University Institute of Lisbon

Department of Information Science and Technology

Synthesis of formation control for an aquatic swarm robotics system

Vasco Craveiro Vieira Teixeira da Costa

A Dissertation presented in partial fulfillment of the Requirements for the Degree of

Master in Computer Engineering

Supervisor

Prof. Dr. Anders Lyhne Christensen, Assistant Professor

ISCTE-IUL

Co-Supervisor

Prof. Dr. Sancho Moura Oliveira, Assistant Professor

ISCTE-IUL

July 2018

Resumo

As formações consistem na organização de objetos ou entidades de acordo com um padrão pré-definido. Elas podem ser encontradas na natureza, em animais sociais tais como peixes ou colónias de insetos, onde a organização espontânea em estruturas se verifica. As formações aplicam-se em diversos contextos, tais como cenários militares ou de aplicação da lei, onde são utilizadas para aumentar a performance operacional. O conceito está também presente em desportos coletivos tais como o futebol, onde as formações são utilizadas como estratégia para aumentar a eficiência das equipas.

Os enxames de robots são uma abordagem para o estudo de sistemas multi-robô compostos de um grande número de unidades simples, inspirado na organização de sociedades animais. Estes têm um elevado potencial na resolução de tarefas demasiado complexas para um único robot. Quando aplicadas na coordenação deste tipo de sistemas, as formações permitem o movimento coordenado e o aumento da sensibilidade do enxame como um todo.

Nesta dissertação apresentamos a síntese de controlo de formação para um sistema multi-robô. O controlo é sintetizado através do uso de robótica evolucionária, de onde o comportamento coletivo emerge, demonstrando ainda funcionalidades-chave tais como tolerância a falhas e robustez. As experiências iniciais na síntese de controlo foram realizadas em simulação. Mais tarde foi desenvolvida uma plataforma robótica para a condução de experiências no mundo real.

Os nossos resultados demonstram que é possível sintetizar controlo de formação para um sistema multi-robô, utilizando técnicas de robótica evolucionária. A plataforma desenvolvida foi ainda utilizada em diversos estudos científicos.

Palavras-chave: Sistemas de enxame, controlo de formação, algoritmos evolucionários, robótica.

Abstract

Formations are the spatial organization of objects or entities according to some predefined pattern. They can be found in nature, in social animals such as fish schools, and insect colonies, where the spontaneous organization into emergent structures takes place. Formations have a multitude of applications such as in military and law enforcement scenarios, where they are used to increase operational performance. The concept is even present in collective sports modalities such as football, which use formations as a strategy to increase teams efficiency.

Swarm robotics is an approach for the study of multi-robot systems composed of a large number of simple units, inspired in self-organization in animal societies. These have the potential to conduct tasks too demanding for a single robot operating alone. When applied to the coordination of such type of systems, formations allow for a coordinated motion and enable SRS to increase their sensing efficiency as a whole.

In this dissertation, we present a virtual structure formation control synthesis for a multi-robot system. Control is synthesized through the use of evolutionary robotics, from where the desired collective behavior emerges, while displaying key-features such as fault tolerance and robustness. Initial experiments on formation control synthesis were conducted in simulation environment. We later developed an inexpensive aquatic robotic platform in order to conduct experiments in real-world conditions.

Our results demonstrated that it is possible to synthesize formation control for a multi-robot system making use of evolutionary robotics. The developed robotic platform was used in several scientific studies.

Keywords: Swarm robotics, multi-robot systems, formation control, evolutionary algorithms, robotics.

Acknowledgments

First of all, I would like to acknowledge my supervisors, Anders Lyhne Christensen and Sancho Moura Oliveira. With them I had the chance to discover what really means to be a scientist and learn how scientific research is marked by several (often long) try-and-error cycles until good results are achieved, how important perseverance is (especially when "the ship seemed to be falling apart"), and how important is to assure that scientific integrity is maintained.

Despite the extraordinary work conducted in our lab, a major part of it would not had been possible without my laboratory (ex-) colleagues. I would like to acknowledgment Rita Ramos, Inês Mamede and Gustavo Martins. A special referral is due to Miguel Duarte, my daily colleague for almost 4 years, for the friendship and the invaluable lessons and everyday example of what is and means to be a scientist, as well as for the fellowship and every day availability to support my doubts as a "junior researcher"; to my past laboratory colleagues Jorge Gomes, Fernando Silva and Tiago Rodrigues for all the support on aquatic robots development process, manufacturing and field experiments, along with the friendship and fellowship.

Thirdly, I would like to refer myself to *VFabLab* for making possible the use of their workshop and digital fabrication equipments, essential in our aquatic robots manufacturing process. A special acknowledge is also due to this laboratory crew and past crew, Maria João Oliveira, João Sousa, Susana Neves, Filipa Osório, Bárbara Varela, and Alexandra Paio, for being always available to help, for the friendship and for the unvaluable lessons on design, architecture and digital fabrication, and with whom I learned much more than I ever had the chance to return.

I would like to address an acknowledge to all my *IEEE* fellows. Being a past-chair and one of the members present in the room when *IEEE ISCTE-IUL student branch* was founded, as well as one of the *IEEE ISCTE-IUL RAS student chapter* co-founders, I was far from imagine the numerous conquers that we, as a student group, were able to achieve together, when it all started. I can proudly say that we had the opportunity to organize the first *Google Dev Portugal* at *ISCTE-IUL*, among several other complex events and workshops, which taught me invaluable lessons on organizational and human resources management. A particular acknowledge is also due to *IEEE Portuguese section* that made possible for several of our student members to represent not only the branch and *ISCTE-IUL*, but also Portugal in several international events, allowing us to show to the world the best of what is done by us.

I also would like to acknowledge the actual and past administrative staff from the several institutions I had the chance to contact with during my stay at *ISCTE-IUL*: Fátima Silva, Marisa Manteigas and Alice Espada from *ISTA*, Fátima Estevens from *ISTAR* and Sara Correia, Tereza Traquinas and Ana Rodrigues from *IT*, for reducing the complexity of several bureaucratic processes, for the "never ending patience" and the sympathy characteristic from all of them.

I would like to acknowledge the amazing colleagues I had the chance to work with during my academic journey: Rui Madeira, Diogo Pedroso, Eduardo Gonzalez, João Paulino, Rudi Luís, Filipe Martins, Hugo Sousa and Fábio Martins for supporting me in several projects and academic assignments we developed together, and making possible for me to successfully split my time among several academic projects including *IEEE ISCTE-IUL student branch* and scientific research.

I thank my friends Gonçalo Viegas, Manuel Sousa, Constança Fernandes and Bárbara Coelho for the numerous adventures, continuous support to my academic career and for the continuous life lessons. With them I had the opportunity to travel the world, see some of the most beautiful landscapes, climb extraordinary routes, and share what I keep as some of the best moments and lessons in my live. A special acknowledgment is also due to my girlfriend and muse, Margarida, which demonstrated to be an important support and motivator during this process.

The greatest acknowledgment is due to my parents, Ana Paula and Américo, for the never ending patience, love and support, putting up with some of my worst moods and encouraging me when scientific work was not presenting the expected results. To them I owe my education and this thesis is nothing but a result of what they taught me and made me possible to access.

Finally, to *Fundação para a Ciência e Tecnologia* (FCT) and *Instituto de Telecomunicações* (IT) for founding CORATAM and HANCAD projects, that served as basis to the accomplishment of this dissertation (EXPL/EEIAUT/0329/2013 and UID/EEA/50008/2013, respectively).

Contents

List of Figures

Acronyms

AI Artificial intelligence

ANN Artificial neural network

ASV Autonomous surface vehicle

AUV Autonomous underwater vehicle

EA Evolutionary algorithm

EOD Explosive ordnance disposal

ER Evolutionary robotics

FSM Finite state machine

GPS Global positioning system

IMU Inertial measurement unit

MOOS Mission oriented operating suite

NEAT Neuroevolution of augmenting topologies

ROS Robot operating system

SR Swarm robotics

SRS Swarm robotic system

Chapter 1

Introduction

Patterns are present in our everyday lives. In nature, from plants to animals, in society, and in knowledge areas so apart as computer sciences and arts, patterns consist of regular sequences or forms of structure, organization or action. The use and recognition of patterns made possible and strongly contributed to the development of technologies and techniques such as DNA samples comparison [1], speech recognition [2], computer vision [3], anti-money laundering systems [4], big data analytics [5], among others, and are object of study in several areas.

Formations consist of a specific use of patterns in individuals organization and disposition in space. It is possible to find them in nature, where social animals such as fish schools and flock of birds use motion according to patterns to improve motion efficiency, reducing energy consumption [6, 7], and as an anti-predation behavior to decrease the exposure to predators [8, 9]. Humans make use of formations for a long time, primarily in war tactics. These date back to ancient Greece and Persia, with the use of formations such as *Skjaldborg*, a wall of shields, and *phalanx* formation, a dense rectangular formation composed of infantry, which increase troop mobility while maintaining protection. Later in Roman period, various formations are some of the contributions of this civilization to war science, with the development of formations such as *testudo* and *infantry square*. The last one was supposedly used latter in Battle of Aljubarrota, according to chronicler Fernão Lopes, during the Luso-Castilian war in 1384-1397, and by the French troops in Napoleonic Wars as an effective technique to repel cavalry attacks by the infantry. Many other formations and tactics were also developed by the several military troops across the globe, with numerous improvements during the first and second World War. This military knowledge is still used, despite some of the old formations had went into disuse, due to the improvements of the various military branches. The knowledge is also used in civil scenarios, such as in riot control situation by law enforcement authorities, where authorities' number is usually reduced when compared with the number of offenders.

Robots have been used in military and emergency scenarios for some time [10]. These allow for the conduction of missions with reduced troops exposure to dangerous scenarios. On emergency situations, robots have their use in tasks such Explosive ordnance disposal (EOD), search and rescue, and to carry out tasks in toxic environments [11, 12], reducing the exposure of the rescue and law-enforcement teams.

With the advances on Artificial Intelligence (AI), the use of AI techniques in military robots automation is a subject under debate and study [13]. Apart from the numerous ethical considerations regarding the Laws of War and the Rules of Engagement, the automation and use of AI techniques for military robots may conduct to an improvement of the missions performance, as human factors continue to present a high influence in accidents and incidents in the use of this type of systems [14].

Several AI techniques can be used to provide robots' control. Moving away from the traditional rigid programming, a main line of research now consists of the use of AI control on robotics [15].

One of many scientific areas studying this problem is Evolutionary Robotics (ER). This area of knowledge has the *potential* for control systems synthesis [16] making use of Darwin's *Survival of the fittest* concept. Through this technique, the objective for the controllers to achieve is initially defined, and the controllers are optimized based on how well they perform the objective, rather than manually programed. The synthesis process usually takes place in simulation environment, although it has been demonstrated that training or evolving robots in real environments is possible [17]. The number of trials needed to test the system discourages the use of physical robots during the training period [18].

ER can be applied on the control synthesis of both single-robot and multi-robot systems. A sub-set of multi-robot systems types are Swarm Robotic Systems (SRS). These are inspired in swarm societies present in nature, composed of multiple simple units such as ants or bees [19]. The control of this systems is decentralized, meaning that there is no central point of coordination. During the control synthesis, swarm behaviour emerges, and the different units coordinate through the environment sensing and communication with each other.

In this dissertation, we combine the several concepts, through synthesis of formation control for a SRS composed of Autonomous Surface Vehicles (ASVs), making use of evolutionary techniques.

The maritime environment was chosen as sea represents one of the major resources of the Portuguese territory. Considering that the Portuguese territory extends for 3.8 million km^2, and that 97% of it is sea [20], this resource has an elevated strategic interest and a large number of opportunities may arise from it. This interest is amplified due to the proposal for the extension of the continental shelf, submitted on May 11$^{\text{th}}$, 2009 and latter subject to addendum on August 1$^{\text{st}}$, 2017 [21].

Despite representing the majority of the Portuguese territory, the sea continues to present several uncertainties and to be highly unexplored. The use of SRSs has potential to transform this reality, invigorating how certain types of maritime missions are carried out, and enabling a totally new class of these. Taking advantage of the key-features demonstrated by this family of systems, such as fault tolerance, robustness and flexibility, as well as from the decentralized control and self-organization, and provided that the cost of each drone is kept sufficiently low, swarms of drones could be deployed in large numbers. Systems composed of large numbers of drones could overcome some of the key limitations of current systems, namely the ability to cover large areas, which is essential in missions involving tasks such as search, monitoring, and patrolling.

The use of ASVs is justified by the reduced complexity and cost of this systems, when compared with Autonomous Underwater Vehicles (AUVs). The use of underwater vehicles implies for an increased effort on enclosures sealing, and communications and positioning systems design, as communications based on radio-frequency are limited both on propagation and bandwidth. Inexpensive positioning systems such as Global positioning system (GPS) are also unavailable underwater, for the same reason. The use of surface vehicles also allows for the possibility of interfacing both air and water environments, making possible for this systems to act as information gateways between the environments, along with the collection of environmental information from both water and air.

1.1 Objectives

The main goal of this dissertation consists of formation control demonstration on a decentralized self-organized multi-robot system, or SRS. In order to accomplish formation control, a number of control synthesis techniques will be studied, which should allow for the produced controllers to successfully transfer from simulation environment to real world conditions. The synthesized controllers should display a number of key-features, such as fault tolerance, robustness, scalability and flexibility, typically inherent from swarm system.

1.2 Scientific Contribution

The work covered in this dissertation led to the following scientific contributions:

- A review on the use of evolutionary processes to accomplish formation control;
- Demonstration of the use of evolutionary robotics processes to synthesize control for a multi-robot systems performing motion in formation patterns;
- Design and construction of an aquatic swarm robotic platform that served as base in several scientific studies;

The conducted work has been published in several national and international scientific conferences and resulted in 10 scientific publications:

- Vasco Costa, Miguel Duarte, Tiago Rodrigues, Sancho Moura Oliveira, and Anders Lyhne Christensen. Design and development of an inexpensive aquatic swarm robotics system. In *Proceedings of OCEANS 2016 - Shanghai*, pages 1–7. IEEE Press, Piscataway, NJ, 2016. DOI: `10.1109/OCEANSAP.2016.7485496`
- Tiago Rodrigues, Miguel Duarte, Margarida Figueiró, Vasco Costa, Sancho Moura Oliveira, and Anders Lyhne Christensen. Overcoming limited onboard sensing in swarm robotics through local communication. *Transactions on Computational Collective Intelligence*, 9420(XX):201–223, 2015. DOI: `10.1007/978-3-319-27543-7_10`

- Miguel Duarte, Vasco Costa, Jorge Gomes, Tiago Rodrigues, Fernando Silva, Sancho Moura Oliveira, and Anders Lyhne Christensen. Evolution of collective behaviors for a real swarm of aquatic surface robots. *PLoS ONE*, 11(3): 1–25, 2016. DOI: 10.1371/journal.pone.0151834

- Miguel Duarte, Vasco Costa, Jorge Gomes, Tiago Rodrigues, Fernando Silva, Sancho Moura Oliveira, and Anders Lyhne Christensen. Unleashing the potential of evolutionary swarm robotics in the real world. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion - GECCO '16 Companion*, pages 159–160. ACM Press, 2016. DOI: 10.1145/2908961.2930951

- Miguel Duarte, Jorge Gomes, Vasco Costa, Tiago Rodrigues, Fernando Silva, Victor Lobo, Mário Monteiro Marques, Sancho Moura Oliveira, and Anders Lyhne Christensen. Application of swarm robotic systems to marine environmental monitoring. In *Proceedings of OCEANS 2016 - Shanghai*, pages 1–8. IEEE Press, Piscataway, NJ, 4 2016. DOI: 10.1109/OCEANSAP.2016.7485429

- Miguel Duarte, Jorge Gomes, Vasco Costa, Sancho Moura Oliveira, and Anders Lyhne Christensen. Hybrid control for a real swarm robotic system in an intruder detection task. In *Proceedings of the 18th European Conference on the Applications of Evolutionary Computation (EvoStar)*, pages 213–230. Springer, Berlin, Germany, Berlin, Germany, 2016. DOI: 10.1007/978-3-319-31153-1_15

- Anders Lyhne Christensen, Sancho Oliveira, Octavian Postolache, Maria João de Oliveira, Susana Sargento, Pedro Santana, Luis Nunes, Fernando Velez, Pedro Sebastião, Vasco Costa, Miguel Duarte, Jorge Gomes, Tiago Rodrigues, and Fernando Silva. Design of communication and control for swarms of aquatic surface drones. In *Proceedings of the International Conference on Agents and Artificial Intelligence - Volume 2: ICAART*, pages 548–555. SCITEPRESS - Science and and Technology Publications, 2015. DOI: 10.5220/0005281705480555

- Anders Lyhne Christensen, Miguel Duarte, Vasco Costa, Tiago Rodrigues, Jorge Gomes, Fernando Silva, and Sancho Oliveira. A sea of robots, 2016. Best Robot Video Award @ AAAI-16 Video Competition (AIVC 2016). Phoenix, Arizona. February 2016

- Fernando Velez, Aleksandra Nadziejko, Anders Lyhne Christensen, Sancho Moura Oliveira, Tiago Rodrigues, Vasco Costa, Miguel Duarte, Fernando

Silva, and Jorge Gomes. Experimental characterization of wsns applied to swarms of aquatic surface drones. In *Proceedings of the 10th Conference on Telecommunications (CONFTELE)*, 2015

- Fernando J. Velez, Aleksandra Nadziejko, Anders Lyhne Christensen, Sancho Oliveira, Tiago Rodrigues, Vasco Costa, Miguel Duarte, Fernando Silva, and Jorge Gomes. Wireless sensor and networking technologies for swarms of aquatic surface drones. In *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, pages 1–2. IEEE, 2015. DOI: `10.1109/VTCFall.2015.7391193`

The work conducted by me, along with the research team, was also covered in four national TV segments and 30+ international outlets, including IEEE Spectrum, GizMag and the Daily Mail.

All the used software modules, hardware components designs and specifications have been made publicly available under the GNU LGPLv3 license, and can be found on our research group's website[1] and GitHub repository[2].

1.3 Dissertation Structure

This dissertation is divided into five separated chapters. After a brief introduction and description of the scope of this work in chapter 1, a revision of the state of the art on the concepts covered by this dissertation can be found in chapter 2. In the following chapter, chapter 3, a description of the used simulation environment and its setup is provided, along with the description of the conducted experiments. In this chapter we also present an analysis of the achieved results. In order to move from simulation to real world conditions, there was the necessity of building a robotic platform with a set of key features that enable swarm robotic experiments. Such robotic platform is presented in chapter 4, focusing both the manufacturing process and the technical details of this platform. The built robotic platform was later used in the conduction of a set of real-world experiments, described in the same chapter. Finally, in chapter 5, a summary of the conclusions is presented, along with some scientific question and topics to be approached in future work.

[1] `http://biomachineslab.com/`
[2] `https://github.com/BioMachinesLab`

Chapter 2

State of the Art

In this chapter, we review the related work and the state of the art of the domains covered by this dissertation. This review is divided into four different sections: *(i)* Swarm robotics, *(ii)* Formation control for multi-robots systems, *(iii)* Synthesis of control for SRSs, and *(iv)* Existing aquatic robotic platforms.

2.1 Swarm robotics

Swarm robotics (SR) is an approach to the study of robotic systems inspired in social insects and self-organizing animal societies [19, 32], such as bees and ants. It relies in the use of robotic systems composed of large quantities of relatively simple and inexpensive, autonomous robots with decentralized control, to solve complex tasks [19]. In such systems, the robotic units make control decisions based on individual sensors readings, coordinating with nearby robots. While usually composed of homogeneous robots, heterogeneous SRSs are also possible as demonstrated in [33, 34].

