UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingenieros de Telecomunicación

Emergence of transferable communication in swarm robotics: from signaling towards language

D O C T O R A L T H E S I S

Submitted for the degree of Doctor by:

Rafael Sendra Arranz
Máster Universitario en Ingeniería de Telecomunicación

Madrid, 2025


UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingenieros de Telecomunicación

Doctoral Degree in Communication Technologies and Systems

Emergence of transferable communication in swarm robotics: from signaling towards language

D O C T O R A L T H E S I S

Submitted for the degree of Doctor by:

Rafael Sendra Arranz
Máster Universitario en Ingeniería de Telecomunicación

Under the supervision of:
Dr. Álvaro Gutiérrez Martín

Madrid, 2025

Title: Emergence of transferable communication in swarm robotics: from signaling towards language

Author: Rafael Sendra Arranz

Doctoral Programme: Communication Technologies and Systems

Thesis Supervision:

Dr. Álvaro Gutiérrez Martín, Full Professor, Universidad Politécnica de Madrid (Supervisor)

External Reviewers:

Thesis Defense Committee:

Thesis Defense Date:

This thesis has been partially supported by the predoctoral grant from the "Programa Propio I+D+i" of the Universidad Politécnica de Madrid and by Projects NAUTILUS and NEMO under grants PID2020-112502RB-C41 and PID2023-146540OB-C42, respectively.

To my Family

Acknowledgement

I am deeply grateful to my supervisor Prof. Álvaro Gutiérrez Martín, for his constant support, guidance, and patience throughout this long journey. It has been a long time since I first walked into his office, hoping to do my bachelor's thesis under his supervision, motivated by his inspiring robotics classes. I am very grateful that he trusted me from the very beginning of this journey. He introduced me to the amazing fields of swarm robotics and artificial intelligence, which are now the basis of my research. His insights, encouragement, and interesting discussions have been crucial in shaping this Thesis and me as a researcher, and he has been a fundamental pillar in keeping me motivated throughout this challenging process.

I would also like to thank ROBOLABO, the lab I have been a part of all these years. It has been a great environment for carrying out this Thesis. I thank all the friends and colleges in ROBOLABO, especially, Blanca, Ainhoa, Milad, César, Alejandro, Javi, Rodrigo, Henar, Jaime, and many others.

I thank support and sponsorship from the predoctoral grant from the "Programa Propio I+D+i" of the Universidad Politécnica de Madrid. I also gratefully acknowledge the Universidad Politécnica de Madrid for providing computing resources on Magerit Supercomputer. I am grateful to Project NAUTILUS (PID2020-112502RB-C41) and Project NEMO (PID2023-146540OB-C42).

I would also like to thank the Maersk Mc-Kinney Moller Institute of the University of Southern Denmark (SDU) for receiving me during my research stay. I am especially grateful to Prof. Anders Lyhne Christensen for his hospitality, guidance and support. I also want to thank all the friends that I met during my time at SDU: Kasper, Junaid, Cao, and Inés, among many others.

More personally, I would like to thank all my family and friends for all their support and help. Especially, to my father, my mother, my siblings, Javi and Bea, and my dog, Heiko. Without them, finishing this PhD Thesis would have been imposible. They have been always there when I needed them the most.

Thank you!

Abstract

Communication is a fundamental aspect of any Swarm Robotics (SR) system, enabling robots to interact and coordinate to solve complex tasks collectively. Traditionally, communication mechanisms in SR have been designed either manually or through evolutionary algorithms. However, research on the evolution of communication has reached a stagnation point due to two key challenges. First, there is a lack of standardized frameworks that facilitate the usage of the same emerged communication across multiple tasks, a property defined in this Thesis as *transferability*. This property is absent in most SR studies devoted to the evolution of communication, where solutions are generally evolved ad-hoc for specific tasks. Second, there have only been few reports on the successful evolution of communication beyond simple signaling in the SR literature. Moreover, the emerged signaling communications are ultimately grounded on behaviors specifically evolved or designed for given task. Thus, there is a significant barrier that hinders the emergence of more complex forms of communication (i.e. language based communications).

This PhD Thesis addresses both of these challenges: the lack of transferability and the barrier to language-based communication. Firstly, transferability is addressed using a minimal communication system that robots can use to solve *primitive* tasks. This system was used to solve three different primitives (election of a leader, identification of the swarm border, and alignment of the robots), obtaining the emergence of effective communications, and exhibiting good scalability and robustness. Moreover, it only needed minor modifications in the design of the communication and robot controller. However, the communication semantics still required a full re-evolution of the system for each task in order to emerge. Surprisingly, even though signaling communications were evolved for all the primitives, the emerged semantics notably diverged depending on the aim of the task. In addition, the transferability of communication in SR systems was further explored during this PhD Thesis. A novel communication module that achieves self-organized and distributed communication, coordination, and robot planning is proposed. Furthermore, this module was successfully applied to three popular SR tasks (aggregation, formation control, and foraging) using the exact same evolved communication and requiring minor and task-specific design updates. The proposed communication module is based on *virtual state spaces* that are partitioned into regions. These regions are linked to meanings grounded on physical behaviors of the robots. Robots communicate their coordinates (states) in this virtual space to other neighbors in range, and they *virtually* navigate across the regions in order to plan and coordinate their physical behavior.

Lastly, this PhD Thesis also explores new solutions to cross the barrier that hinders the emergence of language-based communications. It sheds some light on this open problem by tackling the emergence and self-acquistion of a common lexicon that is shared across all the robots in the swarm. Robots started with no prior knowledge about the environment, explored the environment, and, ultimately, converged to a unique lexicon that is perceptually grounded on geometric objects with diverse properties. For this aim, language games, cultural transmission, and semiotics techniques were used. Furthermore, the emerged lexicons exhibited several complex properties that are characteristic of human language, such as compositionality, displaced communication, duality of patterning, and arbitrariness.

