bimaculoides. *Journal of Experimental Biology*, 218(10):1513–1520, 2015.

[120] R. O'Grady. *Morphologically Responsive Self-assembling Robots*. PhD thesis, Université Libre de Bruxelles, Brussels, Belgium, 2010.

[121] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg. A survey of research on cloud robotics and automation. *IEEE Transactions on Automation Science and Engineering*, 12(2):398–409, 2015.

[122] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *CoRR*, abs/1603.02199, 2016.

[123] M. Yim, W. Shen, B. Salemi, M. Moll, H. Lipson, E. Klavins, and G. Chirikjian. Modular self-reconfigurable robot systems; challenges and opportunities for the future. *IEEE Robotics & Automation Magazine*, 2007.

[124] R. Fitch and Z. Butler. Million module march: scalable locomotion for large self-reconfiguring robots. *The International Journal of Robotics Research*, 27(3-4):331–343, 2008.

[125] S. Kalouche, D. Rollinson, and H. Choset. Modularity for maximum mobility and manipulation: control of a reconfigurable legged robot with series-elastic actuators. In *IEEE/Safety, Security, and Rescue Robotics 2015*, October 2015.

[126] B. E.-F. de Ávila, P. Angsantikul, J. Li, M. Angel Lopez-Ramirez, D. E. Ramírez-Herrera, S. Thamphiwatana, C. Chen, J. Delezuk, R. Samakapiruk, V. Ramez, L. Zhang, and J. Wang. Micromotor-enabled active drug delivery for in vivo treatment of stomach infection. *Nature Communications*, 8(1):272, 2017.

[127] N. Bredeche, E. Haasdijk, and A.E. Eiben. On-line, on-board evolution of robot controllers. volume 5975 of *Lecture Notes in Computer Science*, pages 110–121. Springer Berlin-Heidelberg, 2010.

[128] A.E. Eiben, N. Bredeche, M. Hoogendoorn, J. Stradner, J. Timmis, A. Tyrrell, and A. Winfield. The triangle of life: Evolving robots in real-time and real-space. In *Proceedings of the European Conference on Artificial Life (ECAL)*, pages 1056–1063. The MIT Press, 2013.

[129] J. C. Bongard. Evolutionary robotics. *Communications of the ACM*, 56(8):74–83, August 2013.

[130] A. E. Eiben and J. Smith. From evolutionary computation to the evolution of things. *Nature*, 521:476–482, 2015.

[131] G. Bradski and A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, Cambridge, MA, 2008.

[132] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600. IEEE Press, Piscataway, NJ, 1994.

[133] J.Y. Bouguet. Pyramidal implementation of the Lucas Kanade feature tracker. Description of the algorithm. *Intel Corporation Microprocessor Research Labs*, 2000.

[134] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:993–1008, 2003.

[135] G. Podevijn, R. O'Grady, Y.S.G. Nashed, and M. Dorigo. Gesturing at subswarms: Towards direct human control of robot swarms. In *Proceedings of Towards Autonomous Robotic Systems (TAROS)*, volume 8069, pages 390–403, 2013.