SRSs have a number of potential advantages when compared with traditional multi-robot systems, such as inherent scalability, flexibility, and robustness to faults [19, 35, 36], similar to what can be observed in social insects societies [37]. Theses systems are also inherently scalable [38] and do not present a single point of failure [39]. Such properties make SRS suited to accomplish tasks where high temporal and spatial resolution is a requirement, such as natural-life monitoring and localization, environmental monitoring, and border patrolling.

Simple behavior demonstration making use of SRSs usually include tasks such as aggregation, flocking, foraging, clustering, sorting and path formation, already demonstrated in a variety of experiments [35]. Several studies were conducted in the past in SR, both on land [34, 40–42], in aquatic environments [28, 43, 44], and in the air [45, 46]. The majority of the studies have, however, been limited to controlled laboratory environments [36].

Swarm-bots project [42, 47, 48] conducted experiments using up to 16 small robots named *s-bots* [49], demonstrating task-oriented self-assembly behavior in a swarm of robots. Through the use of a gripping system, the robotic units were able to physically connect with each other and rapidly form structures. This allowed them to perform tasks such as object transport [50], gaps [51] and hill crossing [52], and navigation in rough terrain [49]. In this project, the maintenance of swarming key-features was also subject of scientific study, and the scalability of the self-assembly process was demonstrated [48].

In the scope of Swarmanoid project [34], researchers successfully demonstrated an SRS composed of three different types of robots (foot-bot, hand-bot and eye-bot) performing a search and retrieval task. In this project, decentralized control relying on limited communication and local information was used, in order to achieve heterogeneous SRS coordination while navigating in a complex 3-D environment. The use of evolutionary techniques allowed for the synthesis of artificial neural networks (ANNs) controllers, leading to the emergence of a global swarm behavior [34].

Other scientific contributions have been made, including CoCoRo project [44], where researchers studied control synthesis for a underwater SRS, in order to perform monitoring and search tasks using local processing and information. Scerri et al. [43] were able to demonstrate an SRS composed of up to 5 robotic units performing environmental monitoring tasks. This system used a centralized control approach, which limits the deployment capacity in remote locations. Finally, Rubenstein et al. [40] used a scalable SRS composed of up to 100 low cost robotic units named *Kilobots* to demonstrate collective behaviors such as foraging, formation control, photo-taxis and synchronization.

2.2 Formation control for multi-robots systems

The development of multi-robot systems often requires to find the right balance between *(i)* simple and inexpensive units, and *(ii)* the units' sensing and processing capacity. This balance may lead to poor final sensing ranges and resolutions, due to hardware limitations. A bio-inspired strategy can be used to overcome such limitations, such as the use of formations.

Formations consist of the organization of individuals in a particular arrangement, or pattern [53]. Patterns are present in a majority of things surrounding us in our daily lives. They can be found in several areas such as micro-biology [54], geography [55], botanics [56] and in nature in general [57]. Formations are also present in nature, such as flocks of birds and schools of fish [6, 58], which use such strategies to increase individuals sensing capacity and motion's efficiency, as well as reduce vulnerability to predators [6, 59, 60]. Such concepts are also extensively used in military and emergency scenarios to increase operations performance [61], and in collective sport modalities such as football [62].

When used in multi-robot systems, formations allow to export some of the properties verified in nature to this systems. Through the use of an extensive number of units spread across a large area according to a formation, it is possible to increase the system's sensing efficiency. This is done without adding a communication's overhead [63], in opposition to techniques such the one described in [23]. In these situations, each individual covers a specific area of the environment with greater detail, and robots distribution in space is optimized.

Several studies were previously conducted on formation control and formation control synthesis for multi-robot systems. According to Chen and Wang [64], Lawton et al. [65], Guanghua et al. [66], typical control strategies include *(i)* leader following, a strategy where a subset of robots are chosen as leaders and the remaining as followers [40, 67–69]; *(ii)* virtual structure, approach where the formation is treated as a fixed single entity from where the formation positions are extracted [59, 70, 71]; *(iii)* behavior-based method, where control results from weighing several individual behaviors according to their importance, such as goal seeking or obstacle avoidance [65, 72]; *(iv)* artificial potential functions that make use of potential fields. In this strategy, the areas to be avoided, such as obstacles, produce a repulsion force, while the areas to be occupied produce an attractive force [73, 74]. The last control strategy consists of *(v)* graph theory based approach. In this method a graph is used to represent the robots characteristics and constrains, as it takes advantages of graph and control theories for the formation control [75].

Formation control characteristics can also be divided into three different segments, according to Michaud et al. [76]: *(i)* perceptual characteristics, *(ii)* formation characteristics, and *(iii)* control characteristics.

Oh et al. [77] adds an extra layer of classification, accordingly to the sensed and the control variables. The formation control techniques are divided among three different categories:

- Position-based control: each of the agents senses it position in the environment in respect to a global coordinate system.
- Displacement-based control: the positions are calculated from the displacement in respect to a global coordinate system.
- Distance-based control: the agents control the relative positions and distances to the neighbors.

Multiple approaches also exist at the decision control process level. These are usually classified into either distributed [63, 65, 69, 70, 72, 78, 79] or centralized [59, 80, 81] control approaches.

2.3 Synthesis of control for SRSs

Control for SRSs can be synthesized making use of several different techniques. ER [16] is a promising research field that studies the use of automatic synthesis of robot controllers, making use of evolutionary computation techniques. Classic approaches based on manual programming have a tendency to impose a high complexity level, and ER has become a more viable alternative [36]. Starting with a specification of the task to accomplish, an evolutionary algorithm (EA) optimizes the candidate solutions, or *genomes*, and evaluates them according to their performance accomplishing the specified task. The optimization of the candidate solutions follows a Darwinian approach, where the fittest individuals are the survivors [82]. Through this technique, collective self-organized behavior emerges [16, 83, 84], avoiding the need for manual specification of the low-level and behavior of each individual in the swarm [85].

Several scientific contributions making use of evolutionary synthesized control were made in robotics field, demonstrating tasks such as coordinated motion [86–88], area patrolling [89], intruder detection [27], synchronization [84], flocking [90], prey hunting [91], hole avoidance [92], chain formation [93], creation of communication networks [46], pattern formation [94], aggregation [95–98], and communication emergence [99]. However, the majority of the studies were conducted in simulated or in highly controlled environments, mostly making use of robotic platforms such as the *e-puck* [100], the *Khepera* [101], the *s-bot* [49] and the *Thymio* [102].

Obtained high popularity in 80′s and 90′s, ANNs are algorithms inspired on the computational process that takes place in a biological brain [103, 104]. Using graphs theory [105], the ANNs can be represented as a set of nodes and vertices, which represent the neurons and the synapses of a real brain, respectively [106]. In this representation, the vertices are characterized by numeric values, the weights. The set of this numeric values is named *genome*, which encodes an ANN parameters, including the network structure and the connections weights.

ANNs have been extensively used in a variety of scientific studies. They have demonstrated the successful performance of these in tasks such as pattern recognition [107, 108], data validation [109], classification [110], and industrial [111, 112] and robotic systems control [34, 42, 112, 113]. They present several advantages over other algorithms, namely robustness to noise and to intra-network faults [104]. However, there are also drawbacks in their use namely *(i)* the high computational cost required to synthesize ANN-based controllers, due to the large number of required evaluations, and *(ii)* the presence of non-explicit knowledge, becoming difficult to extract rules from the neural network.

Through the use of ER techniques it is possible to synthesize ANNs based controllers from where collective swarm behavior emerges [24, 38]. Synthesizing control for groups of robots is then scalable: the emergence of control is only dependent on the evaluation functions tailoring, being independent from the swarm size.

As an alternative to the standard monolithic controllers, *task decomposition* can be also be used in order to build complex robot behaviors [114, 115]. This control architecture consists of the division of the main complex task into several simpler sub-tasks, and synthesizing control for each of the sub-tasks [27, 115]. The different sub-controllers are then combined using an arbitrator, which delegates the system control to one of the sub-controllers at a time [27, 114]. The technique allows for a simpler job on fitness functions definition, taking advantaged of the division-to-conquer methodology [114]. It has also been demonstrated that such technique allows for the synthesis of control solutions that are able to out-perform traditional ER techniques, on single-robot systems [87].

Another challenge in the ER is how to transfer the evolved synthesized control from simulation into real-world conditions. This presents a challenge since evolution tends to exploit simulation specific characteristics, usually not present in real-world conditions [18]. This is usually referred to as *reality gap* [116] and represents one of the issues why evolved controllers present a low performance when transfered from simulation to real-world conditions [117]. There are several causes for this problem, according to Miglino et al. [18], namely: (i) numerical simulations usually do not take in account all robot's and environment's physical laws, since models are often simplified in order to reduce computational cost [118], (ii) simulated sensors usually present perfect and noise free information, different from real sensors that introduce noise, and (iii) simulation and real sensors and actuators may perform or be positioned in slightly different locations in robots, translating different dynamics and sensing parameters. Several strategies, however, can be adopted in order to overcome such challenge, as described by Miglino et al. [18], namely: (i) the use of an accurate model that mimics the dynamics and the interaction of the robot in the environment, which can be developed through the measure of the real-world parameters making use of robot's sensors and actuators, (ii) the introduction of noise during controllers evolution, and the (iii) use of a hybrid evolutionary process, through prior evolution in simulation environment and the continuation of the evolutionary process in the real-world conditions. Jakobi [116] also proposes the use of a reduced simulation model, were the model is based on a reduced set of features identified as minimal for the controllers' synthesis, while the remaining features are injected with noise. Koos et al. [118] on the other hand proposes an hybrid model, where the controllers performance is evaluated both in simulation and in real-world conditions. During the evolutionary process, the controller periodically transfered, and the model is updated.

2.4 Existing aquatic robotic platforms

Significant development was made on AUVs and ASVs robotic systems in last recent years [119, 120], with application in several different scenarios including environmental mapping and monitoring [43, 121–124], search and rescue [125, 126]. Despite the existing systems, these are usually expensive and only capable of simple tasks. The use of evolutionary techniques in the control synthesis for aquatic robots was also subject of research in only a reduced amount of studies. Some examples include tasks such as station keeping [127] and predator-prey [128]. Nonetheless, a few more complex systems were idealized and developed to demonstrate coordination and environmental monitoring tasks [43, 44]. The different robotic platforms can be classified and organized accordingly to the several different characteristics [120, 129], which include:

- **Hull and structural components** which holds the sensors, actuators and processing components. Several hull topologies have been studied namely single [126, 130] and multi-hull [121, 122];
- **Propulsion systems**, which can follow several topologies namely single thrust (including both *jet* and rudder systems) [43, 126, 130] or differential thrust [121, 122];
- **Energy and power systems**, which include a variety technologies in order to generate (solar, wind and gas generators), manage and distribute (voltage and current meters, voltage converters) and store (batteries and fuel) energy, depending on missions type and the required autonomy;
- **Navigation and control systems**, which include both processing systems and the on-board software that performs navigation and coordination decisions. Several on-board softwares and operative systems have been developed including Robot operating system (ROS) [131] considered the *de facto* standard by the robotics community [132], and Mission oriented operating suite (MOOS)/ MOOS-IvP [133, 134] more commonly used in aquatic robotics [135];
- **Communication systems**, which covers different technologies including radio [121, 126, 130, 136] and light [44], allowing coordination with other robotic units, remote control and telemetry reporting to a control station;
- **Data collection devices**, usually named *sensors*, which enable to sense specific environmental parameters. The collected data can either be used by the control systems in the decision making process or stored for posterior analysis. Typical sensors include compasses, Inertial measurement units (IMUs) and GPS receivers [120].

Chapter 3

Experiments with Formations

In this chapter, we describe the approach used to synthesize formation control for a SRS where individuals have a limited sensory capacity.

The objective of the experiments is for the robots to accomplish a formation through the occupation of the formation's spots. Formations are composed of spots, which can be defined as areas in space arranged in a pattern, when observed from a macroscopic point-of-view. During the experiments duration, a robot should move to each these areas, and keep within it. As a formation translates and rotates in space, the different areas (or spots) also move, and robots should be able to maintain the formation through the preservation of their positions inside the spots.

The first control approach consisted in monolithic ANN control synthesis (Section 3.4). We further conducted experiments on FSM-based control synthesis and compared the two control approaches (Section 3.5). Finally, in Section 3.6, we assessed the robustness and fault tolerance of both the control strategies.

3.1 Methodology

The experiments were conducted in an unbounded simulation environment where robots and formation spots are represented as circles, and can freely move in the environment. In experiments, the robots must locate the formation's spots within the environment and distribute themselves along them, in order to accomplish the desired formation shape.

Due to the need of a controlled environment and the level of complexity emergent from the necessity of testing behaviors with large number of robots [34], real-world experiments are usually not easy to perform. A common approach is therefore the use of simulation environment. For that purpose, simulations where performed making use of *JBotEvolver* simulation framework [137], developed and extensively used within our research group. In order to accelerate the synthesis, we used a distributed computation system also developed by our research group [138], that accelerates evolutionary processes through tasks parallelization [137].

3.2 Experimental Setup

Each experiment starts with the generation of a formation to be accomplished during the experiment. These consist in virtual entities defined by a set of spots randomly distributed in a circular area around the formation's center. An example of a randomly generated formation can be found in figure 3.1. The formations are also characterized by movement equations, providing translation and rotation movement types. Each of the movements properties speed and direction are randomly generated according to uniform distributions, with the parameters presented in table 3.1. The use of randomly generated movement parameters allows for the reduction of controllers' over-fitting to specific formation shapes and movements settings. During the post-evaluation process other formation shapes were used, as later detailed in subsection 3.4.1. A detailed compilation of the parameters used in the experiments can be found in table 3.1.

Prior to the simulation process starts, the robots are pre-loaded with the characterization of the generated formation, which defines the formation they have to collectively achieve as a virtual entity. This characterization includes: (i) the formation geographic center coordinates, (ii) the formation spots' relative geographic positions, and (iii) a description of the formation movement, since formation motion can be decomposed into simple movement equations. The robots then make use of this information to calculate the sensory information for each time step. This technique allows for the robots to know the status of the formation they have to accomplish, in each of the time steps, avoiding for the need of extra communication with the status information. It also allows for fault tolerance, as each of the robotic units is able to calculate the formation status.

In order to evaluate how the synthesized controllers perform the desired task and allow for comparison between several control approaches, four different metrics where created, namely:

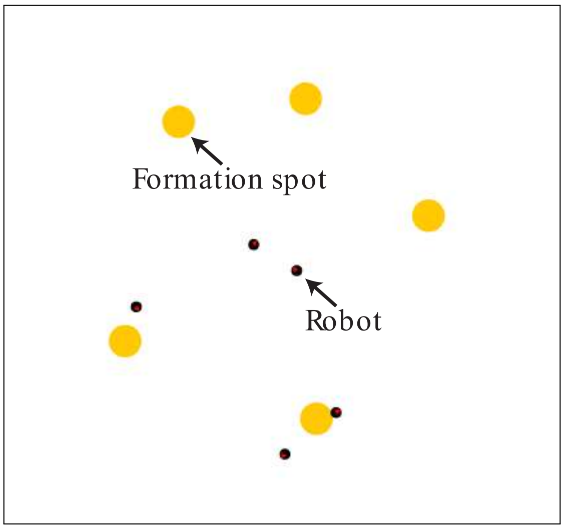


FIGURE 3.1: Illustration of a randomly generated formation used during controllers evolution. The formation spots are identified in yellow, while the robots are pictured in black color

Parameter	Value
Formation shape parameters	
Available formation shapes	Random, line of bearing, arrow, circle
Randomly generated shape object positioning radius	[0, 15[m
Line of bearing shape horizontal spacing	6.0 m
Arrow shape horizontal spacing	6.0 m
Arrow shape vertical spacing	6.0 m
Circle shape radius	10.0 m
Formations spots radius	1.5 m
Formation motion parameters	
Translation velocity	[0.15, 0.4[m/s
Translation azimuth	[0, 360[°
Angular velocity	[0.015, 0.02[m/s
Rotation direction	CW, CCW

TABLE 3.1: Parameters used in the formations shape and motion generation

- ***Time of robots inside the formation spots*** allows to ascertain how controllers perform into guiding robots to the correct position inside a formation spot and maintain the robot within the spot boundary. An ideal controller must maximize this value, producing a behavior where the robots occupy a formation's spot and maintain their position inside it.

- ***Time until first total formation occupation*** measures the quantity of time spent by the robots until all formation's spots are occupied from the beginning of the experimental run. This metric allow us to estimate how efficiently controllers guide the robots into occupying the spots in the formation and achieve the desired formation.

- ***Number of different formation spots occupied by a robot*** allows to measure the number of different formation spots that a single robot occupies while the experiment takes place. Since energy is a limitation in real hardware, ideally the robots must occupy and follow a unique spot, since spot changing may lead to unnecessary energy consumption.

- ***Time until a formation spot is reoccupied*** allows to ascertain how efficiently the swarm overcomes a fault situation. In a fault situation where a robot becomes inoperative, a formation spot will become free, as the formation moves in the environment and the robot is inert. This metric measures the time that a swarm as a all takes to reoccupy the unoccupied formation spot, using a spare robot.

3.3 Simulation Model

For the experiments, a model was created based on the real aquatic robots developed within our research team [22] and later described in chapter 4. The real aquatic robots were subjected to measurement of several parameters, namely acceleration, minimum and maximum speeds, turning rate and communication range. All this measurements were then used to adjust the simulation model in order to match the real robots' characteristics and minimize the differences between simulated an real sensors and actuators. The use of complex physic and dynamics simulation was avoided, in order to keep a low computational cost, as other strategies were used to reduce *reality gap*, detailed later in subsection 3.4.1.

The simulated robots were equipped with three different sensors, namely(i) a robot sensor, (ii) a formation sensor, and (iii) a compass and position sensor. A compilation of all the sensors configuration parameters can be found in table 3.2.

3.3.1 Robot Sensor

The robot sensor allows for each of the robots to detect the distance to the closest robot, in fours quadrants. This sensor, illustrated in figure 3.2, covers the 360° area around the robot, and is divided in four segments oriented toward the angles

Parameter	Value
Robot sensor	
Range	40.0 m
Number of segments	4
Cone aperture	90 °
Formation sensor	
Range	40.0 m
$d_{commute}$	2.5 m

Table 3.2: Sensor's parameters used in simulation experiments

0 °, 90 °, 180 ° and 270 °. Each segment is able to sense the distance to the closest neighboring robot within a range limit, as previously identified in table 3.2. This sensor was implemented in the real robotic platform, latter described in chapter 4, where each of the robotic units is capable of sensing the distance to the closest robot in each quadrant, based on the information exchanged through wireless communication. The sensor's range limit parameter used in simulation corresponds to the maximum wireless communication range verified between the real robots. The response of each sensor's segment fed in the ANN is defined by:

$$i = \begin{cases} \frac{r_{sensor} - d_{closest\_robot}}{r_{sensor}} & \text{if } r_{\text{sensor}} - d_{\text{closest\_robot}} \geq 0 \\ 0 & \text{if } r_{\text{sensor}} - d_{\text{closest\_robot}} < 0 \end{cases} \tag{3.1}$$

where i is the sensor response in range [0,1[, r_{sensor} is the sensor range and $d_{\text{closest\_robot}}$ is the distance to the closest neighboring robot.