Resumen

La comunicación es un aspecto fundamental de cualquier sistema de robótica de enjambre, permitiendo la interacción y coordinación entre robots con el fin de resolver tareas complejas de forma colectiva. Tradicionalmente, la comunicación en robótica de enjambre ha sido diseñada manualmente o mediante algoritmos evolutivos. No obstante, la investigación sobre la evolución de la comunicación ha alcanzado un punto de estancamiento, debido a dos desafíos fundamentales. En primer lugar, existe una falta de marcos estandarizados que permitan reutilizar la misma comunicación emergente en múltiples tareas, una propiedad definida en esta tesis como *transferibilidad*. Esta propiedad está ausente en la mayoría de sistemas de robótica de enjambre dedicados a la evolución de la comunicación, con soluciones generalmente evolucionadas ad-hoc para tareas concretas. En segundo lugar, existen muy pocos estudios sobre robótica de enjambre que reporten la evolucion de sistemas de comunicación más allá de señalizaciones. Además, la señalización emergente está generalmente anclada a comportamientos robóticos especificamente evolucionados o diseñados para la tarea concreta a resolver. Por lo tanto, existe una barrera considerable que dificulta la emergencia de formas de comunicación complejas.

Esta Tesis aborda estos dos desafios: la falta de transferabilidad y la barrera que impide alcanzar comunicaciones basadas en lenguaje. Primeramente, la adquisición de la transferabilidad ha sido abordada usando un sistema de comunicación minimalista que los robots pueden emplear para resolver primitivas. Este sistema ha sido aplicado en tres tareas diferentes (elección de un líder, identificación del borde del enjambre y alineación de los robots), obteniendo la emergencia de comunicaciones efectivas y mostrando soluciones estables y robustas. Además, el sistema solo requiere modificaciones menores en el diseño de la comunicación y controlador robótico. No obstante, la emergencia de la semántica de la comunicación necesita una re-evolución completa para cada tarea. Sorprendentemente, aunque se evolucionaron señalizaciones en todas las primitivas, la semántica emergente divergió notablemente en función del objetivo de la tarea. Adicionalmente, en esta Tesis se profundizó en la transferibilidad de la comunicación en robótica de enjambre. Se ha propuesto un nuevo módulo de comunicación que permite una coordinación y planificación robótica auto-organizadas y distribuidas. Asimismo, el módulo de comunicación se aplicó correctamente en tres tareas de robótica colectiva populares (agregación, formación, y recolección) usando exactamente la misma comunicación previamente evolucionada. El módulo de comunicación está basado en *espacios de estados virtuales*, divididos en regiones. Estas regiones están ancladas en comportamientos físicos de los robots. Los robots comunican sus coordenadas (estados) en el espacio virtual a otros vecinos, y navegan *virtualmente* a través de las regiones con el fin de coordinar y planificar su comportamiento.

Finalmente, esta Tesis también explora nuevas soluciones que permitan cruzar la barrera hacia la emergencia de lenguaje. Arroja algo de luz sobre este problema abierto mediante la emergencia y auto-adquisición de un léxico común y compartido por todos los robots en el enjambre. Los robots comenzaron sin conocimiento previo del entorno y, trás un proceso de exploración, convergieron a un léxico único y perceptualmente basado en objetos geométricos con propiedades diversas. Para ello, se emplearon juegos de lenguaje, transmisión cultural

y técnicas semióticas. Además, los léxicos emergentes mostraron propiedades complejas características del lenguaje humano, como la composicionalidad, la comunicación desplazada, la dualidad de patrones y la arbitrariedad.

Contents

List of Figures

List of Algorithms

Abbreviations

AES Active Elastic Sheet

AI Artificial Intelligence

AP Artificial Physics

APF Artificial Potential Functions

ARK Augmented Reality for Kilobots

CMA-ES Covariance Matrix Adaptation Evolution Strategy

CoM Center of Mass

CPFA Central-Place Foraging Algorithm

CSS Compositionality Sharing Score

DDPG Deep Deterministic Policy Gradient

DDSA Distributed Deterministic Spiral Search

EA Evolutionary Algorithm

ER Evolutionary Robotics

ES Evolutionary Strategy

ILM Iterated Learning Model

LAD Language Acquisition Device

LSS Lexicon Sharing Score

MAN Meaning Associative Network

MARL Multi-Agent Reinforcement Learning

MAS Multi-Agent System

MLP Multi-Layer Perceptron

MRS Multi-Robot System

MNG Minimal Naming Game

MPFA Multiple-Place Foraging Algorithm

PFSM Probabilistic Finite State Machine

PSO Particle Swarm Optimization

RF RadioFrequency

RL Reinforcement Learning

SGP Symbol Grounding Problem

SOM Self-Organizing Map

SR Swarm Robotics

SI Swarm Intelligence

TNR True Negative Rate

TPR True Positive Rate

VLSR Very Large Scale Robotics

VSDM Virtual Spring-Damper Mesh

WOSP Wave Oriented Swarm Paradigm

Chapter 1

Introduction

1.1 Motivation and problem statement

The cooperation and collaboration of multiple robots to achieve a common goal is of essence in many real-world applications. With this goal in mind, Swarm Robotics (SR) [3, 4] has emerged as a field of research that treats societies of robots as a single entity (the *swarm*). In SR, the robots that compose a swarm are simple and homogeneous, yet self-sufficient agents that interact among them through local rules, with the ultimate goal of solving a common task. SR takes direct inspiration from diverse collective behaviors in nature, such as flocks of birds, fish schools, or ant foraging, among many others. Such biological inspiration is clearly displayed in the global behaviors, the local rules, and overall computation paradigm of SR. Moreover, some of the most popular SR tasks, e.g. flocking, foraging, and formation control, resemble actual behaviors of societies of animals in nature. As an example, Figure 1.1