3.3.2 Formation spot sensor

The formation sensors sense the closest unoccupied spot, and provides three pieces of information: (i) the relative orientation, (ii) the distance, and (iii) the relative velocity. These information are computed based on the formation information loaded on the robots prior to simulation takes place. The distance and relative velocity sensor components present a linear response. In opposition, the relative orientation calculation varies accordingly to the distance to the closest free spot. When far from the free spot, the sensor makes use of robot's and spot's geographic positions to calculate the orientation. When close to the free spot, the robot's and spot's velocity vector are used in the calculation. This strategies has as objective to first lead the robot to approach the spot and then to optimize its trajectory to match the spot's movement trajectory. The following equations translate the

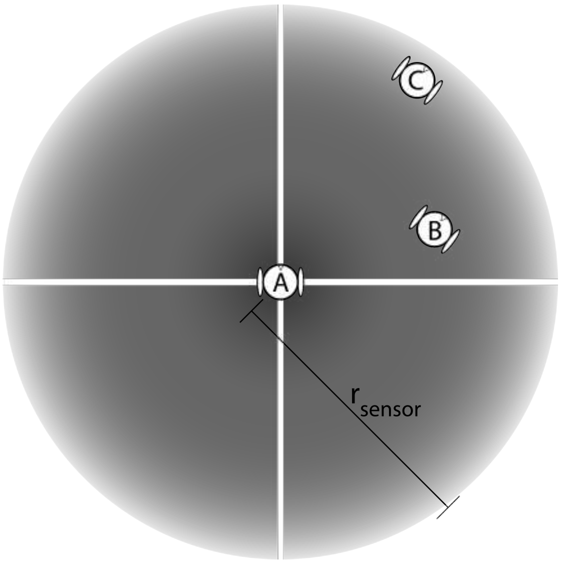


FIGURE 3.2: Robot sensor is composed of four 90° sections, covering 360° around the robot, with r_{sensor} detection range. In this example, for the top right sensing quadrant, the sensor uses the distance to robot B for the calculation, as it is closer than robot C. The distance sensing is always made in relation to the nearest robot.

different sensor components responses, respectively the relative orientation (equation 3.2), the distance (equation 3.3) and the relative velocity (equation 3.4):

$$j = \begin{cases} \frac{\alpha}{360.0} + 0.5 & \text{if } d_{\text{closest\_robot}} \leq d_{\text{commute}} \\ \frac{\vec{v}_{\text{robot}} \measuredangle \vec{v}_{\text{closest\_spot}}}{360.0} + 0.5 & \text{if } d_{\text{closest\_robot}} > d_{\text{commute}} \end{cases} \tag{3.2}$$

$$k = \frac{r_{\text{sensor}} - d_{\text{closest\_spot}}}{r_{\text{sensor}}} \tag{3.3}$$

$$l = \frac{\|\vec{v}_{\text{robot}}\| - \|\vec{v}_{\text{closest\_spot}}\|}{5 \cdot r_{\text{sensor}}} + 0.5 \tag{3.4}$$

where i is the sensor response in range [0,1[, which is fed in the ANN; α is the difference between the direction to north and the direction to the closest free spot (also called azimuth), varying in range [-180, 180[°; $\vec{v}_{\text{robot}}$ is the robot's velocity vector; $\vec{v}_{\text{closest\_spot}}$ is the closest free spot velocity vector; d_{commute} is the distance between the robot and the free spot at which the sensor commutes from using the geographic position to use the velocity vector, on the orientation component calculation; r_{sensor} is the sensor's range; and $d_{\text{closest\_spot}}$ is the distance to the closest formation's free spot. The several parameters used by this sensor are outlined in figure 3.3, and are calculated from the formation's characterization loaded in the robots, prior to simulation process start.

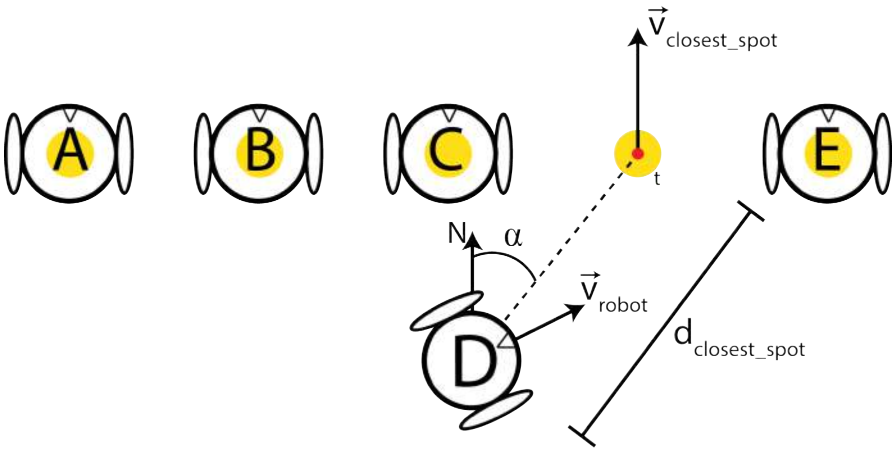


FIGURE 3.3: Parameters used by formation sensor. During mission, the robot D must move to target t, and occupy it.

3.3.3 Compass and position Sensor

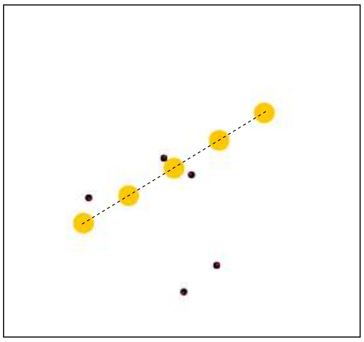
The compass and position sensor mimics the sensory information available in the real robot, provided by both the compass sensor and the GPS receiver. The environment is divided into a Cartesian grid, and the robots absolute position within the grid is fed to the sensor. The strategy allows to map the absolute coordinates to various latitude and longitude coordinates, when conducting experiments in real-world scenarios.

3.4 Monolithic control approach

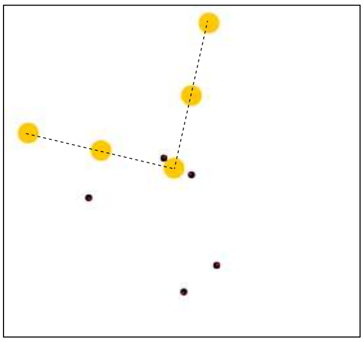
In this section, we present the results of the experiments conducted using monolithic ANN controllers. In these, a single controller is responsible to provide both guidance towards the closest unoccupied formation's spot and maintain the robot within its boundary while it moves in the environment. Each of the robot must occupy a position in the formation, as there are as many formation spots as robots.

3.4.1 Evolutionary Setup

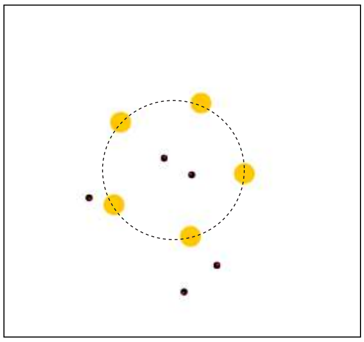
In order to synthesize control for the monolithic approach, a generational evolutionary algorithm was used. Each generation was composed of 150 genomes that



(A) Line of bearing formation



(B) Arrow formation



(C) Circle formation

FIGURE 3.4: Illustration of the three available formation shapes used during the post-evaluation phase. The formation spots are identified in yellow, while the robots are pictured in black color

correspond to different ANN topologies. Each genome was evaluated in conditions where both the number of robots and the formation's movement type is varied. The fitness mean is then used in the selection process. Each evaluation lasts 2000 time steps, equivalent to 200 seconds. After all the genomes are evaluated, an elitist approach was used. The top five controllers are chosen to parent the next generation, where each genome origins 29 new genomes. The mutated 145 genomes plus the original 5 genomes constitute the next generation. This process is then repeated for a total of 400 generations. During the evolutionary process, a randomly generated formation shape is used, robots quantity is varied in the $[3, 5]$ range, and three different motion types are used (rotation only, translation only, and both), leading to a total of 9 simulation configurations.

The evolutionary setup was replicated in ten independent evolutionary runs, with different random seeds. After the evolutionary process had taken place, we post-evaluated each of the runs' top controllers in order to obtain a more accurate estimate on how well the controllers' perform the desired task, as well as collect the set of metrics previously identified in section 3.2. Each controller was post-evaluated in a total of 162 different simulation configurations, with each evaluation lasting for 4000 time steps, equivalent 400 seconds. This number of simulation configurations is achieved through the use of three different formation shapes (line of bearing, arrow and circle, illustrated in figures 3.4a, 3.4b and 3.4c respectively) based on naval formation shapes described in [139]. Different formation's movement types are also tested: translation only, rotation only and both. Robots quantity was again varied in the $[3, 5]$ range. This set of configurations was then repeated 6 times per controller.

In order to synthesize ANN controllers where swarm behavior emerges, Neuroevolution of Augmenting Topologies (NEAT) was used [140]. This widely used generational algorithm evolves not only the ANN weights but also its topology, adding and removing connections between the nodes, during the evolutionary process. Such technique as proven to conduct to reduced complexity solutions in some situations when compared with traditional evolutionary techniques [141]. The default NEAT parameters were used to configure the algorithm, and can be found in table 3.3.

Parameter	Value	Parameter	Value
		NEAT	
Population size	150	Target species count	5
Allowed recurrency	True	Mutation probability	25%
Prob. add node	3%	Probability mutation bias	30%
Prob. add link	3%	Crossover probability	20%
		Simulation Noise	
GPS noise	1.8 m	Compass noise	10°
Motor delay	500 ms	Heading offset	5%
Speed offset	10%	Motor output noise	5%
Drift speed	[0,0.1] m/s		

Table 3.3: NEAT and noise parameters used in the controllers evolution

In order to enable the evolved genomes to better transfer from simulation to real-world conditions, a conservative amount of noise was introduced to sensors, actuators and environment during controllers evolution. This approach is a computational-effective way of evolving individual that do not differ in the simulated and in the real environment [18]. A list of the noise parameters used in the evolutionary process can be found in table 3.3. In addition to noise, each genome tested in different experimental setups with variations on swarm size and on the formation movement type, speed and direction, increasing the transferability of the synthesized controllers thought the exposure to an increased set of environmental situations.

In order to evaluate the controllers, we used the fitness function represented in equation 3.5, defined in function of the experiment's time step t. The fitness value is calculated by the sum of the fitness components $\Phi(t)$ and $\Psi(t)$, represented by equation 3.6 and 3.7, from time step 1 until time step t. This means that at each time step, the fitness value is affected by the previous time steps. The fitness component $\Phi(t)$ was used to force the robots to occupy the formation spots, while $\Psi(t)$ forces the robots to keep aligned with the formation spot's motion direction, once they are inside a formation spot.

In equations 3.5, 3.6, and 3.7, s_{quantity} and s_{occupied} are the number of formation spots and of occupied spots at a certain time step; T is the experiment's duration in time steps; $r_{\text{inside\_spot}}$ and r_{quantity} are the number of robots occupying the formation spots at a certain time step, and the total quantity of robots; and *orientation_ difference* is the difference, in degrees, between the robot's orientation and the velocity vector of the occupied formation spot at a certain time step, given by $\|\vec{v}_{\text{robot}}\| - \|\vec{v}_{\text{closest\_spot}}\|$.

$$F(t) = \sum_{t=1}^{t} \Phi(t) + \Psi(t) \tag{3.5}$$

$$\Phi(t) = \frac{s_{\text{T}}}{s_{\text{quantity}} \cdot T \cdot 10} \tag{3.6}$$

$$\Psi(t) = \frac{\left(\sum_{r=0}^{r_{\text{inside\_spot}}} 1 - \frac{|\text{orientation\_difference}|}{180^{\circ}}\right) \cdot 10}{r_{\text{quantity}} \cdot T} \tag{3.7}$$

3.4.2 Results and Discussion

3.4.2.1 Performance

The normalized fitness scores from the highest performing controller of each of the independent evolutionary runs are summarized in figure 3.5. The best run fitness value is represented by the red line.

In order to assess the controllers' performance, the metrics previously identified in section 3.2 where collected during the post-evaluation process. The results, obtained on 4000 time steps long simulations, can be found in figure 3.6, representing the metrics for the evolutionary top controllers.

Through the *Number of different formation spots occupied by a robot* metric we can observe that robots occupy 1.221 ± 0.079 different spots on average, meaning that the robots have a high tendency of entering and following an unique formation's spot rather than change the occupied spot afterwards. For the formation to be accomplished for the first time in simulation, it takes in average 473 ± 105 simulation steps or 24% of the experiment duration, represented by *Time until first total formation occupation* metric. Last, robots spent on average 837 ± 75 simulation steps inside a spot (or 42% of the experiment time), represented by *Time of robots inside the formation spots* metric.

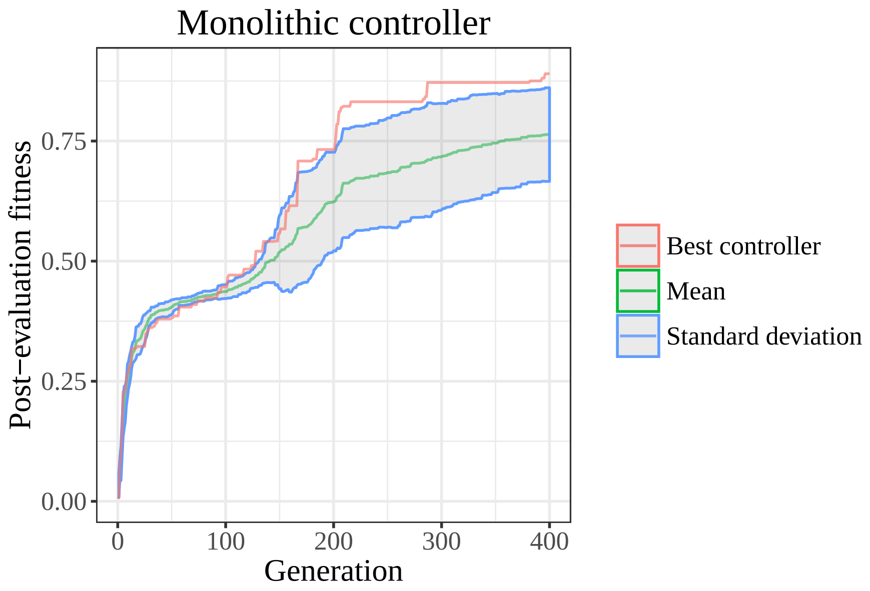


FIGURE 3.5: Plot of the monolithic controllers fitness scores.

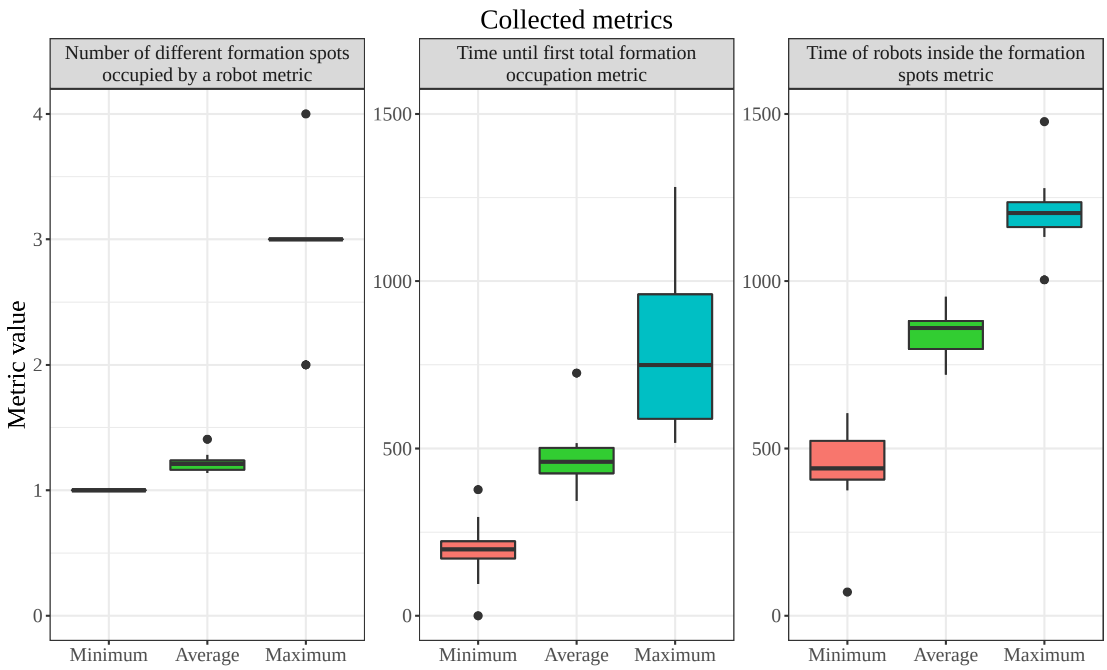
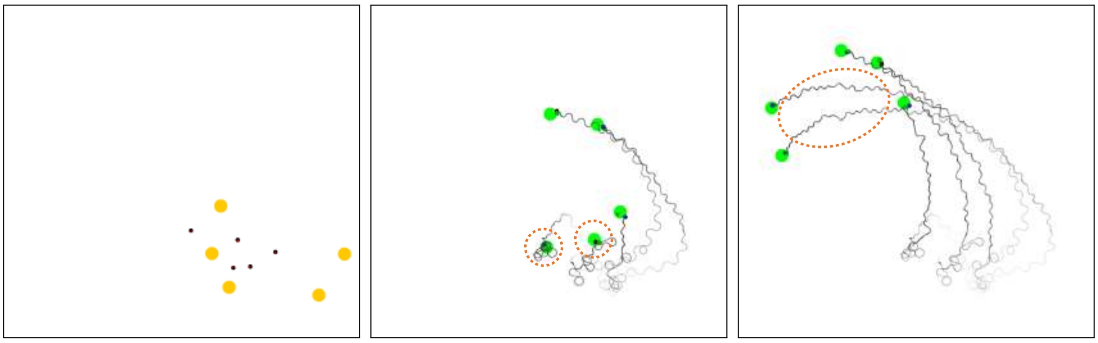


FIGURE 3.6: Plot of the metrics minimum, average and maximum values. These values are obtained from the top performing controller per evolutionary run, in 4000 simulation steps long experiments, and consist in the minimum, average and maximum verified values, per metric.

3.4.2.2 Behavior

In figure 3.7, we can observe an example of a formation accomplishment making use of a swarm of robots with monolithic control. In this situation, a randomly generated formation is used. We can observe that when a formation spot is moving at a low speed or it is almost stopped, it is difficult for the robots to maintain their position within the spot boundaries. Such situation takes place in the first 1000 time steps of the formation accomplishment example, illustrated at figure 3.7, where the three spots at the middle of figure 3.7a almost don't move through out near 1000 time steps, observable at figure 3.7b. In these cases, robots usually adopt a circling pattern around the spot or entering and exiting the same, highlighted in red on figure 3.7b. On the other hand, when formation spots move at a higher speed, the robots are able to maintain their positions within the occupied spot. It is possible to observe a waving pattern on the robots motion, identified in red on figure 3.7c. This type of behavior is caused by the combination of two facts: (i) actuators present in simulation are based on real robots' propulsion system, which is unable to efficiently work at low speeds and precisely correct the trajectory, and (ii) a water current is present, leading the robots to drift in a specific direction.