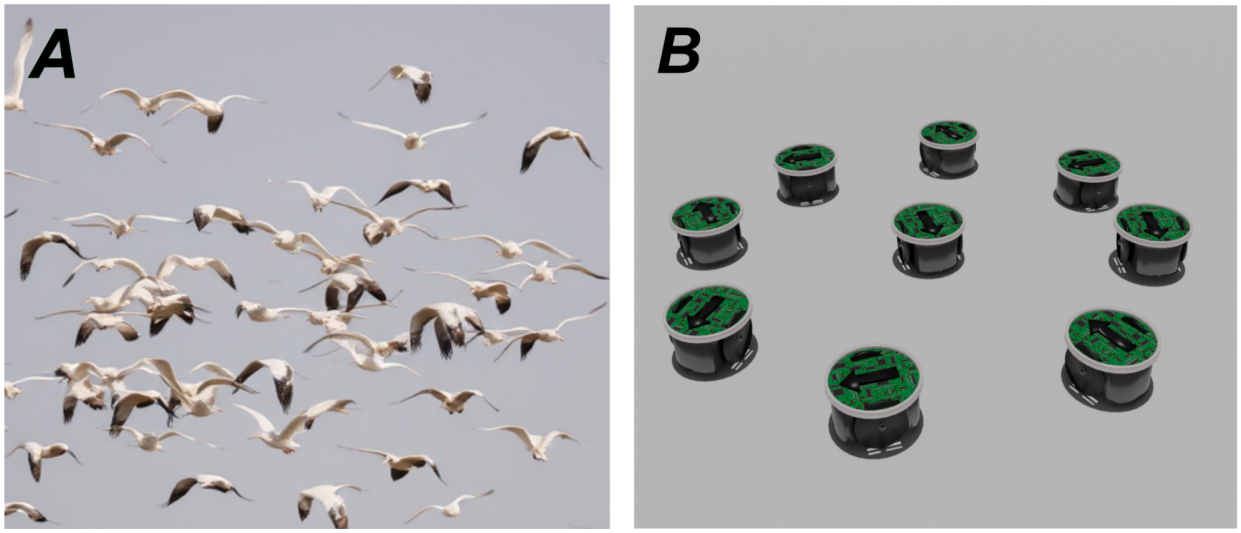


Figure 1.1: Example of a flock of birds in nature (**A**), and a swarm of mobile robots (**B**). Image (**A**) was generated using AI from https://deepai.org/machine-learning-model/text2img

illustrates a flock of birds (biological swarm), and a formation of mobile robots (robot swarm).

Communication is one of the pillars in SR systems, that must be carefully designed in order to achieve effective collective behaviors. However, SR communication is a broad term that encompasses a wide range of technologies, mechanics, semantics, and communication channels. An act of communication between two individuals can be as simple and indirect as plainly being aware of each other's presence. Additionally, communication can be achieved through the direct exchange of messages, or by means of the deposit of virtual pheromone trails within the environment (*stigmergy*), see e.g. [5, 6]. Moreover, communication in SR can be either manually designed by the researcher or, alternatively, it can emerge as a result of an optimization algorithm. In the former case, the researcher entirely defines the semantics and rules of the communication, usually with the aim of solving some specific task. In contrast, the latter scenario considers automatically designed communications, so that it is said that the semantics and rules of the communication emerge. The optimization processes that are generally used in SR to achieve the emergence of communication are evolutionary algorithms.

However, the emergence and evolution of communication presents several important problems that have not been sufficiently addressed in the SR research field:

- *Lack of generalization and standardization*: the evolution of communication has been used in multiple SR tasks, employing a wide range of robot controllers, sensors, actuators, and communication technologies (see e.g., [7, 8, 9, 10]). However, the emerged communication semantics, rules and mechanics are evolved ad-hoc for specific tasks or use cases. This task specialization is useful from a theoretical perspective to assess the limits of evolutionary computation, and study the origins and conditions for the emergence of communication. Nonetheless, this lack of generalization is a problem from an engineering perspective, because robot swarms with evolved communication systems cannot be straightforwardly used in different tasks and contexts apart from the one in which they were originally conceived. Cambier et al. discussed this issue in [11], stating that direct communication semantics and rules are usually designed and constructed for solving specific SR problems. The task generalization is also refered in this PhD Thesis as *transferability*, as the capability of an evolved robot swarm to be used in multiple tasks, with minor modifications. The following levels of transferability are considered:

  (1) *No transferability*: the robot swarm has been designed ad-hoc for a specific task and it cannot be used in a different one without its complete re-desing and re-evolution.

  (2) *Weak transferability*: the system can be used in multiple tasks with minor or no design mofication. Specifically, the same unoptimized communication system can be used across many tasks, without requiring its re-design. Nonetheless, an independent re-evolution of the system is required for every new task being solved.

  (3) *Strong transferability*: the system can be used in multiple independent tasks without the need to re-design and re-evolve the system. It may require minor and task-specific modifications on the robot controller, sensors and actuators. In contrast to *weak transferability*, the emerged communication semantics and rules resulting from previous evolutionary runs can be indistinctly re-used without the need to re-optimize the system.