(A) t=0 (B) t=970 (C) t=1999

FIGURE 3.7: An example of the swarm accomplishing a randomly generated formation with both rotation and translation movement, using a monolithic controller. The black lines represent the robots' trajectory. The yellow circles represent the formation unoccupied spots, while the green circles represent the spots occupied by a robot. From left to right: figure 3.7a - status at the beginning of the simulation (timestep=0), figure 3.7b - first time formation accomplishment (timestep=970), and figure 3.7c - status at the end of the simulation (timestep=1999)

3.5 FSM-based control approach

On a second set of experiments, FSM-based control was used. The accomplishment of a formation by a set of robots implies for two main behaviors: (i) for each of the robots to reach a spot, and (ii) for the robots to maintain their position inside the occupied spot. Taking advantage of *task decomposition*, the second control approach consisted in the synthesis of control for the two behaviors. A Finite state machine (FSM) arbitrator (figure 3.8) is then used, and it is responsible to select between two available sub-controllers. Each of these sub-controllers consists of an ANN, and the decision on the sub-controller to be used is made taking in account the robot's position, *i.e.* if it is occupying a formation spot.

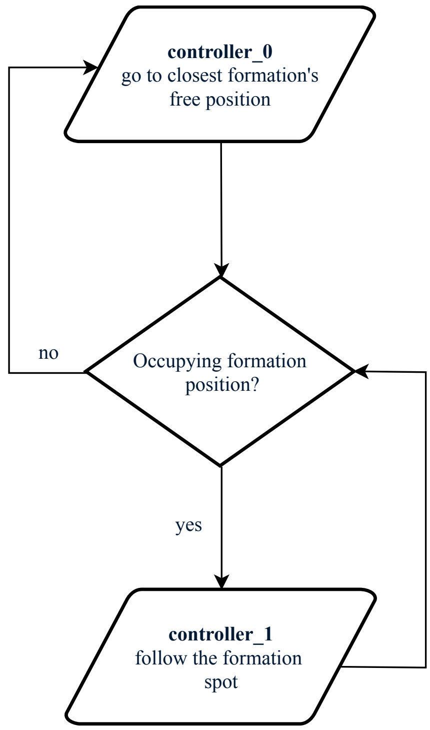


FIGURE 3.8: A schematization of the top-level FSM arbitrator. When the robot is not occupying a formation spot, *controller*_0 provides guidance for the robot to reach the closest free formation's spot. As soon as the robot enters the boundary of a free formation's spot *controller*_1 is activated, aiming to maintain the robot withing the occupied formation's spot boundary as it moves in the environment.

3.5.1 Evolutionary setup

In order to make possible to compare both control approaches, the evolutionary setup of the FSM-based control approach was similar to the one previously described in subsection 3.4.1. Each robot should occupy a formation's spot, as there is the same quantity of robots and spots. The same elitist approach previously described in section 3.4.1 was on genomes evaluation. Evolution was stopped at 250 generations, as fitness values started to stabilize. The experimental setup was repeated for a total of ten independent evolutionary runs and the synthesized controllers were also subjected to a post-evaluation, in order to obtain a more accurate estimate on how the controllers' perform the desired task. The controllers evaluation was conducted using the fitness function previously described in subsection 3.4.1 in equation 3.5.

In this control approach a FSM arbitrator (figure 3.8) selects the controller to be used among two available ANN controllers. Neuroevolution of augmenting topologies (NEAT) generational algorithm was used to synthesize *controller_0* ANN, while *controller_1* was previously evolved and used in several other scientific experiments (*homing* task in Duarte et al. [24, 27]) and proved that it is able to successfully transfer to real-world conditions.

In order to increase the success on the controllers' transferability from simulation to real-world conditions, noise introduction technique was used during controllers evolution. A list of the noise parameters used in the evolutionary process is summarized in table 3.3.

3.5.2 Results and Discussion

3.5.2.1 Performance

The normalized fitness scores from the highest performing controller of each of the independent evolutionary runs are summarized in figure 3.9, for each of the controllers type. The best run fitness value is represented by the red line. Since the evolutionary process of the FSM-based control did not present significantly improvements after 150 evolutionary generations, the evolutionary process was stopped at the 250^{th} generation. Through figure 3.9b, we are able to verify that the FSM-based control evolutionary process is more efficient, as a higher fitness value is achieved earlier and with a lower standard deviation, when compared with the monolithic control approach represented in figure 3.9a. This same observation is also supported by figure 3.10, where is possible to compare the fitness of the evolutionary setup of both controller's types.

In order to be possible to compare both control approaches, there were collected the previously identified four metrics on 4000 time steps long simulations. In figure 3.11 is possible to observe the collected data. Through *Number of different formation spots occupied by a robot* metric, we can observe that robots occupy 1.195 ± 0.073 different spots in average, meaning that the robots have a high tendency of entering and following an unique formation's spot rather than change the occupied spot afterwards. For the formation to be accomplished for the first time

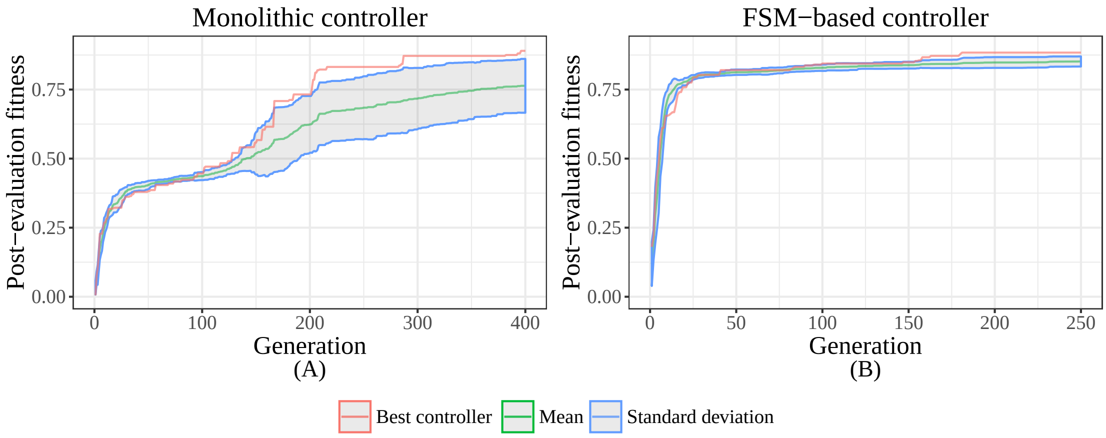


FIGURE 3.9: Comparison of both monolithic and FSM-based controllers fitness scores.

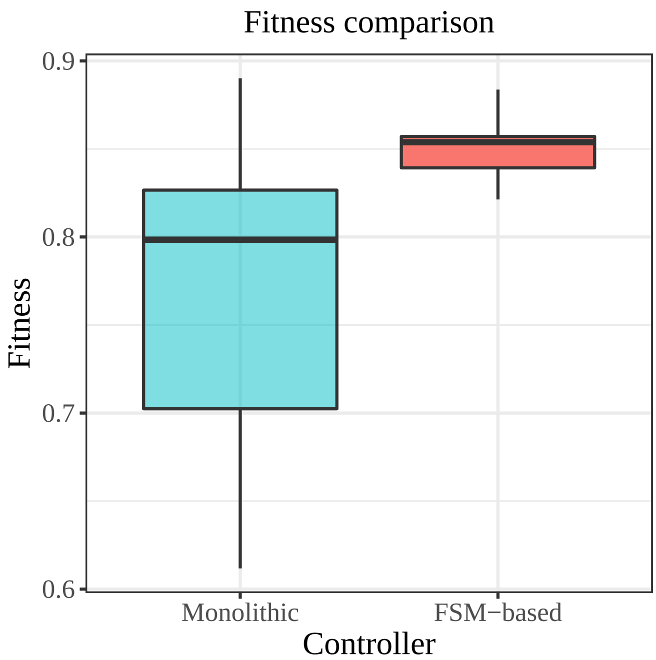


FIGURE 3.10: Fitness values for the highest performing controllers from each of the controllers' topologies.

in simulation, it takes in average 510 ± 77 simulation steps or 26% of the experiment duration, represented by *Time until first total formation occupation* metric. Last, robots spent in average 937 ± 17 simulation steps inside a spot (or 47% of the experiment time), represented by *Time of robots inside the formation spots* metric.

In figure 3.12, the average of each of the collected metrics are compared, per controller type. Through this figure it is possible to observe that the FSM-based control presents a lower tendency to occupy multiple formation spots, despite it takes longer to converge and occupy all the formation positions. Despite that fact,

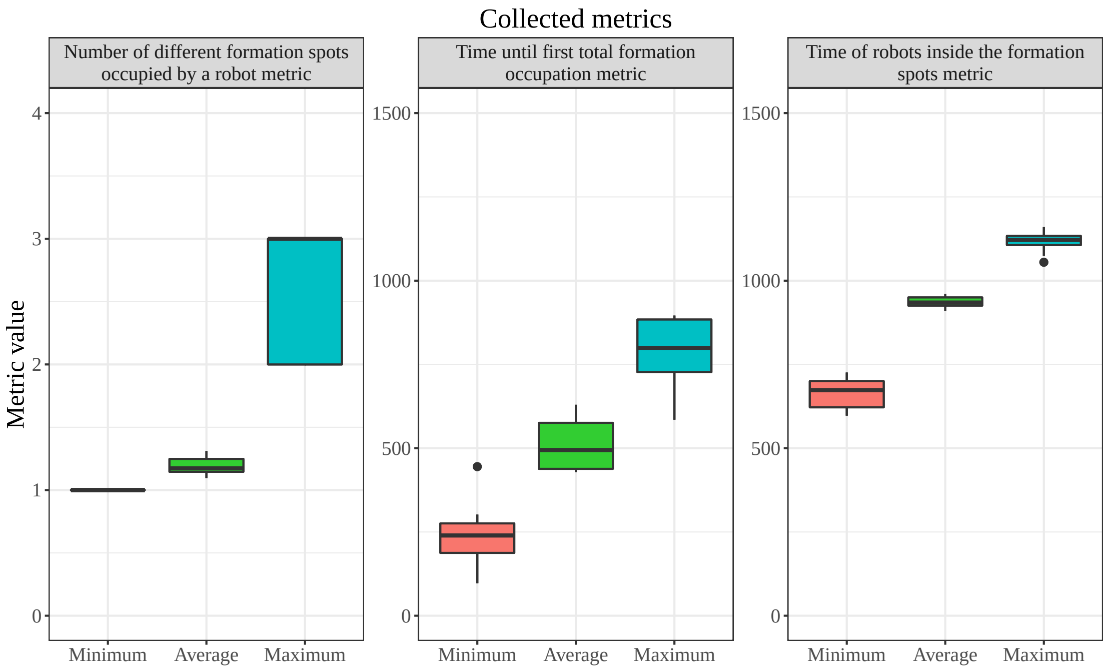


FIGURE 3.11: Plot of the metrics' minimum, average and maximum values. These are calculated from the runs' top controllers, in 4000 simulation steps long experiments.

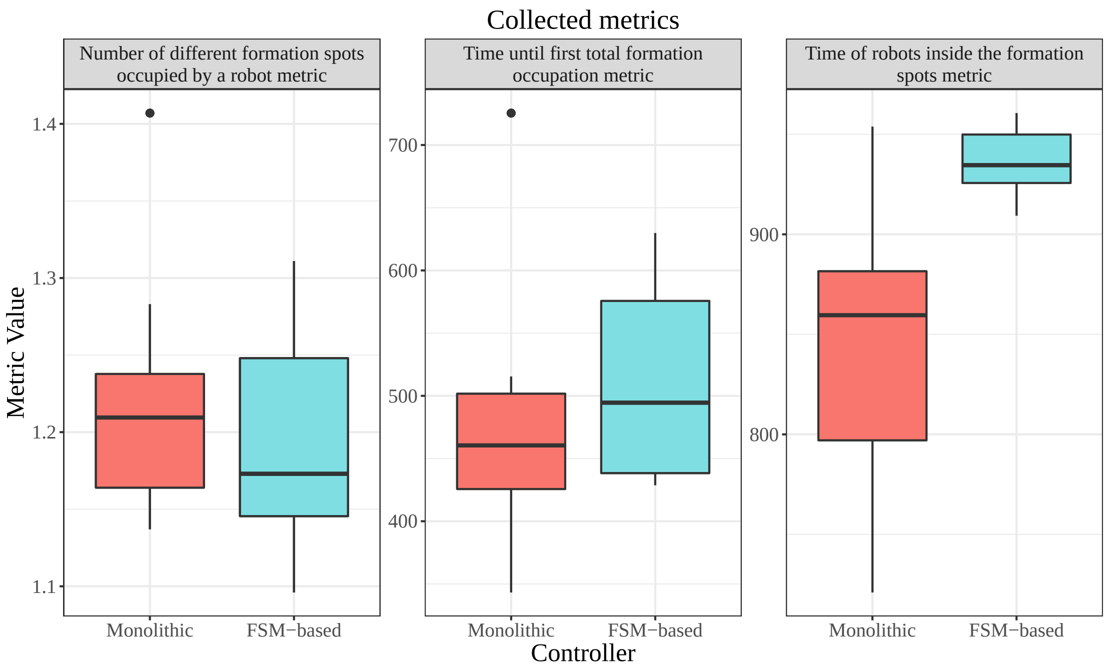
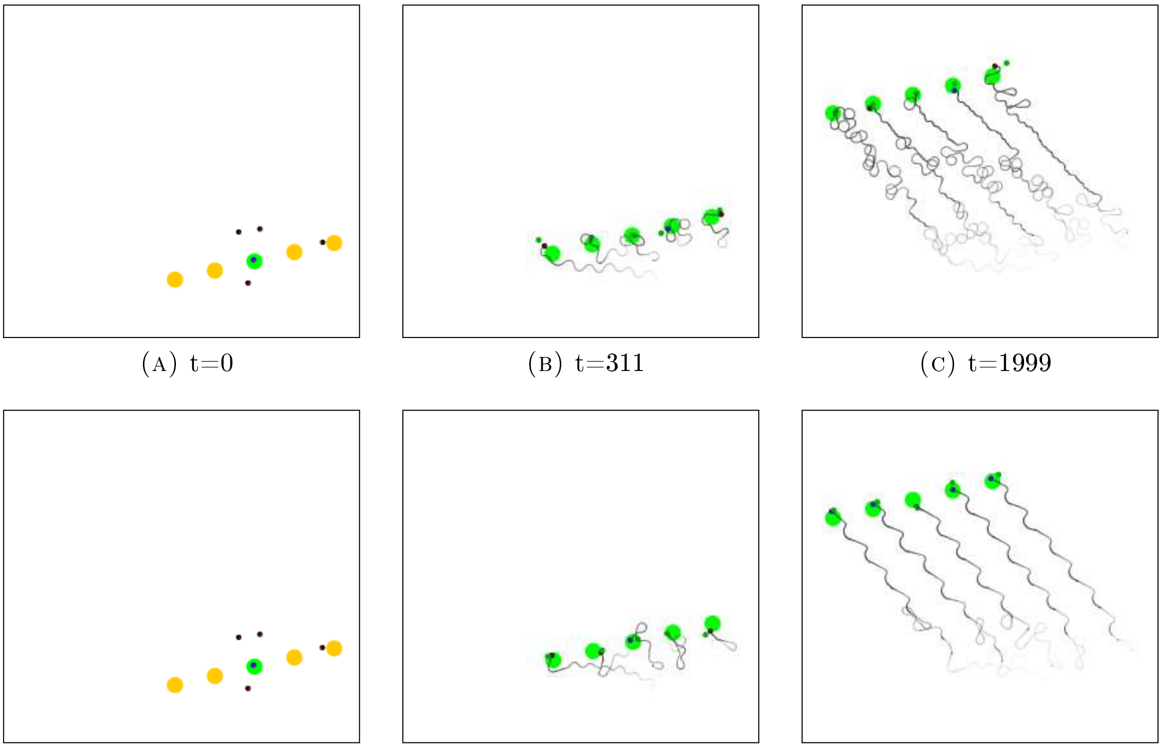


FIGURE 3.12: Plot of the metrics' average, per controller type. These are calculated from the runs' top controllers, in 4000 simulation steps long experiments.

the FSM-based control presents a higher value of time spent inside a formation' spot.

3.5.2.2 Behavior

In figure 3.13 it is possible to observe an example of a formation accomplishment, for each of the control types. On the top line, figures 3.13a, 3.13b and 3.13c demonstrate the behavior of the simulated robots with monolithic control, while on the bottom line ones FSM-based control is used. It is possible to observe that the FSM-based control provides a faster convergence of the robots to a position in the formation (t=294 *vs.* t=311). Comparing the robots trajectory it is also possible to observe that the FSM-based control provides a steadier guidance once the formation spots are occupied (figure 3.13f *vs.* 3.13c). The steadier control may conduct to a better performance and efficiency, since the energy consumption is lower when compared with the robots with monolithic control.



(A) t=0 (B) t=311 (C) t=1999

(D) t=0 (E) t=294 (F) t=1999

FIGURE 3.13: Comparison of the swarm behavior when accomplishing a line of bearing formation. In the top line images, a monolithic control is used, and on the bottom line control is provided by FSM-based control. The black lines represent the robots' trajectory. In order, from left to right: figures 3.13a and 3.13d - robots' positions and formation position at the beginning of simulation (timestep=0), figures 3.13b and 3.13e - first time formation accomplishment for each of the control types (timestep=311 for monolithic and timestep=294 for FSM-based control), and figures 3.13c and 3.13f - robots and formation positions at the end of the simulation (timestep=1999)

3.6 Robustness study

On a final set of experiments, robustness and fault-tolerance properties of both of the control types was assessed. While the simulation takes place, faults are injected. As one of the robots is disabled, another robot should take its place in the formation. For that purpose, the number of formation spots is lower that the number of robots in the swarm, so there are spare robotic units.

3.6.1 Experimental setup

The objective of this set of experiments was to assess the robustness and fault-tolerance of both types of control previously described. For that purpose, the previous evolved controllers were subjected to a post-evaluation where intermittent faults were injected. A common fault observed in real hardware was the clogging of one or both motors within a robot. We aimed to replicate such failure condition in the simulated robots by applying a temporary fault to one of the robots that is occupying a formation spot, at a random time step. During the fault condition period, the robot is unable to move during 500 simulation steps, after which the robot recovers its functional condition.

In order to evaluate the performance of the controllers in collectively detect and replace the faulty robot within the formation, a specific metric was used. *Time until a formation spot is reoccupied* metric allowed for the measurement of the controllers' efficiency in replacing the damaged robots, by measuring how much time a formation's spot takes to be reoccupied, after being unoccupied by the faulty robot.

3.6.2 Results and Discussion

3.6.2.1 Performance

For each of the tested control types, *Time until a formation spot is reoccupied* metric was collected. It is possible to verify through figure 3.14 that the time until a target is reoccupied when a fault occur is similar for both of the control types. Using monolithic control, the swarm takes an average of 117 ± 41 simulation steps or 6% of the experiment duration, considering 2000 time step long experiments. When FSM-based control is used, the swarm takes an average of 121 ± 63 simulation steps or 6% of the experiment duration. A small subset of the results present a minimum value near to zero, i.e. the swarm takes virtually no time to replace a faulty robot. These situations occurs when the spare robot is near the spot to be occupied, leading to a small time of reoccupation. This is possible to take place (i) when the fault injection occurs early in the simulation, and all the robots are near the targets after the initial convergence to occupy the same, or (ii) situations when the spare robot is near the spot to be occupied, leading to a small time of reoccupation.