Most of the SR studies dealing with the evolution of communication lie under the first category, i.e. no transferability. Thus, there is a lack of a design and evolution paradigm that allows transferable (both weak and strong) commmunications in robot swarms. In line with the *lack of generalization*, the evolution of communication in the SR field also suffers from a lack of standardized frameworks that hinder its replicability and use in real-world scenarios [12]. In fact, it is a necessary condition to enable *weak transferability* in robot swarms, as unified and standardized design and evolution frameworks are of essence to allow the use of the same SR design principles across a wide range of tasks.

- *Signaling and the barrier to language-like communications*: most of the SR studies targeting the evolution of communication result in simple signaling communications, which essentially consist of a series of signals that are transmitted under certain conditions and that trigger specific behavioral responses upon receival [11]. Therefore, it can be considered that the SR research has reached an stagnant point in the evolution of communication, that must be bypassed in order to, ultimately, achieve the emergence of complex language-based communications. With the necessary distinctions, it is analogous to the considerable barrier between signaling-based animal communications and human language. To overcome this stagnation, the theoretical bases of language and its evolution should be deeply analysed. The evolved communication must be based on symbols, and their manipulation through grammar rules. However, such symbols must overpass the Symbol Grounding Problem (SGP) [13] and be physically grounded on external entities in the environment [14]. Furthermore, other optimization techniques apart from evolutionary algorithms, such as cultural transmission and language games [11], should be also considered. In order to approach to a language, the emerged communication should desirably satisfy some of its most distinctive features: symbolic communication and compositionality, duality of patterning, arbitrariness, cultural transmission, and displaced communication (see [15, 16] and Chapter 3 for more details).

This PhD Thesis is related to the emergence of communication using artificial evolution, either mimicking biological evolution or cultural transmission. The main motivation behind this Thesis is to address these SR concerns, allowing future SR studies to reach the emergence and evolution of more complex and transferable communications. Instead of tackling all these issues simulaneously, a *divide-and-conquer* strategy is followed along this document: firstly, the emergence of basic signaling communications in *weakly transferable* robot swarms is considered. Secondly, a unified communication framework that allows *strong transferability* is proposed and assessed in three completely different SR tasks. Artificial Neural Networks (ANNs), specificially Continuous-Time Recurrent Neural Networks (CTRNNs), and evolutionary algorithms (EAs) are used to achieve the proposed goals. In third place, the emergence of communication in robot swarms that narrow gap between signaling and language-like communications is proposed. In this last part of the PhD Thesis, the application of language games and semiotics is considered, which, as far as I am concerned, have not been sufficiently explored in SR systems.

1.2 Thesis aim

The aim of this PhD Thesis is twofold, resulting from the problems stated in Section 1.1.

On the one hand, it is to obtain the emergence of communication in SR systems that can be straightforwardly applied to multiple tasks with minor changes. This objective is also divided into two subgoals, addressing *weak* and *strong* transferability, respectively. Firstly, to obtain *weak* transferability, a minimal and highly constrained communication system is proposed and assessed in three different *primitive tasks*, namely, the election of a leader, the identification of the frontier of the swarm, and the heading orientation consensus. The resulting communication system exhibits *weak* transferability properties provided that it can be used for all the three SR tasks with very minor design changes. However, the robot controller still requires to be re-evolved separately for each of the tasks. To fulfill this goal, CTRNNs are the chosen as the main candidate for the robot controllers, that are, in turn, evolved using a Genetic Algorithm (GA). Secondly, to obtain *strong transferability*, a novel communication framework based on *virtual state spaces* is proposed. Once evolved, the communication can be transferred to multiple independent tasks without any modifications to the communication and with minor design changes in the robot controller. This *strong transferability* should be assessed in three different SR tasks, being aggregation in groups, formation control, and foraging the main candidates.

On the other hand, the emergence of a compositional lexicon, as a first step towards language-like communications is proposed. To fulfill this aim, cultural evolution, language games, and hebbian learning are considered. The experiment is based on a simulation in which a robot swarm navigates and explores an arena with multiple geometric objects of diverse colors and shapes. The agents must self-organize an initially empty lexicon to refer to the diverse objects and referents in the environment. To reach this aim, the acquired symbols of the communication must overcome the SGP, being physically grounded based on the external world based each robot's experience and sensory perceptions. Moreover, the emerged communication should desirably acquire, at some extent, the most distinguishable properties of language. These properties are: symbolic communication and compositionality, duality of patterning, arbitrariness, cultural transmission, and displaced communication.

The following features are considered in the experiments and simulations of this PhD Thesis:

- *Decentralization*: the robot control and communication are fully decentralized, implying that there is not a central server or robot that controls and commands the swarm.
- *Scalability*: the robot swarm and the evolved communication must considerably preserve its performance as the number of robots (*swarm size*) increase.
- *Robustness*: the robot swarm and the communication system are fault tolerant, so that the loss or failure of a robot does not have an impact on the overall swarm performance.
- *Transferability*: the robot swarm and the communication system must be transferable to more than one task, either *weakly* or *strongly*.
- *Local sensing and actuation*: a robot can only observe and perceive the nearby environment through its sensor channels. Additionally, its actuation can only have a direct and

immediate influence on the robot itself and on its close surroundings.

- *Local communication*: the interactions and communication among the robots must be local.
- *Simplicity and homogeneity*: each robot in the swarm must be simple by its own. Moreover, all the robots must be homogeneous, endowed with the same sensors and actuators, and controlled by equivalent robot controllers.

1.3 Thesis structure

This PhD Thesis is divided into the following chapters:

- Chapter 2 provides a detailed background on the SR field, emphasising its essential features, main control techniques, and some of the most relevant tasks commonly addressed in SR.
- Chapter 3 reviews the topic of the emergence and evolution of communication in multi-agent systems. Firstly, it defines the concept of communication, its types, and some of its properties. Thereafter, it reviews some of the state of the art studies tackling the evolution of communication in SR, focusing on the origins and evolutionary conditions for the emergence of signaling. Finally, a theoretical background on the emergence and pillars of language in artificial life systems is presented.
- Chapter 4 tackles the emergence of basic signaling communications in three different primitive collective tasks. This chapter focuses on the acquistion of *weak transferability* communications in SR, proposing a mininal communication system that can be used for SR tasks without its re-design.
- Chapter 5 focuses on the acquistion of evolved communications with *strong transferability*, so that the exact same evolved system can be used in multiple SR tasks. For that aim, a novel communication module that SR systems can use to coordinate and interact is presented. It is based on virtual state spaces, where the communication among the robots occur. The correct functioning of the communication module and its *strong transferability* is assessed in three popular SR tasks, namely, the physical aggregation of the robots in groups of a desired size, the formation of target swarm geometries, and a basic foraging task with temporal role allocation.
- Chapter 6 presents a new SR system that addresses one of the main challenges of communication systems in SR, as it is the transition from signaling to language-like communications. The aim of the chapter is the self-organization of a common lexicon shared by all the robots in the swarm, to refer to different geometric objects scattered in the environment. Moreover, compositionality, that is one of the fundamental properties of complex languages, is also addressed and obtained.
- Chapter 7 provodes a general discussion of the Thesis and a synthesis of the main results achieved.
- Chapter 8 presents the conclusions and future lines of research of this PhD Thesis. It

reviews and discusses the work done. It also details all the author contributions and publications resulting from the development of the PhD Thesis.

Finally, Appendices A and B provide a detailed description of some of the mathematical models and algorithms used in the chapters of the PhD Thesis. Specifically, Appendix A provides a theoretical background of the Genetic Algorithms (GA) and the NeuroEvolution of Augmenting Topologies (NEAT) algorithm, whereas Continuous-Time Recurrent Neural Networks (CTRNN) are described in Appendix B.

Chapter 2

Swarm Robotics

2.1 Definitions and properties

Swarm Robotics (SR) [4, 3] is a research field, resulting from the intersection of Artificial Intelligence (AI), robotics and Multi-Agent Systems (MAS). It studies the use of multiple simple and self-sufficient robots that cooperate to solve common complex problems. The term swarm robotics was initially defined with the aim to redirect the already established discipline of Swarm Intelligence (SI) to the robotics systems [3]. In turn, the term *swarm* was originally a replacement for the not so appealing terminology of *cellular robots* and *cellular automaton*. One of the most important aspects of SR is its decentralized paradigm, implying the lack of a central computing or *intelligent agent* that coordinates the others. All the robots have the same *a priori* importance within the swarm, so that the failure or loss of a member is readily absorbed by its neighbors.

The concept behind a swarm in AI is also highly tied to the *emergence of behavior* and the *self-organization*. Emergence is the phenomenon leading to the arising of novel macro-level patterns, properties and structures in complex dynamical systems, resulting from the interactions of their local micro-level components [17, 18]. In contrast, Dempster defines self-organisation as "(...) the systems that organise themselves without external direction, manipulation, or control" [19]. The authors in [18] complement this definition as follows:

> "Self-organisation is a dynamical and adaptive process where systems acquire and maintain structure themselves, without external control." [18]

Even though emergence and self-organisation share some similarities, such as time dependence or robustness, they are separate concepts that describe different parts of a dynamical system [18]. Consequently, a system can display self-organisation without emergence and vice versa. Nonetheless, inspired by self-organising systems that exhibit emergent collective behaviors in nature [20], in SR systems both self-organisation and emergence coexist simultaneously.

In SR, collective behaviors emerge globally, at the swarm level, from the local and simple interactions among the robots. Şahin defines the SR field as follows:

> "Swarm robotics is the study of how large number of relatively simple physically embodied agents can be designed such that a desired collective behavior emerges from the local interactions among agents and between the agents and the environment." [4]

According to this definition, the concept of *embodied agents* is also highly relevant to properly characterize SR systems. Embodied agents can be defined as individuals that are situated and capable of interacting with the environment and modifying it.

In addition to the former definition, Şahin describes in [4] a set of characteristics that SR systems require in order to be distinguished from other similar terms, such as collective robotics or robot colonies:

- *Autonomous robots.* The robots that compose the swarm are self-sufficient, situated and embodied. Although they require interactions with other neighbors to achieve collective behaviors, the agents can behave and interact with the physical environment in isolation. As stated in [4], a sensor network or a metamorphic robot system are not categorized as robot swarms. For instance, under this terminology, a modular centipede robot [21, 22], or an hexapod robot where each leg is seen as an independent agent, are not strictly considered SR systems. The reasoning is that each agent cannot exist as an autonomous unit without being seen as a whole.

- *Large number of robots.* The swarm size or number of robots is an important characteristic of SR systems. It is difficult to establish a lower boundary for the number of robots in SR, as there are published studies ranging from tens to thousands of robots [23]. In [4], Şahin clarifies that the relevant characteristic is rather related to the scalability of the system. That is, the invariance and low degradation in collective behavior and emergent features as the swarm size increases.