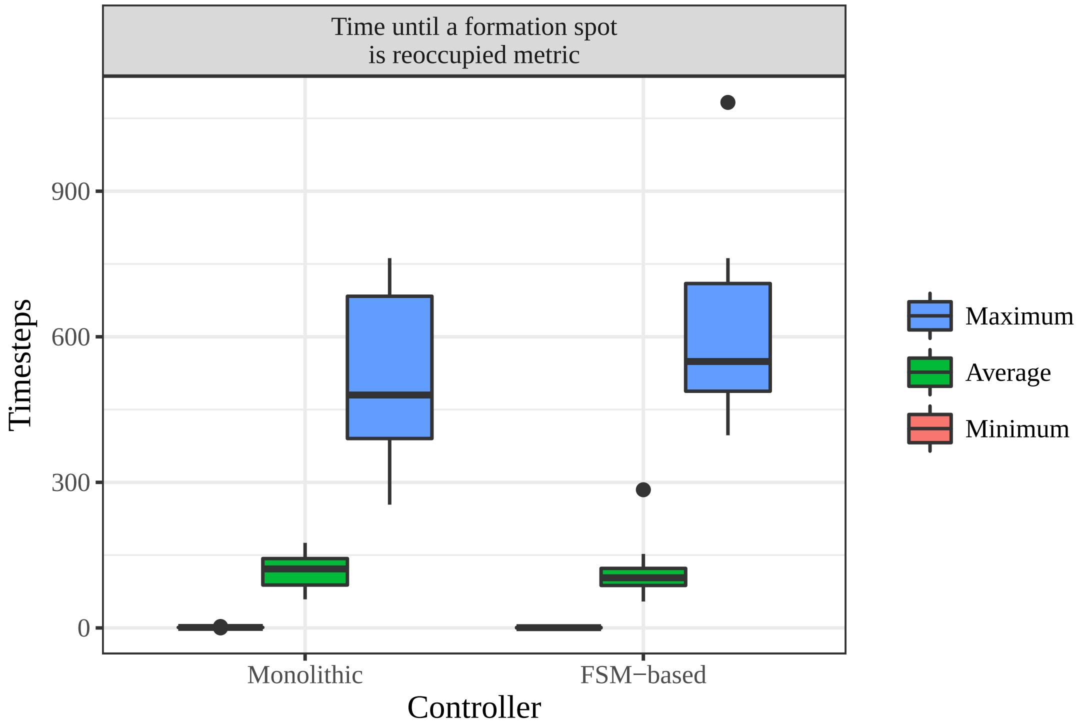


FIGURE 3.14: Plot of the *Time until a formation spot is reoccupied* metric minimum, average and maximum values. These are calculated from the runs' top controllers, in 4000 simulation steps long experiments.

In figure 3.15, it is also possible to observe how the remaining metrics vary when faults are injected, in comparison to the situation where no faults occur. The most notorious differences occur on *Time of robots inside the formation spots*, since robots spend less time inside a formation spot boundary on fault injected runs, as expected.

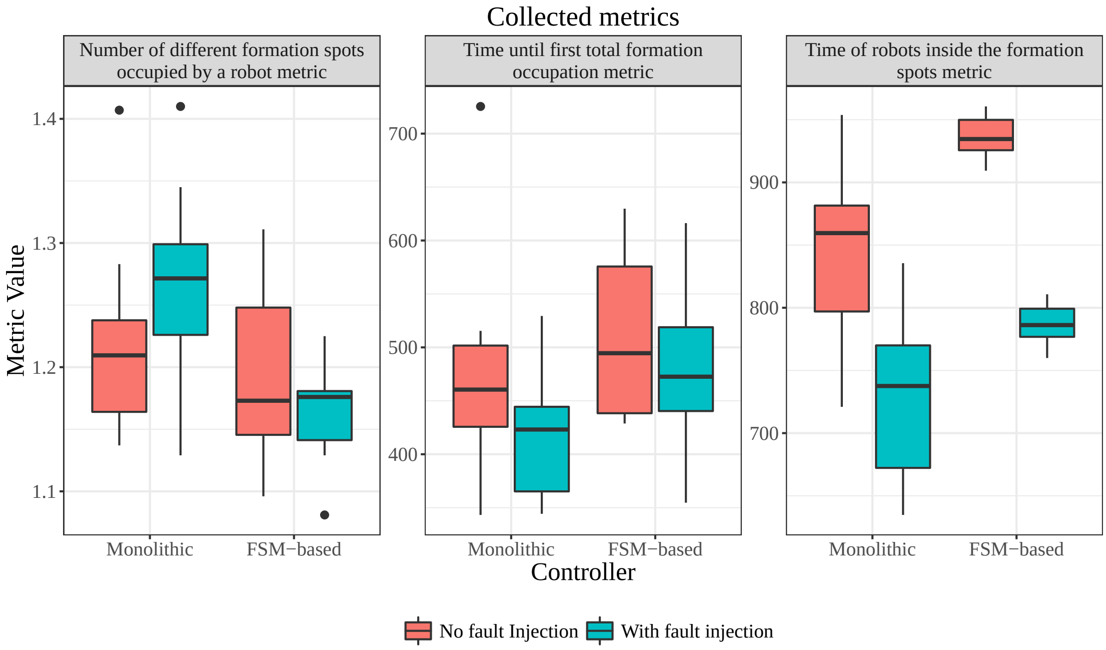
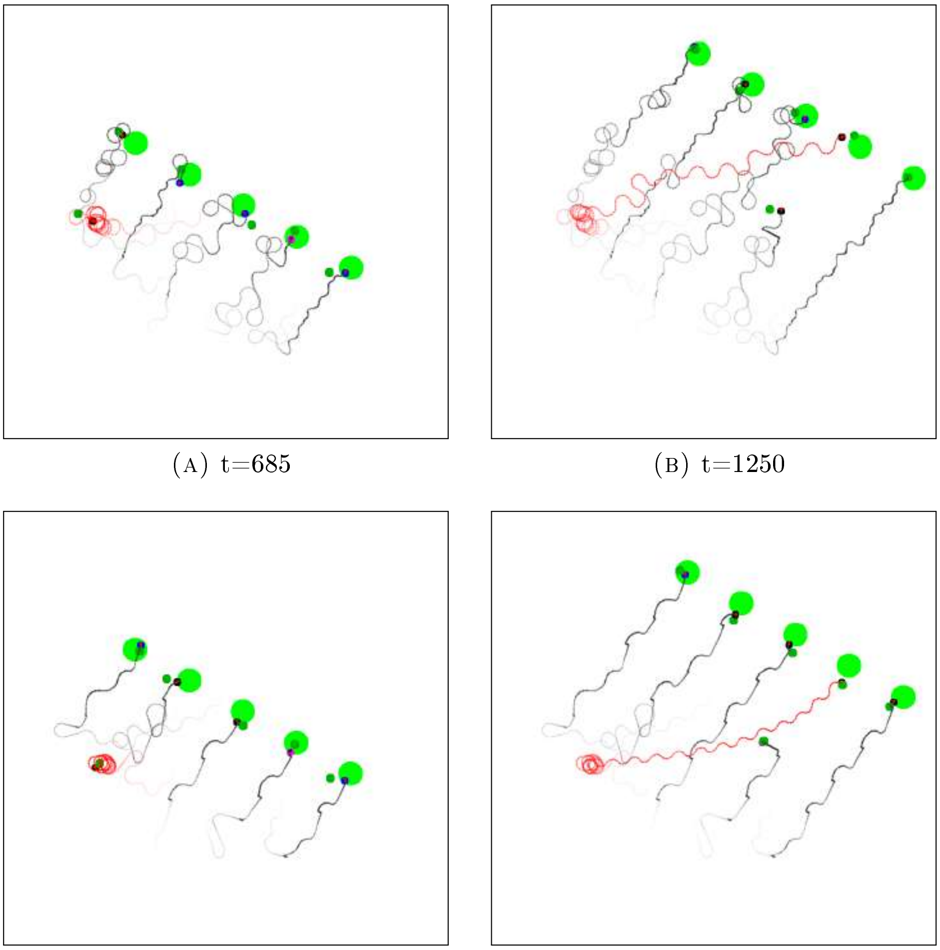


FIGURE 3.15: Plot of the average metrics values per controller type, with and without fault injection. These are calculated from the runs' top controllers, in 4000 simulation steps long experiments.

3.6.2.2 Behavior

In figure 3.16 it is possible to observe an example of the swarm's behavior when a fault takes place, as well as how the swarm recovers from the fault condition. Initially a fault is applied to one of the robots occupying a formation spot (figure 3.16a and 3.16c). The spare robot then detects the just released spot and moves towards it, occupying the empty formation spot (figure 3.16b and 3.16d). The path made by this robot from a standby position towards the formation's spot is identified in red.



(A) t=685 (B) t=1250

(C) t=685 (D) t=1170

FIGURE 3.16: Comparison of the robots behavior when accomplishing a line of bearing formation. In the top line images, a monolithic control is used, and on the bottom line FSM-based control is used. The black and red lines represent the robots' trajectory. In order, from left to right: figures 3.16a and 3.16c - moment in which one of the robots enters in a fault condition, (timestep=685), and figures 3.16b and 3.16d - moment in which the free formation spot is reoccupied (timestep=1250 for monolithic and timestep=1170 for FSM-based controller)

3.7 Summary

In this chapter, we described two formation control synthesis approaches, making use of evolutionary techniques, for a SRS. The first control synthesis approach made use of a monolithic artificial network controller, while the second used FSM-based control. To perform the desired task, the robots should occupy the different spots of a formation and keep their position within the spot boundaries, as the formation moves in the environment.

We first demonstrated that both control approaches make possible the formation maintenance, achieved in an autonomous manner, as robots coordinate among themselves to occupy the various spots in the formation. We then compared both approaches and verified that the hierarchal control synthesis conducts to a quicker evolutionary process, as well as the defined metrics present a more fruitful result when compared with the monolithic approach. The FSM-based control tends to maximize the occupation time, as well as produce a more steadier guidance. Finally, the robustness and the fault tolerance of both of the controllers' types was assessed. In this setup, fault injection was used. Through the results we observed that the swarm is able to detect the fault as a system and to replace the damaged robot.

Chapter 4

A New Aquatic Platform for Swarm Robotics Experiments

In this chapter, we provide a technical overview of the design and development process of a SRS platform composed of small and inexpensive autonomous surface vessels (see Figure 4.1). The aim of the platform was to enable swarm robotics experiments outside of a laboratory environment [24]. In order to develop a SRS, several constraints have to be taken into account. For instance, in order to make deployment of large swarms viable, the cost of each individual unit must be kept low, which implies that robots must be kept relatively simple. The design of our system was based on the following four objectives:

1. The solution should be a low-cost robotic platform. This was achieved through the use of inexpensive off-the-shelf and widely available components, as well as through the use of digital fabrication processes.

2. The solution should allow for easy logistics, namely transportation and deployment. This was achieved through the design of small and compact units (65 cm length by 40 cm wide).

3. Each robotic unit should be capable of autonomous decision-making. This was achieved through the inclusion of on-board processing, communication, and sensing.

4. The system should provide a human-machine interface that allows an operator to monitor and supervise a swarm of aquatic robots. This was achieved through the development of an easy-to-use command and control console.

FIGURE 4.1: A swarm of eight robots (out of a total of ten developed) at Parque das Nações, Lisbon, Portugal.

The developed robotic platform is versatile and customizable, and all hardware specification and designs, as well as all software modules, are made available as open-source under the GNU LGPLv3 license, enabling replication and extension by third parties. The total cost of each unit is approximately 300 EUR in materials. To facilitate studies on control synthesis and swarming behavior for real-world robotic systems [28], we combined the robotic platform with our simulation framework, JBotEvolver [137].

This chapter is based on an internationally published scientific article [22], which sums all SRS platform design and development process.

In the following sections, we provide an overview of the design and manufacturing of our robotic units (Section 4.1), a description of the onboard hardware (Section 4.2), and software (Section 4.3). Finally, Section 4.4 contains some concluding remarks.

4.1 Hull design

For our robotics units, we opted for a monohull-shaped vessel (see Figure 4.3), which is machinable from a single block of raw material. The robots are relatively small (L 65 cm × W 40 cm × H 15 cm), and light (3 Kg). While we have used low-cost Computerized Numeric Cut (CNC) and 3D-printing fabrication processes and materials, the open-source nature of the platform allows for different fabrication processes, such as casting. Our platform can furthermore be adapted to support different sensors payloads and actuators.

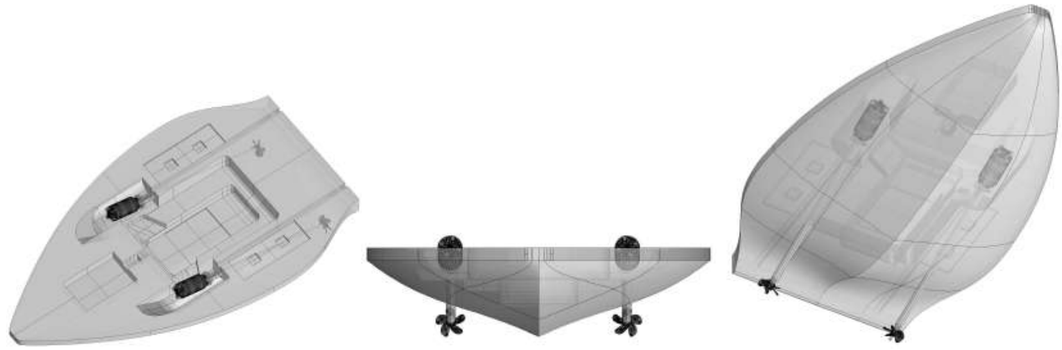
4.1.1 Fabrication Process

We designed the hull and support parts in computer-aided design (CAD) software (*Rhinoceros 3D*), which were then produced using digital fabrication techniques. The hulls were milled using an *Ouplan 3020* CNC machine, and 12 support parts were produced using a *BQ Prusa i3 Hephestos* 3D printer. The use of digital design, modeling, and fabrication processes allowed us to quickly iterate and optimize the hull and the support parts designs, and to have a short and inexpensive design-to-product cycle. In total, we produced 19 different hulls, 9 of them prototypes, and 10 operational units. In figure 4.2, it is possible to observe the various hull designs produced, including the final one on the figure's top right.

FIGURE 4.2: Different hulls produced during the design-to-product cycles. In this photography it is possible to identify the 9 different hull prototypes and the final design, on the top right.

4.1.2 Materials

We used extruded polystyrene foam (XPS) for the hull production since it is buoyant, easily machinable, and inexpensive. This material can also be hand worked, allowing for manual shaping and finishing. The 12 support parts were 3D-printed in Polylactic Acid (PLA), which is an inexpensive biodegradable thermoplastic. The 3D-printed parts were installed in the hull using silicon-based glue in order to support the different hardware component, such as motors, shafts, enclosures, and sensors. The shaft support design (see Figure 4.4) allows for a quick motor and shaft replacement, reducing the repair time in case of motor breakdown. The



(A) Top view (B) Front view (C) Bottom view

FIGURE 4.3: CAD model of the designed hull

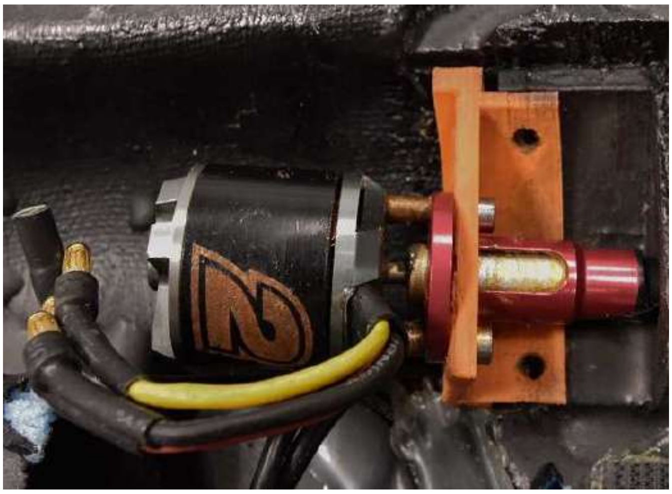


FIGURE 4.4: Detachable motor support. The orange piece supports both the motor and the shaft, and allows the module to be detached from the hull support (in black).

final batch of robots were coated in black epoxy resin and fiberglass in order to increase strength and robustness, and to waterproof the hull.

4.2 Electronics and propulsion

Maritime environments represent a challenge for roboticists: the vessel's exposure to harsh environmental elements, such as solar UV-light, heat, and salt water, requires a high degree of isolation for sensitive components. Most of the electronic components were therefore housed in one of two enclosures.

4.2.1 Enclosures

We used 2.5 L (*main enclosure*) and 0.24 L (*secondary enclosure*) plastic containers to house all the electronic components and circuitry. We found that this inexpensive and flexible solution presents a degree of protection similar to IP67 standard, therefore fitting our needs. The main enclosure contained the power source, along with main processing and sensing components. The secondary enclosure contained two diagnostic LEDs to facilitate immediate status reporting, and sensors that needed to be isolated from electromagnetic interference from the motors or other components in the main enclosure. The connections between enclosures, and between components inside and outside enclosures, were made through IP68-rated cable glands. In order to minimize equipment overheating, the main enclosure was covered with aluminum tape to reflect sunlight.

4.2.2 Propulsion

Several experiments with different propulsion options were conducted, including experiments with turbines and inboard motors. The turbine system, based on *EDF Ducted Fan Unit 6 Blade 66 mm*, despite fast and efficient, proved prone to motor oxidation and debris entanglement. We therefore opted for a differential propulsion system composed of two motors coupled to a 4 mm drive shaft with a 3-blade 28 mm propeller. The drive shaft ran on a 255 mm length shaft sleeve filled with lithium-based grease. This solution was chosen for the final batch of operational units. Two different motor models were used: (i) *NTM Prop Drive Series 28-30A 750 kv/ 140 w* and (ii) *Emax 2215/25 950 kv 2-3S*. Each motor is driven by a *HobbyKing 50 A Boat ESC 4 A UBEC* electronic speed controller (ESC), which present a current limit nearly twice the one necessary, therefore decreasing chances of equipment overheating while providing good compatibility with the motors used. The ESCs were installed outside on the bottom of the main enclosure. This propulsion setup enabled the final batch of robotic units to move at speeds up to 1.7 m/s (3.3 kts), to achieve turning rates of 90 °/s, and to accelerate to full speed in one second.

4.2.3 Energy

Energy was provided by two batteries, both located in the main enclosure: (i) a unit that powers all the equipment related with motors and propulsion (*motor* *battery*), and (ii) a unit to power control, processing and sensing components (*control battery*). We conducted experiments with both lithium-polymer (LiPo) and lithium-iron-phosphate (LiFePo4) batteries. LiPo batteries were chosen for the final iteration of the platform due to their lower price and relatively higher power density. For the motor battery, we chose a *ZIPPY Flightmax 8000 mAh 3S1P* battery, which provided an autonomy between 1h30m and 4h30m depending on motor usage. The control battery used was a *ZIPPY Flightmax 5000 mAh 3S1P*, which supplied power to all the remaining components through a *Turnigy 5A* (8-26 V) switched battery eliminator circuit (SBEC), that regulates and stabilizes the battery voltage to 5 VDC. The control battery provided a run time of approximately 4h30m.

4.2.4 Computation & Communications

Onboard computation was provided by a Raspberry Pi 2 single-board computer (SBC). The Raspberry Pi 2 is composed of a quad-core ARM Cortex-A7 CPU clocked at 900 MHz, 1GB RAM, 4 USB ports and 40 general purpose input/output (GPIOs) pins supporting diverse protocols such as UART, I^2C, SPI and One-Wire, which facilitates integration with different electronic components and modules. The SBC is located in the *main electronics enclosure* and is connected to the remaining components through a custom breakout cable. In order to enable communication between neighboring robots, we included a wireless communication system using a TP-Link TL-WN722N High-Gain Wi-Fi adapter, connected to the SBC through an USB interface. The adapter was coupled to a monopole 4 dBi gain antenna, providing an effective communication range between neighboring robots of 40 m on the water surface.

4.2.5 Sensors

Various sensors were included in each robot, namely a GPS receiver, a digital compass unit, and a temperature sensor. Global position information was provided by an *Adafruit Ultimate GPS Breakout*, based on *GlobalTop FGPMMOPA6H GPS Standalone* module [142], which was placed in the main enclosure. This module is a 66 channel GPS receiver providing position updates with a 5 Hz frequency, and interfaced with the SBC through the UART protocol. It was coupled with an active 26 dB gain GPS antenna, increasing the received signal quality and providing positioning information with a ± 3 m accuracy.

Heading information was provided by a *STMicroelectronics LSM303D* magnetometer, which interfaced with the SBC through a standard I^2C protocol [143]. This unit contains both a triple-axis magnetometer and a triple-axis accelerometer, allowing for the compensation of the magnetic readings according to the pose of the robot. The location of the sensors in the vessel was also subject to experimentation, since we verified that high current wires, motors, and batteries interfered with the magnetic field readings. Therefore, we installed the magnetometer in the secondary enclosure, which was located in the prow of the vessel.

Finally, temperature information was provided by both the on-board SBC temperature sensor and by a waterproof *Maxim DS18B20* sensor [144]. The first sensor was used to monitor the conditions inside the main enclosure. The second sensor, positioned in the bottom of the vessel, was used to measure the water temperature. This latter is a digital 12-bit resolution temperature sensor, which gives readings in 0.0625°C increments and has an error of ±0.5 °C. This unit has an update frequency of approximately 1.25 Hz and interfaced with the SBC through a One-Wire standard protocol.

The location of the electronic and propulsion components on board each of the robots can be found in Figure 4.5, and a summary of all the components can be found in Table 4.1.

4.3 Software

The software that enables the control and monitoring of the robotic platform is divided into three different elements:

- An onboard software component, responsible for the control and management of each robotic unit (*Raspberry Controller*);
- A console that enables command and control of the swarm by a human operator (*Control Console*);
- An API layer, which makes the use of simulation or the real robotic hardware transparent to the robotic controller (*Common Interface*).

4.3.1 Onboard Software

The Raspberry Pi 2 SBC runs a Raspbian Jessie 8_4.4.9-7+ Linux operative system, which is based on Linux Debian Jessie 8 distribution compiled for ARM

Component	**Make and Model**
Enclosures	
Main enclosure	2.5 L watertight plastic box
Secondary enclosure	0.24 L watertight plastic box
Propulsion	
Motor (A)	NTM Prop Drive Series 28-30A 750 kv/ 140 w
Motor (B)	Emax 2215/25 950 kv 2-3S
Shaft	4 mm drive shaft
Shaft Sleeve	255 mm length shaft sleeve
Propeller	3-blade 28 mm propeller
ESC	HobbyKing 50 A Boat ESC 4 A UBEC
Power	
Motor battery	ZIPPY Flightmax 8000 mAh 3S1P
Control battery	ZIPPY Flightmax 5000 mAh 3S1P
SBEC	Turnigy 5A (8-26 V)
Computation & Communications	
Single board computer	Raspberry Pi 2
Wi-Fi Adapter	TP-Link TL-WN722N
Sensors	
GPS	Adafruit Ultimate GPS Breakout
Compass	STMicroelectronics LSM303D
Water Temperature Sensor	Maxim DS18B20

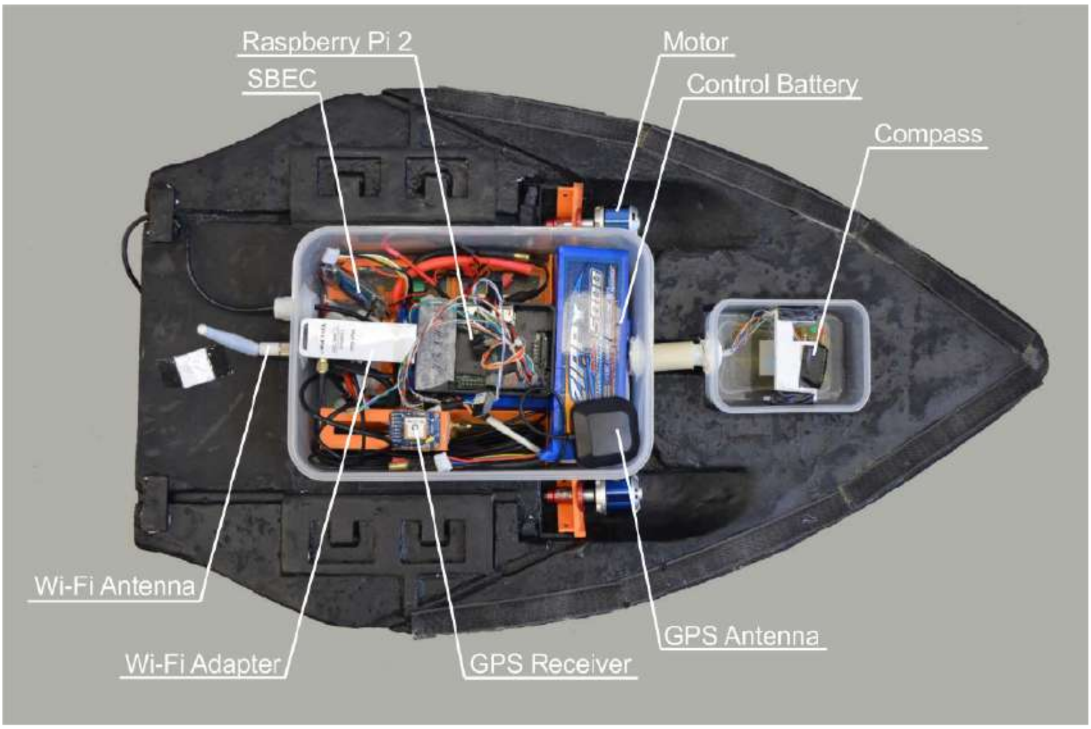
Table 4.1: Robotic units components

architecture and with hard-float support. In order to interact with the different hardware components, we used several existent open-source software components. A guide on how to replicate the robot's software system configurations can be found in our team's GitHub page.[1]

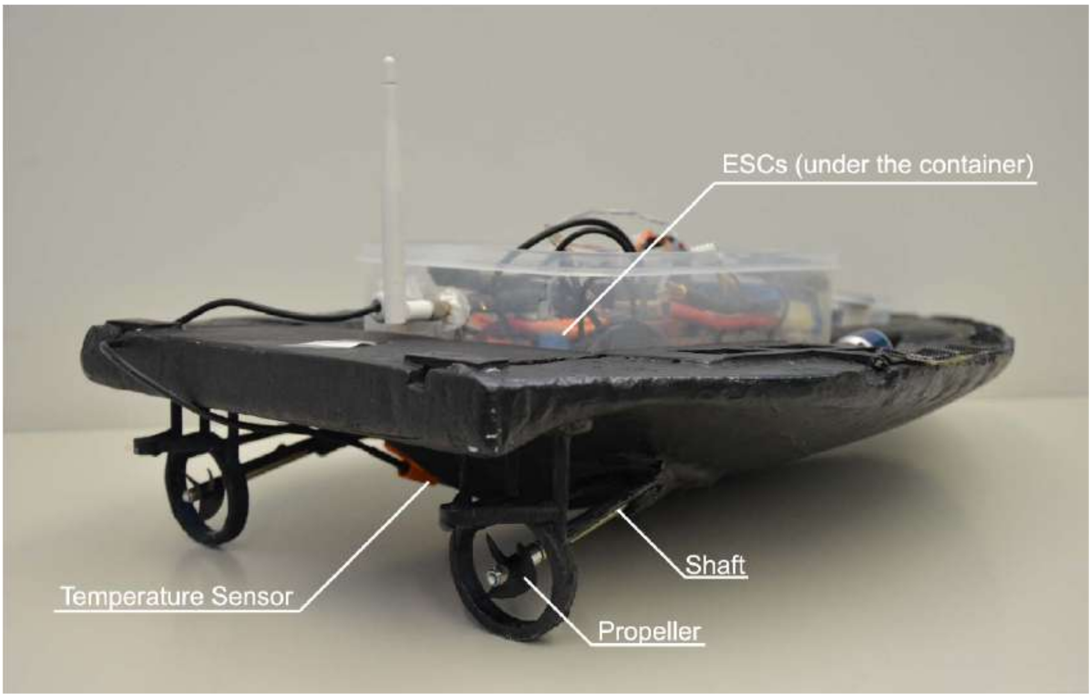
The Raspberry Controller is the Java-based software running on board each robot. This software is responsible for interacting with all sensors and actuators, executing the behavioral control logic, and for communicating with nearby robots and the control console. It relies on the Pi4J library to interact with the hardware components, except for the interaction with ESCs, which is achieved using the ServoBlaster kernel module. The source code for our Raspberry Controller software is available under open-source license.[2]

ServoBlaster[3] is a kernel module that enables the generation of pulse position modulated (PPM) signals through the Raspberry Pi's GPIOs. This modulation

[1] `https://github.com/BioMachinesLab/drones/wiki`
[2] `https://github.com/BioMachinesLab/drones/tree/master/RaspberryController`
[3] `https://github.com/richardghirst/PiBits/tree/master/ServoBlaster`



(A) Top view



(B) Side view

FIGURE 4.5: Robotic unit components

enables the transmission of position information encoded in temporal pulses [145], the signal necessary to control the ESCs used in our robots.

We use WiringPi v2.25 C library[4] to manipulate the GPIO and to interact with the different sensors. To the access the WiringPi C library's methods from the onboard software, we use Pi4J 1.1-SNAPSHOT library.

The communication between a human experimenter and the swarm is performed through an ad-hoc *IEEE 802.11g* wireless network. An *Ubiquiti BULLET-M2-HP* running *OpenWrt Chaos Calmer 15.05 r46133* firmware[5] with LuCI Configuration Interface coupled to a 12 dBi gain monopole antenna is installed at the base station. The setup provided a communication range of 150 m between the base station and the robots operating on the water surface. Two pieces of information are broadcast by standard on the network using UDP messages, namely the robot's GPS position and keep-alive messages. When reliability is required, such as when a robotic unit is teleoperated by an operator or when new control logic is uploaded, TCP/IP connections are used.

In order to reduce the noise fed to the ANN controllers, a Kalman filter [146] is applied to GPS receiver and compass sensors readings, running at the on-board software. Before the experiments execution, a compass calibration routine is also run, assuring that the readings are standard and normalized.

To make transparent for controllers the transference from simulation to real hardware, position information is translated to a Cartesian grid in the *common interface*, previously described in subsection 4.3.3. During operation, the geographic position of our experimentation area's center (38 °45 '57.9 "N 9 °05 '36.5 "W, in case of Parque das Nações experiments) is mapped into the Cartesian grid center (0,0).

4.3.2 Control Console

For command and control, we developed a stand-alone multi-platform desktop application (see Figure 4.6). This application[6] enables the experimenter to control and monitor a swarm of aquatic robots. Each unit's location and heading is displayed on a map. Additional telemetry information can be displayed when required, along with data collected by the on-board sensors. The robots' on-board control logic can furthermore be updated through the console, and various spatial entities can be configured and deployed to specific robots, such as waypoints, geofences, and the location of obstacles to avoid. The software generates log files of the commands sent to individual units along with all broadcasted messages

[4] `http://wiringpi.com/`
[5] `https://openwrt.org/`
[6] `https://github.com/BioMachinesLab/drones/tree/master/DroneControlConsole`

that enable off-line replay of the experiments and facilitate off-line debugging and data extraction. Multiple instances of the control console can be executed simultaneously, providing control redundancy and allowing for multiple operators.

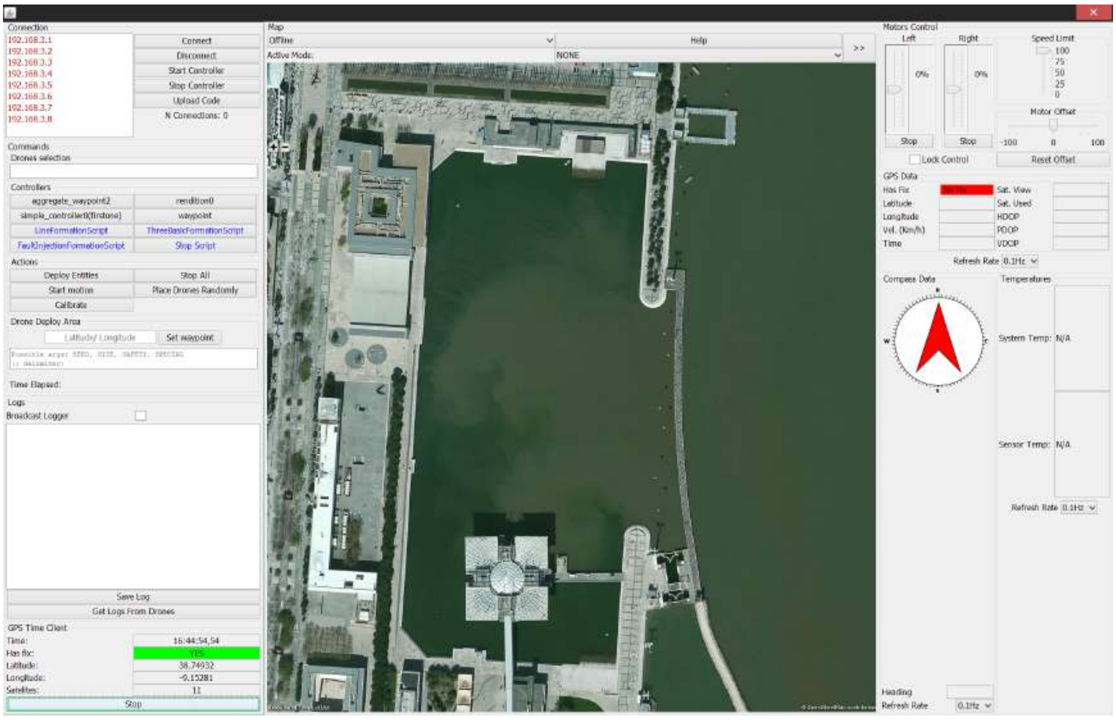


FIGURE 4.6: A screenshot of the control console.

In order to assure correct synchronization between both robotic units and control console instances clocks, all clocks are synchronous with GPS time. This strategy, enable us to correlate log files from multiple sources using time information. A Java-based time server was also developed (see figure 4.7), providing time information to all control station instances.

4.3.3 Common Interface

We developed a *common interface* API layer, which provides source code level compatibility between control logic executed in simulation and on the real robots. This component sits between the high-level control logic and the low-level hardware interface, facilitating the synthesis of control and its transfer from simulation to the real robots. The common interface was integrated with our simulator JBotEvolver [137] in order to synthesize self-organized swarm control, which was then transferred successfully to the real robotic swarm.

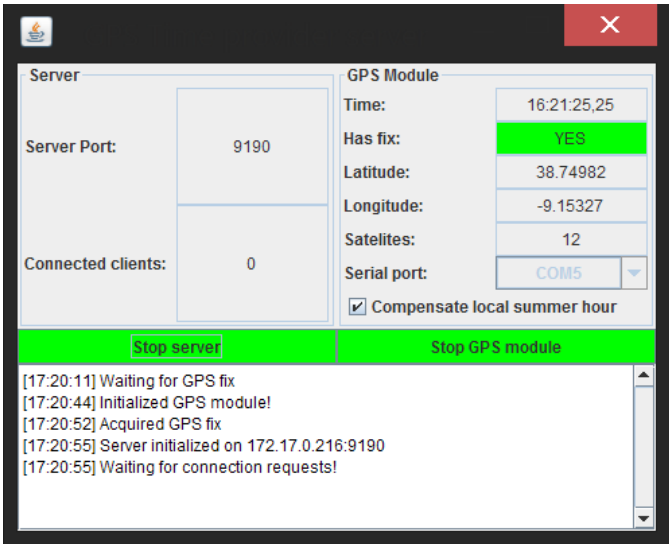


FIGURE 4.7: A screenshot of GPS time provider server.

4.4 Summary

In this chapter, we provided an overview of the developed robotic platform, demonstrating how the four key design objectives were achieved. Our solution represents a simple, inexpensive, flexible, and open platform for maritime swarm robotics studies, which can be extended and improved by third parties. The robotic platform was used in the conduction of perliminary experiments on formation control in real hardware. These experiments allowed to tune several simulation parameters, contributing to future demonstrations.

This same robotic platform was also used in the conduction of several scientific studies outside of the scope of this dissertation. Making use of it, we were able to demonstrate the successful transfer of evolved control from simulation to real hardware in a series of experiments [24]. In a first study [24], control was synthesized for four canonical swarm behavior tasks: (i) homing, (ii) dispersion, (iii) clustering and (iv) area monitoring. Afterwards, we experimented with a sequential composition of the different behaviors in an environmental monitoring task, where the robots had to navigate to a predefined area, disperse, cover the area while continuously collecting water temperature measurements, and finally aggregate and collectively navigate back to the base station [24, 26].

The described robotic platform also allowed to study the application of *hierarchical control synthesis* for SRSs [87]. We tested this approach on an intruder detection task with realistic constraints [27]. The robots had to monitor an area, detect, and follow any intruder that attempted to cross it and periodically recharge their batteries at a base station.

In the studies discussed above, the performance and behavior observed on the real robots was similar to the one observed in simulation [24, 27]. In this way, the robotic platform presented in this chapter facilitated novel contributions to the field of swarm robotics, and most notably, was used in the first successful demonstration of evolved control outside of strictly controlled laboratory conditions [24, 36], as well as to demonstrate the use of a SRSs in environmental monitoring task [26].

Chapter 5

Conclusions and Future Work

In this dissertation, we showed control synthesis on formation control for SRSs, making use of evolutionary techniques. The highly distributed and autonomous nature of SRS can be advantageous in many real-world maritime missions, and potentially enable completely new classes of tasks to be addressed. The use of formations in such systems allow for a better use of the robots' sensory capabilities, maximizing the coverage and improving efficiency.

In order to provide formation control, two different controller types were compared. A first approach made use of a monolithic controller, composed of a single ANN. The second approach made use of a FSM-based controller, composed of two ANN and a FSM arbitrator. The present study demonstrates that the monolithic control presents a less efficient evolutionary process, when compared with FSM-based control. As demonstrated, the controllers of the second type also provide a more energetic and motion efficient control, as the formations accomplishment occurs earlier. The time that the formation spots are occupied is maximized and the traveled path is steadier, when compared with the monolithic control. The use of *task decomposition* also presents an additional advantage, as the division of the main task in sub-tasks allows for sub-controllers simplification and specialization on specific sub-tasks.

With the objective of conduction real-world experiments on control synthesis for SRSs, we developed a SRS platform composed of small and inexpensive ASVs. We successful produced a total of 10 operational robotic units, in addition to 9 prototypes. This robotic platform and its associated software stack was used in the conduction of several scientific studies outside of the context of this dissertation. It also has the potential for the conduction of experiments on formation control transference to real-world conditions.

5.1 Future Work

The conducted work has the potential for extension in several different scientific areas. Apart from the conduction of full studies on the formation control on real-world conditions, there is also future work to be conducted both on the developed robotic platform and on the control synthesis.