- *Homogeneous robots.* The robots should be identical or very similar in terms of capabilities, resources and "*intelligence*". Generally, the agents are equipped with the same sensors and actuators, and they are governed by the same robot controller logic and local rules. However, it should be differentiated between two types of heterogeneity: (i) innate and programmed heterogeneity, and (ii) the one that arises as a result of self-organisation, starting from the same initial conditions. For instance, a leader of the swarm may be manually selected by a human researcher (innate heterogeneity), or elected through distributed negotiation and self-organization [24, 25] (emerged heterogeneity). A relaxation to the homogeneity property would be that swarms of few groups or clusters of homogeneous agents can also be considered as SR systems, provided that the number of members of each group is relatively large.

- *Simplicity and inefficiency.* In spite of being self-sufficient, the robots are notably simple, inefficient and unable to tackle the SR task individually. It is the coordination and interaction among simple units the process that gives rise to complex behaviors and features. In general terms, this characteristic is met when the aggregation of robots clearly leads to a higher performance in the problem being addressed, when compared to the results provided by a single agent.

- *Local sensing and communication.* The robots can perceive the environment via the equipped sensors. Nonetheless, perception is local and constrained to its nearby surroundings. Robots should not have access to global information or references, nor perceiving a global picture of the whole swarm state. Additionally, the action of the robots should only influence the environment surroundings and should only influence on local neighbors.

Other similar characteristics are also provided in [26] and [3]. In case of not fulfilling some of these properties, a MAS can be categorized as a collective robotics system or as a Multi-Robot System (MRS), among other possible terminology.

2.2 Relevant properties

In addition to decentralization and self-organisation, that are inherent characteristics of SR systems, there are three additional properties that robot swarms desirably should fulfill in most SR tasks and aplications. These properties, that are defined and explained below in this section, are scalability, robustness, and flexibility [4].

Scalability

A robot swarm is scalable when its performance and efficiency in solving some collective task is not considerably affected by the number of robots in the swarm, i.e. the *swarm size*. The local and decentralized rules that govern the robots' behavior should not be affected by the total number of individuals. However, there is an unavoidanble degradadation resulting from the increase of the swarm size. The degree of such performance deterioration is highly dependent on the task being addressed, and it is mostly produced in terms of convergence time. For instance, a formation control behavior, in which the robots have to aggregate in certain patterns or shapes, would converge much faster and more accurately with fewer robots. On the contrary, a collective exploration, patrolling, and surveillance tasks are generally solved faster and more efficiently as the swarm size grows, up to a certain level. This issue was precisely addressed Kwa et al. in [27] in a collective tracking task.

Robustness

A SR system is said to be robust if it can overcome and respond successfully to some unexpected external perturbation during the accomplishment of some task. Such perturbation is normally related to the loss or failure of individuals belonging to the swarm (fault tolerance) [26], although, more generally, it can be any type of disruption (e.g. increse of noise, unexpected behavior of some robots, or unforeseen changes in the environment). Robustness and fault tolerance is in line with SR control paradigm because of the following reasons [4]:

(a) the high redundancy caused by large swarm sizes and homogeneous behaviors,

(b) the distributed control that is essential in any robot swarm, and

(c) the small contribution that each individual provides to the global emergent behavior.

Flexibility

A robot swarm is flexible if it can successfully perform its required behavior under diverse environments, and tasks [4, 26]. The former implies that the robot swarm behavior is not biased to a certain type of environment or scenario. For instance, a flexible collective motion behavior should perform similarly with and without obstacles, and regardless of their shape and dimensions. The latter form of flexibility is more challenging and less common in SR studies. It states that a flexible robot swarm should be composed of individuals capable of assuming different roles and performing multiple tasks. A representative behavior of this condition is *foraging* [28], in which robots can assume different roles, such as exploring the environment, foraging from known areas, or waiting in the nest. However, most SR behaviors in which robots can perform different tasks can be categorized as *division of labour*, so that a complex problem is divided into lower-level primitives. Finally, even though these two concepts are not typically associated, flexibility can be also related to the capacity of a robot swarm to behave similarly both in simulations and in real-world missions (minimization of the *reality gap*).

2.3 Sources of inspiration

The research field of SR is utterly inspired and influenced by diverse collective behaviors in nature and biological systems. Some of these social behaviors are the following:

- **Ants** (Figure 3.1A): Swarm robotics and swarm intelligence have been vastly inspired by the social behavior of ants. Ants exhibit an extraordinary variety of collective and swarming behaviors. For instance, ants perform division of labor and allocation of different roles within the colony [29] based on multiple factors, such as their age, size, morphology, or caste (e.g. queens and workers). Additionally, ants perform collective foraging behaviors mediated by *stigmergic* communication, using pheromone trails. Specifically, ants leverage this communication through the environment to create chain formations that connect colony nests with areas that are rich in food and resources. During their foraging round trips, ants deposit pheromones that can be perceived by other colony members. Furthermore, ant foragers tend to follow the trails with highest density of pheromones. It has been observed that through this stigmergic process ants can create chemical trails that connect to the richest food source through the shortest path [30, 31, 32].
- **Honey bees** (Figure 3.1B): The honey bees perform a complex waggle dance to notify others about the location of resources [33]. It is a signaling process in which honey bee workers communicate the location of a resource or target spot, namely, pollen, nectar or water source, through an intricate dancing. In [33], Karl von Frisch observed two different motion patterns: the *round dance* and the *waggle dance*. Both dances are engaged inside the hive, and encode information about the distance and orientation of the target areas. The round dance is performed by worker bees when the resource is close to the hive. In contrast, the waggle dance is accomplished when the resource spot is farther than 150 meters from the hive. The waggle dance is, moreover, composed by two main stages that are interleaved. Firstly, there is a *waggle phase* in which the honey