5.1.1 Robotic Platform

In ongoing work, several potential improvements to the platform can be studied. In large-scale swarms, different robots might be equipped with different types of communication capabilities and serve as gateways for the rest of the swarm [28], or a few of the robots may be equipped with different sensors payloads and share information with the neighboring robots [23]. Such approaches can allow for the increase of the swarm's capabilities, while keeping the cost of the average robot low. In the ongoing work, the developed software stack can also be integrated with the ROS, the *de facto* standard by the robotics community [132].

5.1.2 Connection Mechanisms and Control for Self-assembling Surface Robots

As previously stated, SRSs are ideally suited for tasks where redundancy and large spatial coverage is required. While the capabilities of simple units tend to be limited, it has been shown that giving units the ability to form physical connections with one another, enables them to overcome the limitations of the individual units [48]. In aquatic environments, self-assembly is particularly challenging for two reasons: any connection mechanism must be robust enough to operate in harsh conditions, and the stochasticity of the environment requires new approaches to coordination of self-assembling robots. This line of future work consists of the development of novel connection mechanisms for self-assembling aquatic surface vessels and study control and coordination strategies for robotic swarms with self-assembly capabilities. The proposed research is a significant step towards a new class of marine robots that are able to change their size and shape on-the-fly.

Bibliography

[1] Nyo Me Tun and Thin Mya Mya Swe. Comparison of three pattern matching algorithms using dna sequences. *International Journal of Scientific Engineering and Technology Research*, 3(35):6916–6920, 2014. ISSN 2319-8885.

[2] B. Atal and L. Rabiner. A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(3):201–212, 6 1976. DOI: `10.1109/TASSP.1976.1162800`.

[3] Chi Hau Chen. *Handbook of pattern recognition and computer vision.* World Scientific, 5 edition, 2016. ISBN 9-814-65654-2.

[4] Rafał Dreżewski, Jan Sepielak, and Wojciech Filipkowski. System supporting money laundering detection. *Digital Investigation*, 9(1):8–21, 6 2012. DOI: `10.1016/J.DIIN.2012.04.003`.

[5] Seref Sagiroglu and Duygu Sinanc. Big data: A review. In *2013 International Conference on Collaboration Technologies and Systems (CTS)*, pages 42–47. IEEE, 5 2013. DOI: `10.1109/CTS.2013.6567202`.

[6] Stefano Marras, Shaun S. Killen, Jan Lindström, David J. McKenzie, John F. Steffensen, and Paolo Domenici. Fish swimming in schools save energy regardless of their spatial position. *Behavioral Ecology and Sociobiology*, 69(2):219–226, 2 2015. DOI: `10.1007/s00265-014-1834-4`.

[7] Federico S. Cattivelli and Ali H. Sayed. Modeling bird flight formations using diffusion adaptation. *IEEE Transactions on Signal Processing*, 59(5): 2038–2051, 5 2011. DOI: `10.1109/TSP.2011.2107907`.

[8] Akira Okubo. Dynamical aspects of animal grouping: Swarms, schools, flocks, and herds. *Advances in Biophysics*, 22:1–94, 1 1986. DOI: `10.1016/0065-227X(86)90003-1`.

[9] J K Parrish. Complexity, pattern, and evolutionary trade-offs in animal aggregation. *Science*, 284(5411):99–101, 4 1999. DOI: `10.1126/science.284.5411.99`.

[10] Paul J. Springer. *Military robots and drones : a reference handbook*. ABC-CLIO, 2013. ISBN 1598847325 9781598847321.

[11] Keiji Nagatani, Seiga Kiribayashi, Yoshito Okada, Kazuki Otake, Kazuya Yoshida, Satoshi Tadokoro, Takeshi Nishimura, Tomoaki Yoshida, Eiji Koyanagi, Mineo Fukushima, and Shinji Kawatsuma. Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots. *Journal of Field Robotics*, 30:44–63, 1 2013. DOI: `10.1002/rob.21439`.

[12] Brian M. Yamauchi. Packbot: a versatile platform for military robotics. In *Unmanned Ground Vehicle Technology VI*, volume 5422, pages 228–237. International Society for Optics and Photonics, 9 2004.

[13] Patrick Lin, George Bekey, and Keith Abney. Autonomous military robotics: Risk, ethics, and design. Technical report, U.S. Deparment of Defense, 2008.

[14] Kevin W. Williams. A summary of unmanned aircraft accident/incident data: Human factors implications. Technical report, U.S. Department of Defense, 2004.

[15] E. Garcia, M.A. Jimenez, P.G. De Santos, and M. Armada. The evolution of robotics research. *IEEE Robotics & Automation Magazine*, 14(1):90–103, 3 2007. DOI: `10.1109/MRA.2007.339608`.

[16] Fernando Silva, Luís Correia, and Anders Lyhne Christensen. Evolutionary robotics. *Scholarpedia*, 11(7), 2016. DOI: `10.4249/scholarpedia.33333`.

[17] Fernando Silva, Paulo Urbano, and Anders Lyhne Christensen. Online evolution of adaptive robot behaviour. *International Journal of Natural Computing Research*, 4(2):59–77, 4 2014. DOI: `10.4018/ijncr.2014040104`.

[18] O. Miglino, H. H. Lund, and S. Nolfi. Evolving mobile robots in simulated and real environments. *Artificial life*, 2(4):417–434, 1996. DOI: `10.1162/artl.1995.2.417`.

[19] Erol Şahin. Swarm robotics: From sources of inspiration to domains of application. In *Swarm Robotics*, volume 3342 of *Lecture Notes in Computer*

Science, pages 10–20. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. DOI: 10.1007/b105069.

[20] Estrutura de Missão para a Extensão da Plataforma Continental. MAP "PORTUGAL IS SEA", . URL https://www.emepc.pt/en/kit-do-mar/projets/mapa-2. Visited on 2018-04-17.

[21] Estrutura de Missão para a Extensão da Plataforma Continental. THE PORTUGUESE SUBMISSION, . URL https://www.emepc.pt/en/the-portuguese-submission. Visited on 2018-04-17.

[22] Vasco Costa, Miguel Duarte, Tiago Rodrigues, Sancho Moura Oliveira, and Anders Lyhne Christensen. Design and development of an inexpensive aquatic swarm robotics system. In *Proceedings of OCEANS 2016 - Shanghai*, pages 1–7. IEEE Press, Piscataway, NJ, 2016. DOI: 10.1109/OCEANSAP.2016.7485496.

[23] Tiago Rodrigues, Miguel Duarte, Margarida Figueiró, Vasco Costa, Sancho Moura Oliveira, and Anders Lyhne Christensen. Overcoming limited onboard sensing in swarm robotics through local communication. *Transactions on Computational Collective Intelligence*, 9420(XX):201–223, 2015. DOI: 10.1007/978-3-319-27543-7_10.

[24] Miguel Duarte, Vasco Costa, Jorge Gomes, Tiago Rodrigues, Fernando Silva, Sancho Moura Oliveira, and Anders Lyhne Christensen. Evolution of collective behaviors for a real swarm of aquatic surface robots. *PLoS ONE*, 11(3):1–25, 2016. DOI: 10.1371/journal.pone.0151834.

[25] Miguel Duarte, Vasco Costa, Jorge Gomes, Tiago Rodrigues, Fernando Silva, Sancho Moura Oliveira, and Anders Lyhne Christensen. Unleashing the potential of evolutionary swarm robotics in the real world. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion - GECCO '16 Companion*, pages 159–160. ACM Press, 2016. DOI: 10.1145/2908961.2930951.

[26] Miguel Duarte, Jorge Gomes, Vasco Costa, Tiago Rodrigues, Fernando Silva, Victor Lobo, Mário Monteiro Marques, Sancho Moura Oliveira, and Anders Lyhne Christensen. Application of swarm robotic systems to marine environmental monitoring. In *Proceedings of OCEANS 2016 - Shanghai*, pages 1–8. IEEE Press, Piscataway, NJ, 4 2016. DOI: 10.1109/OCEANSAP.2016.7485429.

[27] Miguel Duarte, Jorge Gomes, Vasco Costa, Sancho Moura Oliveira, and Anders Lyhne Christensen. Hybrid control for a real swarm robotic system in an intruder detection task. In *Proceedings of the 18th European Conference on the Applications of Evolutionary Computation (EvoStar)*, pages 213–230. Springer, Berlin, Germany, Berlin, Germany, 2016. DOI: 10.1007/978-3-319-31153-1_15.

[28] Anders Lyhne Christensen, Sancho Oliveira, Octavian Postolache, Maria João de Oliveira, Susana Sargento, Pedro Santana, Luis Nunes, Fernando Velez, Pedro Sebastião, Vasco Costa, Miguel Duarte, Jorge Gomes, Tiago Rodrigues, and Fernando Silva. Design of communication and control for swarms of aquatic surface drones. In *Proceedings of the International Conference on Agents and Artificial Intelligence - Volume 2: ICAART*, pages 548–555. SCITEPRESS - Science and and Technology Publications, 2015. DOI: 10.5220/0005281705480555.

[29] Anders Lyhne Christensen, Miguel Duarte, Vasco Costa, Tiago Rodrigues, Jorge Gomes, Fernando Silva, and Sancho Oliveira. A sea of robots, 2016. Best Robot Video Award @ AAAI-16 Video Competition (AIVC 2016). Phoenix, Arizona. February 2016.

[30] Fernando Velez, Aleksandra Nadziejko, Anders Lyhne Christensen, Sancho Moura Oliveira, Tiago Rodrigues, Vasco Costa, Miguel Duarte, Fernando Silva, and Jorge Gomes. Experimental characterization of wsns applied to swarms of aquatic surface drones. In *Proceedings of the 10th Conference on Telecommunications (CONFTELE)*, 2015.

[31] Fernando J. Velez, Aleksandra Nadziejko, Anders Lyhne Christensen, Sancho Oliveira, Tiago Rodrigues, Vasco Costa, Miguel Duarte, Fernando Silva, and Jorge Gomes. Wireless sensor and networking technologies for swarms of aquatic surface drones. In *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, pages 1–2. IEEE, 2015. DOI: 10.1109/VTCFall.2015.7391193.

[32] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems.* Oxford University Press, 1999. ISBN 0-19-513159-2.

[33] Nithin Mathews, Anders Lyhne Christensen, Eliseo Ferrante, Rehan O'Grady, and Marco Dorigo. Establishing spatially targeted communication in a heterogeneous robot swarm. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems:*

volume 1 - Volume 1, pages 939–946. International Foundation for Autonomous Agents and Multiagent Systems, 2010. ISBN 978-0-9826-5711-9.

[34] Marco Dorigo, Dario Floreano, Luca Maria Gambardella, Francesco Mondada, Stefano Nolfi, Tarek Baaboura, Mauro Birattari, Michael Bonani, Manuele Brambilla, Arne Brutschy, Daniel Burnier, Alexandre Campo, Anders Lyhne Christensen, Antal Decugniere, Gianni Di Caro, Frederick Ducatelle, Eliseo Ferrante, Alexander Forster, Javier Martinez Gonzales, Jerome Guzzi, Valentin Longchamp, Stephane Magnenat, Nithin Mathews, Marco Montes de Oca, Rehan O'Grady, Carlo Pinciroli, Giovanni Pini, Philippe Retornaz, James Roberts, Valerio Sperati, Timothy Stirling, Alessandro Stranieri, Thomas Stutzle, Vito Trianni, Elio Tuci, Ali Emre Turgut, and Florian Vaussard. Swarmanoid: A novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics & Automation Magazine*, 20(4):60–71, 2013. DOI: 10.1109/MRA.2013.2252996.

[35] Levent Bayındır. A review of swarm robotics tasks. *Neurocomputing*, 172: 292–321, 8 2016. DOI: 10.1016/j.neucom.2015.05.116.

[36] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 1 2013. DOI: 10.1007/s11721-012-0075-2.

[37] Scott Camazine. *Self-organization in biological systems*. Princeton University Press, 2003. ISBN 978-0-6911-1624-2.

[38] Marco Dorigo, Mauro Birattari, and Manuele Brambilla. Swarm robotics. *Scholarpedia*, 9(1):1463, 1 2014. DOI: 10.4249/scholarpedia.1463.

[39] A.L. Christensen, R. O'Grady, and M. Dorigo. From fireflies to fault-tolerant swarms of robots. *IEEE Transactions on Evolutionary Computation*, 13(4):754–766, 8 2009. DOI: 10.1109/TEVC.2009.2017516.

[40] Michael Rubenstein, Christian Ahler, Nick Hoff, Adrian Cabrera, and Radhika Nagpal. Kilobot: A low cost robot with scalable operations designed for collective behaviors. *Robotics and Autonomous Systems*, 62(7): 966–975, 7 2014. DOI: 10.1016/j.robot.2013.08.006.

[41] Jörg Seyfried, Marc Szymanski, Natalie Bender, Ramon Estana, Michael Thiel, and Heinz Wörn. The i-swarm project: Intelligent small world autonomous robots for micro-manipulation. *Swarm Robotics*, 3342 (November):70–83, 2005. DOI: 10.1007/978-3-540-30552-1_7.

[42] Marco Dorigo, Elio Tuci, Roderich Groß, Vito Trianni, Thomas Halva Labella, Shervin Nouyan, Christos Ampatzis, Jean-Louis Deneubourg, Gianluca Baldassarre, Stefano Nolfi, et al. The swarm-bots project. *Swarm Robotics*, 3342(November):31–44, 2005. DOI: 10.1007/978-3-540-30552-1_4.

[43] Paul Scerri, Prasanna Velagapudi, Balajee Kannan, Abhinav Valada, Christopher Tomaszewski, John Dolan, Adrian Scerri, Kumar Shaurya Shankar, Luis Bill, and George Kantor. Real-world testing of a multi-robot team. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 3*, pages 1213–1214. International Foundation for Autonomous Agents and Multiagent Systems, 6 2012. ISBN 978-0-9817381-3-0.

[44] Thomas Schmickl, Ronald Thenius, Christoph Moslinger, Jon Timmis, Andy Tyrrell, Mark Read, James Hilder, Jose Halloy, Alexandre Campo, Cesare Stefanini, et al. Cocoro-the self-aware underwater swarm. In *Fifth IEEE Conference on Self-Adaptive and Self-Organizing Systems Workshops (SASOW)*, pages 120–126, Ann Arbor, MI, 2011. IEEE Press, Piscataway, NJ. DOI: 10.1109/SASOW.2011.11.

[45] Quentin Lindsey, Daniel Mellinger, and Vijay Kumar. Construction with quadrotor teams. *Autonomous Robots*, 33(3):323–336, 6 2012. DOI: 10.1007/s10514-012-9305-0.

[46] Sabine Hauert, Severin Leven, Jean-Christophe Zufferey, and Dario Floreano. Communication-based swarming for flying robots. *Proceedings of the Workshop on Network Science and Systems Issues in Multi-Robot Autonomy, IEEE International Conference on Robotics and Automation*, pages 15–20, 5 2010. DOI: 10.1109/ROBOT.2010.5509421.

[47] Francesco Mondada, Giovanni C. Pettinaro, Andre Guignard, Ivo W. Kwee, Dario Floreano, Jean-Louis Deneubourg, Stefano Nolfi, Luca Maria Gambardella, and Marco Dorigo. Swarm-bot: A new distributed robotic concept. *Autonomous Robots*, 17:193–221, 2004. DOI: 10.1023/B:AURO.0000033972.50769.1c.

[48] Roderich Groß, Rehan O 'grady, Anders Lyhne Christensen, and Marco Dorigo. The swarm-bot experience: Strength and mobility through physical cooperation. In Serge Kernbach, editor, *Handbook of Collective Robotics*, chapter 2, pages 49–80. Pan Stanford Publishing, 2013. ISBN 978-9-814-31642-2.

[49] Roderich Groß, Michael Bonani, Francesco Mondada, and Marco Dorigo. Autonomous self-assembly in a swarm-bot. *Proceedings of the 3rd International Symposium on Autonomous Minirobots for Research and Edutainment (AMiRE 2005)*, 22(6):314–322, 12 2006. DOI: 10.1007/3-540-29344-2_47.

[50] Roderich Gross and Marco Dorigo. Towards group transport by swarms of robots. *International Journal of Bio-Inspired Computation*, 1(1/2):1, 2009. DOI: 10.1504/IJBIC.2009.022770.

[51] F. Mondada, L.M. Gambardella, D. Floreano, S. Nolfi, J. Deneubourg, and M. Dorigo. The cooperation of swarm-bots - physical interactions in collective robotics. *IEEE Robotics & Automation Magazine*, 12(2):21–28, 6 2005. DOI: 10.1109/MRA.2005.1458313.

[52] Rehan O'Grady, Roderich Groß, Anders Lyhne Christensen, and Marco Dorigo. Self-assembly strategies in a group of autonomous mobile robots. *Autonomous Robots*, 28(4):439–455, 5 2010. DOI: 10.1007/s10514-010-9177-0.

[53] Definition of the word formation in english language. URL https://en.oxforddictionaries.com/definition/formation. Visited on 2018-04-17.

[54] Yutaka Hori, Hiroki Miyazako, Soichiro Kumagai, and Shinji Hara. Coordinated spatial pattern formation in biomolecular communication networks. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, 1(2):111–121, 2015. DOI: 10.1109/TMBMC.2015.2500567.

[55] Benoit B. Mandelbrot. *The fractal geometry of nature*, volume 173. W. H. Freeman and Company, 1983. ISBN 978-0-7167-1186-5.

[56] Ka-Kit Tung. *Topics in mathematical modeling*. Princeton University Press, Princeton, NJ, 2007. ISBN 978-1-4008-8405-6.

[57] Jessica W. Chen. Understanding pattern formation during morphogenesis, 2012. URL http://sitn.hms.harvard.edu/flash/2012/morphogenesis/. Visited on 2018-04-17.

[58] Julia K. Parrish, Steven V. Viscido, and Daniel Grunbaum. Self-organized fish schools: An examination of emergent properties. *The biological bulletin*, 202(3):296–305, 2002.

[59] M Anthony Lewis and Kar-Han Tan. High precision formation control of mobile robots using virtual structures. *Auton. Robots*, 4(4):387–403, 1997. DOI: 10.1023/A:1008814708459.

[60] P. B. S. Lissaman and C. A. Shollenberger. Formation flight of birds. *Science*, 168(3934):1003–1005, 5 1970. DOI: 10.1126/science.168.3934.1003.

[61] Robert L Shaw. *Fighter combat: Tactics and maneuvering*. Naval Institute Press, 1985. ISBN 978-0-8702-1059-4.

[62] Lemoine and Jullien. Jeu en déviation et configuration du jeu en football. *Ejrieps, 15, 5*, 20:5–20, 1 2008.

[63] T. Balch and R.C. Arkin. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, 1998. DOI: 10.1109/70.736776.

[64] Yang Quan Chen and Zhongmin Wang. Formation control: a review and a new consideration. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3181–3186. IEEE, 2005. DOI: 10.1109/IROS.2005.1545539.

[65] Jonathan R. T. Lawton, Randal W. Beard, and Brett J. Young. A decentralized approach to formation maneuvers. *IEEE Transactions on Robotics and Automation*, 19(6):933–941, 12 2003. DOI: 10.1109/TRA.2003.819598.

[66] Wang Guanghua, Li Deyi, Gan Wenyan, and Jia Peng. Study on formation control of multi-robot systems. In *2013 Third International Conference on Intelligent System Design and Engineering Applications*, pages 1335–1339. IEEE, 1 2013. DOI: 10.1109/ISDEA.2012.316.

[67] R. Fierro, A.K. Das, V. Kumar, and J.P. Ostrowski. Hybrid control of formations of robots. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, volume 1, pages 157–162. IEEE, 2001. DOI: 10.1109/ROBOT.2001.932546.