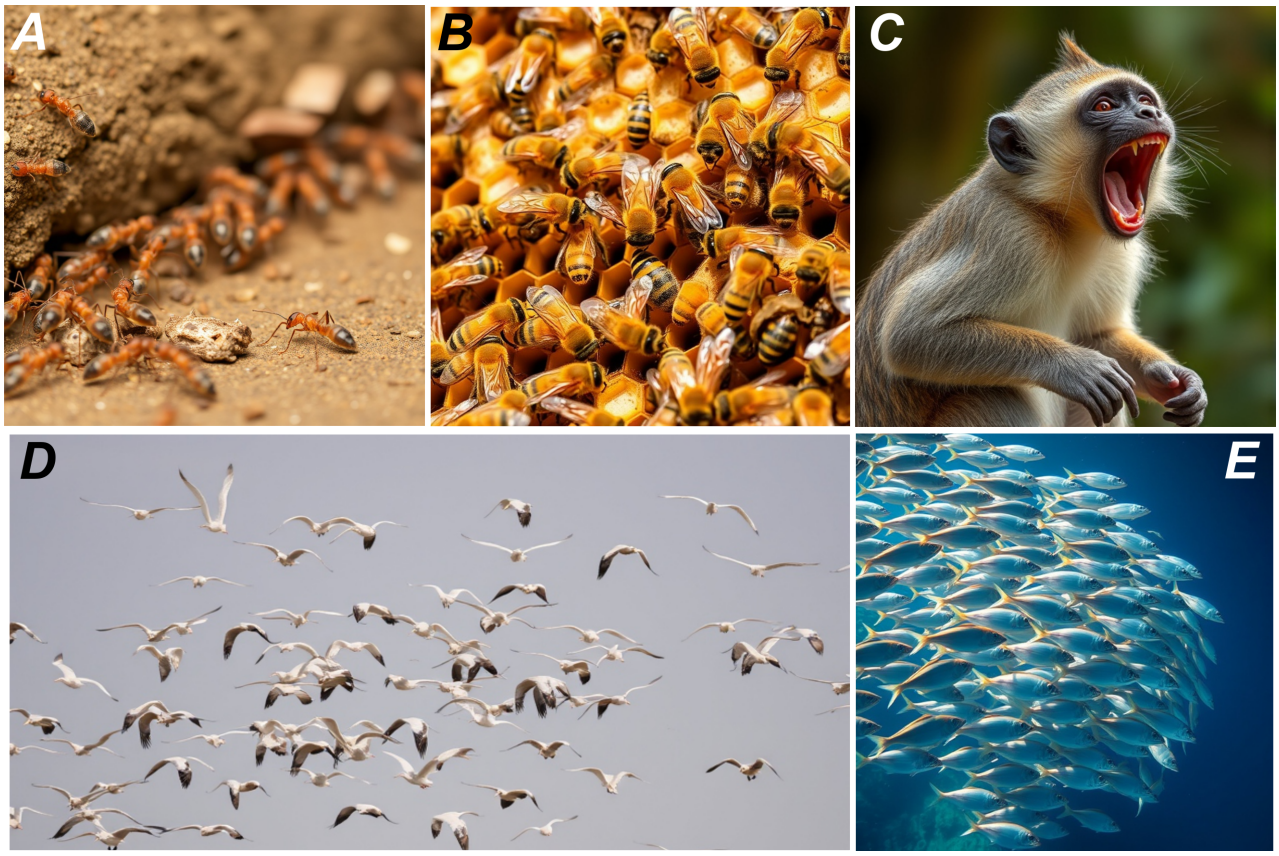


Figure 2.1: Collective behaviors in nature. (**A**) Chain of ants guided by pheromone trails. (**B**) Honey bee waggle dance. (**C**) Vervet monkey. (**D**) Flock of birds. (**E**) Fish school. All the images were generated using AI from https://deepai.org/machine-learning-model/text2img

bee describes a tail-wagging movement. Thereafter, it is followed by a *return phase*, that returns the dancing movements to its initial state. The sequencing between waggle and return stages occurs periodically. Karl von Frisch believed that the dance language of honey bees was composed by these two discrete and differentiated motion patterns. Nonetheless, in [34], the authors showed evidence that rather than two discrete states, the round and waggle dances are two extremes of a continuum.

- **Vervet monkeys** (Figure 3.1C): another interesting example of social behaviors in nature is shown in communities of vervet monkeys, that warn others about the presence of predators through different alarm calls. Futhermore, diverse calls are produced depending on the context and the detected threat, so that hearing monkeys can distinguish the predator approaching and respond accordingly [35]. For example, three accoustically different alarm calls are generated when either leopards, snakes, or eagles are sighted, and, upon receival, listening monkeys react by either climbing a tree, looking down to the floor, or looking up to the sky, respectively.

- **Flocks of birds** (Figure 3.1D): groups of birds from many species show social behaviors, characterized by an impressive and intricate collective motion. Such coordinated avian flight is commonly refered as flocking [36] (term not restricted to bird species, but rather

to any group organisms that describes a gregarious collective motion), and it is generally produced during migratory flights, foraging, and protection from predators. The most common flocking formations are line formation (or V-shape), and cluster formation [37], which depend on the species and the number of birds, among other factors. Flocking has been typically modeled as a fully decentralized process that is characterized by three simple rules: *cohesion*, *separation*, and *velocity matching*. The former ones are two complementary behaviors that aggregate birds together while preserving a minimum separation to close neighbors. Through the latter behavioral rule, *velocity matching*, the flocking individuals tend to converge to the same direction of motion.