[68] Jinyan Shao, Guangming Xie, Junzhi Yu, and Long Wang. Leader-following formation control of multiple mobile robots. In *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.*, pages 808–813. IEEE, 2005. DOI: 10.1109/.2005.1467118.

[69] Hongjun Yu, Peng Shi, and Cheng-Chew Lim. Robot formation control in stealth mode with scalable team size. *International Journal of Control*, 0 (0):1–14, 3 2016. DOI: 10.1080/00207179.2016.1149887.

[70] William M. Spears, Diana F. Spears, Jerry C. Hamann, and Rodney Heil. Distributed, physics-based control of swarms of vehicles. *Autonomous Robots*, 17:137–162, 2004. DOI: 10.1023/B:AURO.0000033970.96785.f2.

[71] Wei Ren and Randal Beard. Decentralized scheme for spacecraft formation flying via the virtual structure approach. *Journal of Guidance, Control, and Dynamics*, 27:73–82, 2004. DOI: 10.2514/1.9287.

[72] Nathan Michael, Michael M. Zavlanos, Vijay Kumar, and George J. Pappas. Distributed multi-robot task assignment and formation control. In *2008 IEEE International Conference on Robotics and Automation*, pages 128–133. IEEE, 5 2008. DOI: 10.1109/ROBOT.2008.4543197.

[73] Reza Olfati-Saber and Richard M. Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. *IFAC Proceedings Volumes*, 35(1):495–500, 1 2002. DOI: 10.3182/20020721-6-ES-1901.00244.

[74] N.E. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No.01CH37228)*, volume 3, pages 2968–2973, Orlando, FL, USA, USA, 2001. IEEE. DOI: 10.1109/CDC.2001.980728.

[75] J.P. Desai, J.P. Ostrowski, and V. Kumar. Modeling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 17(6):905–908, 2001. DOI: 10.1109/70.976023.

[76] F. Michaud, D. Letourneau, M. Guilbert, and J.-M. Valin. Dynamic robot formations using directional visual perception. In *IEEE/RSJ International Conference on Intelligent Robots and System*, volume 3, pages 2740–2745. IEEE, 2002. DOI: 10.1109/IRDS.2002.1041684.

[77] Kwang-Kyo Oh, Myoung-Chul Park, and Hyo-Sung Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 3 2015. DOI: 10.1016/j.automatica.2014.10.022.

[78] A.K. Das, R. Fierro, V. Kumar, J.P. Ostrowski, J. Spletzer, and C.J. Taylor. A vision-based formation control framework. *IEEE Transactions*

on Robotics and Automation, 18(5):813–825, 10 2002. DOI: 10.1109/TRA.2002.803463.

[79] Petter Ögren, Edward Fiorelli, and Naomi Ehrich Leonard. Formations with a mission: Stable coordination of vehicle group maneuvers. In *Proceedings of the 15th International Symposium on Mathematical Theory of Networks and Systems*, 2002.

[80] Laura Barnes, MaryAnne Fields, and Kimon Valavanis. Unmanned ground vehicle swarm formation control using potential fields. In *2007 Mediterranean Conference on Control & Automation*, pages 1–8. IEEE, 6 2007. DOI: 10.1109/MED.2007.4433724.

[81] Michael Defoort, Thierry Floquet, A. Kokosy, and Wilfrid Perruquetti. Sliding-mode formation control for cooperative autonomous mobile robots. *IEEE Transactions on Industrial Electronics*, 55(11):3944–3953, 11 2008. DOI: 10.1109/TIE.2008.2002717.

[82] Alex S. Fraser. Simulation of genetic systems by automatic digital computers vi. epistasis. *Australian Journal of Biological Sciences*, 13(2): 150–162, 1960. DOI: 10.1071/BI9600150.

[83] Stefano Nolfi. Evolutionary robotics: Exploiting the full power of self-organization. *Connection Science*, 10(3-4):167–184, 1998. DOI: 10.1080/095400998116396.

[84] Vito Trianni and Stefano Nolfi. Engineering the evolution of self-organizing behaviors in swarm robotics: a case study. *Artificial life*, 17(3):183–202, 1 2011. DOI: 10.1162/artl_a_00031.

[85] Dario Floreano and Laurent Keller. Evolution of adaptive behaviour in robots by means of darwinian selection. *PLoS biology*, 8(1), 1 2010. DOI: 10.1371/journal.pbio.1000292.

[86] Gianluca Baldassarre, Vito Trianni, Michael Bonani, Francesco Mondada, Marco Dorigo, and Stefano Nolfi. Self-organized coordinated motion in groups of physically connected robots. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 37(1):224–239, 2 2007. DOI: 10.1109/TSMCB.2006.881299.

[87] Miguel Duarte, Sancho Moura Oliveira, and Anders Lyhne Christensen. Evolution of hybrid robotic controllers for complex tasks. *Journal of Intelligent & Robotic Systems*, 78(3-4):463–484, 6 2015. DOI: 10.1007/s10846-014-0086-x.

[88] Valerio Sperati, Vito Trianni, and Stefano Nolfi. Evolving coordinated group behaviours through maximisation of mean mutual information. *Swarm Intelligence*, 2(2-4):73–95, 9 2008. DOI: 10.1007/s11721-008-0017-1.

[89] Miguel Duarte, Jorge Gomes, Vasco Costa, Sancho Moura Oliveira, and Anders Lyhne Christensen. Hybrid control for a real swarm robotics system in an intruder detection task. *Proceedings of the 18th European Conference on the Applications of Evolutionary Computation (EvoStar)*, pages 213–230, 2016. DOI: 10.1007/978-3-319-31153-1_15.

[90] Ashish Kumar, Sanjeev Sharma, Ritu Tiwari, and Samriddhi Majumdar. Area exploration by flocking of multi robot. *Procedia Engineering*, 41: 377–382, 2012. DOI: 10.1016/j.proeng.2012.07.187.

[91] Amanda J.C. Sharkey and Noel Sharkey. The application of swarm intelligence to collective robots. In *Advances in Applied Artificial Intelligence*, pages 157–185. IGI Global, 2006. DOI: 10.4018/978-1-59140-827-7.ch006.

[92] Vito Trianni, Stefano Nolfi, and Marco Dorigo. Cooperative hole avoidance in a swarm-bot. *Robotics and Autonomous Systems*, 54(2):97–103, 2 2006. DOI: 10.1016/j.robot.2005.09.018.

[93] Valerio Sperati, Vito Trianni, and Stefano Nolfi. Self-organised path formation in a swarm of robots. *Swarm Intelligence*, 5(2):97–119, 4 2011. DOI: 10.1007/s11721-011-0055-y.

[94] E. Şahin, T.H. Labella, V. Trianni, J.-L. Deneubourg, P. Rasse, D. Floreano, L. Gambardella, F. Mondada, S. Nolfi, and M. Dorigo. Swarm-bot: pattern formation in a swarm of self-assembling mobile robots. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 4, page 6. IEEE, 2002. DOI: 10.1109/ICSMC.2002.1173259.

[95] E. Bahceci and E. Şahin. Evolving aggregation behaviors for swarm robotic systems: a systematic case study. In *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, pages 333–340. IEEE, 2005. DOI: 10.1109/SIS.2005.1501640.

[96] Vito Trianni, Roderich Groß, Thomas H. Labella, Erol Şahin, and Marco Dorigo. Evolving aggregation behaviors in a swarm of robots. In Wolfgang Banzhaf, Jens Ziegler, Thomas Christaller, Peter Dittrich, and Jan T. Kim, editors, *Advances in Artificial Life*, volume 2801 of *Lecture Notes in*

Computer Science, pages 865–874. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. DOI: 10.1007/b12035.

[97] Onur Soysal, Erkín Bahçecí, and Erol Şahín. Aggregation in swarm robotic systems: Evolution and probabilistic control. *Turkish Journal of Electrical Engineering & Computer Sciences*, 15(2):199–225, 8 2007.

[98] Jorge Gomes, Paulo Urbano, and Anders Lyhne Christensen. Evolution of swarm robotics systems with novelty search. *Swarm Intelligence*, 7(2-3): 115–144, 9 2013. DOI: 10.1007/s11721-013-0081-z.

[99] Dario Floreano, Sara Mitri, Stéphane Magnenat, and Laurent Keller. Evolutionary conditions for the emergence of communication in robots. *Current Biology*, 17(6):514–519, 2007. DOI: 10.1016/j.cub.2007.01.058.

[100] Francesco Mondada, Michael Bonani, Xavier Raemy, James Pugh, Christopher Cianci, Adam Klaptocz, Stephane Magnenat, Jean-Christophe Zufferey, Dario Floreano, and Alcherio Martinoli. The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, volume 1, pages 59–65, Castelo Branco, 2009. IPCB: Instituto Politécnico de Castelo Branco.

[101] F. Mondada, E. Franzi, and A. Guignard. The development of khepera. In *Experiments with the Mini-Robot Khepera, Proceedings of the First International Khepera Workshop*, pages 7–14, 11 1999.

[102] Fanny Riedo, Philippe Rétornaz, Luc Bergeron, Nathalie Nyffeler, and Francesco Mondada. A two years informal learning experience using the thymio robot. In *Advances in Autonomous Mini Robots*, pages 37–48. Springer Berlin Heidelberg, 2012. DOI: 10.1007/978-3-642-27482-4_7.

[103] Martin T. Hagan, Howard B. Demuth, Mark H. Beale, and Orlando De Jesús. *Neural Network Design*, volume 20. Martin Hagan, 2014, 2 edition, 2014. ISBN 978-0-9717-3211-7.

[104] Imad Basheer and M Hajmeer. Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods*, 43 (1):3–31, 12 2000. DOI: 10.1016/S0167-7012(00)00201-3.

[105] John Adrian Bondy. *Graph Theory With Applications*. Citeseer, 6 1976. ISBN 0-444-19451-7.

[106] W. Thomas Miller, Paul J. Werbos, and Richard S. Sutton. *Neural Networks for Control*. MIT press, 1995. ISBN 0-262-63161-X.

[107] Christopher M. Bishop. *Neural Networks for Pattern Recognition.* Oxford university press, 1995. ISBN 0-198-53864-2.

[108] Robert Hecht-Nielsen. Applications of counterpropagation networks. *Neural Networks*, 1(2):131–139, 1 1988. DOI: 10.1016/0893-6080(88)90015-9.

[109] Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems 7*, pages 231–238. Morgan Kaufmann Publishers, 7 edition, 1995. ISBN 0-262-20104-6.

[110] J Khan, J S Wei, M Ringnér, L H Saal, M Ladanyi, F Westermann, F Berthold, M Schwab, C R Antonescu, C Peterson, and P S Meltzer. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature medicine*, 7(6):673–679, 6 2001. DOI: 10.1038/89044.

[111] T. Fukuda and T. Shibata. Theory and applications of neural networks for industrial control systems. *IEEE transactions on industrial electronics*, 39 (6):472–489, 1992. DOI: 10.1109/41.170966.

[112] Robert Hecht-Nielsen. Neurocomputer applications. In *Neural Computers*, pages 445–453. Springer Berlin Heidelberg, Berlin, Heidelberg, 1989. DOI: 10.1007/978-3-642-83740-1_45.

[113] Dean A. Pomerleau. *Neural Network Perception for Mobile Robot Guidance*, volume 239. Springer Science & Business Media, 2012. DOI: 10.1007/978-1-4615-3192-0.

[114] Wei Po Lee. Evolving complex robot behaviors. 121(1):1–25, 12 1999. DOI: 10.1016/S0020-0255(99)00078-X.

[115] Vito Trianni. *Evolutionary swarm robotics : evolving self-organising behaviours in groups of autonomous robots.* Springer, 2008. ISBN 978-3-5407761-2-3.

[116] N. Jakobi. Evolutionary robotics and the radical envelope-of-noise hypothesis. *Adaptive Behavior*, 6(2):325–368, 9 1997. DOI: 10.1177/105971239700600205.

[117] Sylvain Koos, Jean-Baptiste Mouret, and Stéphane Doncieux. Crossing the reality gap in evolutionary robotics by promoting transferable controllers. In *Proceedings of the 12th annual conference on Genetic and evolutionary*

computation - GECCO '10, page 119, New York, New York, USA, 2010. ACM Press. DOI: 10.1145/1830483.1830505.

[118] Sylvain Koos, J-B Mouret, and S. Doncieux. The transferability approach: Crossing the reality gap in evolutionary robotics. *IEEE Transactions on Evolutionary Computation*, 17(1):122–145, 2 2013. DOI: 10.1109/TEVC.2012.2185849.

[119] Justin E. Manley. Unmanned surface vehicles, 15 years of development. In *OCEANS 2008*, pages 1–4, Quebec City, QC, 2008. IEEE Press, Piscataway, NJ. DOI: 10.1109/OCEANS.2008.5152052.

[120] Zhixiang Liu, Youmin Zhang, Xiang Yu, and Chi Yuan. Unmanned surface vehicles: An overview of developments and challenges. *Annual Reviews in Control*, 41:71–93, 2016. DOI: 10.1016/j.arcontrol.2016.04.018.

[121] Eduardo Pinto, Francisco Marques, Ricardo Mendonca, Andre Lourenco, Pedro Santana, and Jose Barata. An autonomous surface-aerial marsupial robotic team for riverine environmental monitoring: Benefiting from coordinated aerial, underwater, and surface level perception. In *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, pages 443–450. IEEE, 2014. DOI: 10.1109/ROBIO.2014.7090371.

[122] Alfredo Martins, Hugo Ferreira, Carlos Almeida, Hugo Silva, José Miguel Almeida, and Eduardo Silva. Roaz and roaz ii autonomous surface vehicle design and implementation. In *International Lifesaving Congress 2007*, La Coruna, Spain, 2007.

[123] Dario Albani, Joris IJsselmuiden, Ramon Haken, and Vito Trianni. Monitoring and mapping with robot swarms for agricultural applications. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 8 2017. DOI: 10.1109/AVSS.2017.8078478.

[124] Brandon M. Zoss, David Mateo, Yoke Kong Kuan, Grgur Tokić, Mohammadreza Chamanbaz, Louis Goh, Francesco Vallegra, Roland Bouffanais, and Dick K. P. Yue. Distributed system of autonomous buoys for scalable deployment and monitoring of large waterbodies. *Autonomous Robots*, pages 1–21, 2 2018. DOI: 10.1007/s10514-018-9702-0.

[125] Geert De Cubber, Daniela Doroftei, Daniel Serrano, Keshav Chintamani, Rui Sabino, and Stephane Ourevitch. The eu-icarus project: Developing assistive robotic tools for search and rescue operations. In *2013 IEEE*

International Symposium on Safety, Security, and Rescue Robotics (SSRR), pages 1–4, Linkoping, 10 2013. IEEE Press, Piscataway, NJ. DOI: 10.1109/SSRR.2013.6719323.

[126] Bruno M. Ferreira, Anibal C. Matos, and Jose C. Alves. Water-jet propelled autonomous surface vehicle ucap: System description and control. In *Proceedings of OCEANS 2016 - Shanghai*, pages 1–5, Shanghai, 4 2016. IEEE. DOI: 10.1109/OCEANSAP.2016.7485364.

[127] Jared M. Moore, Anthony J. Clark, and Philip K. McKinley. Evolution of station keeping as a response to flows in an aquatic robot. In *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference - GECCO '13*, page 239, New York, New York, USA, 7 2013. ACM Press. DOI: 10.1145/2463372.2463402.

[128] Tomasz Praczyk. Using augmenting modular neural networks to evolve neuro-controllers for a team of underwater vehicles. *Soft Computing*, 18 (12):2445–2460, 2 2014. DOI: 10.1007/s00500-014-1221-0.

[129] Geoff N. Roberts and Robert Sutton. *Advances in Unmanned Marine Vehicles*, volume 69. IET, The Institution of Engineering and Technology, 1 2006. DOI: 10.1049/PBCE069E.

[130] J. Curcio, J. Leonard, and A. Patrikalakis. Scout - a low cost autonomous surface platform for research in cooperative autonomy. In *Proceedings of OCEANS 2005 MTS/IEEE*, pages 1–5. IEEE, 2005. DOI: 10.1109/OCEANS.2005.1639838.

[131] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. *ICRA workshop on open source software*, 3(3.2):5, 2009.

[132] Eduardo Pinto, Pedro Deusdado, Francisco Marques, Andre Lourenco, Ricardo Mendonca, Pedro Santana, Luis Flores, and Jose Barata. A health and usage monitoring system for ros-based service robots. In *2015 10th International Symposium on Mechatronics and its Applications (ISMA)*, pages 1–6. IEEE, 12 2015. DOI: 10.1109/ISMA.2015.7373493.

[133] Paul Michael Newman. Moos - mission orientated operating suite. *Massachusetts Institute of Technology, Technical Report*, 2006.

[134] Michael R. Benjamin, John J. Leonard, Henrik Schmidt, and Paul M. Newman. An overview of moos-ivp and a brief users guide to the ivp helm

autonomy software. Technical report, Massachusetts Institute of Technology, Cambridge, MA, 2009.

[135] MIT LAMSS. March 2016 moos-ivp survey, 2016. URL `http://oceanai.mit.edu/moos-ivp/docs/survey_results.pdf`. Visited on 2016-11-02.

[136] Hugo Miguel Ferreira, Carlos Almeida, Alfredo Martins, José Miguel Almeida, André Dias, Guilherme Silva, and Eduardo Silva. Environmental modeling with precision navigation using roaz autonomous surface vehicle. In *IROS 2012-IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1–6, 2012. ISBN 978-1-4673-1737-5.

[137] Miguel Duarte, Fernando Silva, Tiago Rodrigues, Sancho Moura Oliveira, and Anders Lyhne Christensen. Jbotevolver: A versatile simulation platform for evolutionary robotics. In *Artificial Life 14: Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems*, pages 210–211. The MIT Press, 7 2014. DOI: `10.7551/978-0-262-32621-6-ch035`.

[138] Hélio Silva, Sancho Moura Oliveira, and Anders Lyhne Christensen. Conillon: A lightweight distributed computing platform for desktop grids. In *Information Systems and Technologies (CISTI), 2011 6th Iberian Conference on*, pages 1–6. IEEE, 2011. ISBN 978-1-4577-1487-0.

[139] Navy Warfare Development Command. Extac 1000 - maritime maneuvering and tactical. 8 1996. URL `http://nato.radioscanner.ru/files/article66/1000.pdf`. Visited on 2018-04-17.

[140] Kenneth O. Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 6 2002. DOI: `10.1162/106365602320169811`.

[141] Kenneth Owen Stanley and Risto P. Miikkulainen. *Efficient evolution of neural networks through complexification.* PhD thesis, The University of Texas at Austin, 2004.

[142] GlobalTop Technology Inc. FGPMMOPA6H Datasheet, 2012. URL `https://www.adafruit.com/datasheets/GlobalTop-FGPMMOPA6H-Datasheet-V0A.pdf`. Visited on 2018-04-17. Original document from GlobalTop Technology Inc.

[143] STMicroelectronics. LSM303D datasheet, 2013. URL `http://www.st.com/web/en/resource/technical/document/datasheet/DM00057547.pdf`. Visited on 2018-04-17.

[144] Maxim Integrated Products, Inc. DS18B20 datasheet, 2015. URL `https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf`. Visited on 2018-04-17.

[145] Jon Hamkins. Pulse position modulation. In *Handbook of Computer Networks: Key Concepts, Data Transmission, and Digital and Optical Networks, Volume 1*, pages 492–508. John Wiley & Sons, Inc, Hoboken, NJ, USA, 2007. DOI: `10.1002/9781118256053`.

[146] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 3 1960. DOI: `10.1115/1.3662552`.