- **Fish schools** (Figure 3.1E): is another example of a fully decentralized collective behavior in nature. Flocking behaviors are also present in many fish species, exhibiting swarm behaviors of aggregation, three dimensional pattern formation, and coordinated motion, in the absence of a leader. Similar to flocks of birds, the individuals in the fish school tend to approach to other nearby fishes while preserving a minimum separation distance, and mimimicking their direction of movement with high precision. When observed from the distance, the aggregated fish school, that may be composed of up to few millions of fishes (depending on the species), can be mistaken for a large animal [38]. Collective behaviors of fish schools have been observed in a wide variety of activities such as traveling and migration, energy preservation, chasing away and evading predators, resting, and feeding on placton or preys [38].

2.4 Methods and algorithms

2.4.1 Manual design methods

As suggested by its name, in manual design methods the overall logic, mechanics, and functioning of the robot controllers are handcrafted by a human expert. In other words, a human engineers the behavior of the robots in the swarm. In most scenarios, manual design paradigm follows a *bottom-up* approach, so that the designer figures out the low-level local rules of each individual that give rise to the desired global behavior of the robot swarm. The *bottom-up* approach is generally based on a combination of trial-and-error and expertice in the field [26].

Physics-based approaches

In physics-based SR systems, robots are seen as particles that are attracted or repelled among them based on physics-based virtual forces. In 1998, J. H. Reif and H. Wang [39] proposed the *social potential fields* as a distributed control method for Very Large Scale Robotics (VLSR) systems, which is a similar and prior concept to SR. The robots are abstracted as particles in molecular systems, interacting based on inverse-power law virtual forces. Nonetheless, social potential fields differ from molecular systems in two major aspects: (i) they allow different force laws for each pair of particles, and (ii) virtual forces are not necessarily symmetric.

The force applied by robot j to robot i, with positions X_j and X_i, is computed as in

Equation 2.1

$$F_{i,j} = \left(\sum_{k=1}^{L} \frac{c_{i,j}^{(k)}}{r_{ij}^{\sigma_{i,j}^{(k)}}} \right) \left(\frac{X_j - X_i}{r_{ij}} \right), \tag{2.1}$$

where r_{ij} is the Euclidean distance between the robots, and $c_{i,j}^{(k)}$ and $\sigma_{i,j}^{(k)}$ are coefficients that define independent laws between each pair of robots i and j. Notice that the first term corresponds to the magnitude of the virtual force, while the second term is the direction of the force vector.

The total force applied to some robot i is defined in Eq. 2.2,

$$F_i = \sum_{j \neq i} F_{i,j}(r_{ij}). \tag{2.2}$$

Additionally, they also defined virtual force laws between groups of robots, motivated by the idea that robots inside each group are expected to perform common tasks or similar behaviors. Therefore, the emergent formations and self-organized behavior generally diverges from other agents engaging in some other tasks. In such cases, the final force applied to a particle is computed as a two step process. Firstly, the intra-group virtual forces are computed, and, subsequently, inter-group social forces are added to obtain the final force vector.

In addition to the inverse-power law model, they extended social potential fields using virtual spring laws. Spring laws differ from the inverse-power law model because forces between robots are necessarily symmetric. They abstracted this model as a undirected graph where the nodes are the robots and the edges represent the pairwise interactions between the robots using the spring laws. They defined a spring graph which is *rigid* in some d-dimensional space, so that the *local-minima problem* is avoided when having multiple minimum-energy embeddings. The social potential fields allow the convergence to predefined structures and patterns and dynamically transition between them via the distributed control of few robots in the swarm.

One of the most relevant and early works under the frame of virtual physics in SR is [40]. The authors introduced an artificial physics framework called "*physicomimetics*" for formation and motion control. The agents are represented as physical particles that sense other particles in the form of virtual forces, that are used to modify their positions accordingly. Depending on the type and source of the interaction, forces can be either attractive or repulsive. The framework behaves based on simulated molecular dynamics in order to minimize the overall system potential energy. The authors highlighted some key features and advantages resulting from using physics-based approaches, namely, self-organization, fault-tolerance, and self-repair (essential properties of SR systems, as described in Section 2.2).

In this framework, every particle has a position $\overrightarrow{x}$ and a velocity $\overrightarrow{v}$. Positions are modified according to the current velocity of the particle (see Eq. 2.3), while the velocity of the particles is updated based on the resultant applied force (see Eq. 2.4).

$$\overrightarrow{x} = \overrightarrow{v}\Delta t \tag{2.3}$$

$$\overrightarrow{v} = \overrightarrow{F}\Delta t/m \tag{2.4}$$

In addition to virtual forces from other particles there is an additive frictional force. The module of the virtual force $\overrightarrow{F}_{ij}$ between two particles i and j is computed as in Eq. 2.5,

$$F_{ij} = G\frac{m_i m_j}{r^p}, \tag{2.5}$$

where m_i and m_j are the masses of the two interacting particles, G is a gravitational constant and p is a parameter defined by the user that establishes the relation between distance and force module. There is a maximum force, F_{max}, that a particle can apply to another. A virtual force $\overrightarrow{F}_{ij}$ is attractive if the distance between particles i and j is greater than some radius R. Alternatively, the force is repulsive provided that the distance is less than R. The physicomimetics framework also considers obstacles as repelers that apply repulsion forces to particles to avoid collisions. Goal or target spots are treated as attractors in the virtual force field, so that particles tend to converge to them.

Many authors have proposed other *physicomimetics* systems inspired by other physics phenomena and dynamics beyond molecular systems, such as gas particles [41], viscoelastic forces based on the Voigt model [42, 43, 44], Active Elastic Sheets (AES) [45, 46, 47], and liquid spheres [48].

Probabilistic finite state machines

Probabilistic Finite State Machines (PFSM) are composed by a fixed set of states $(S_1, \ldots, S_N)$, and stochastic transitions between states. The state transitions, T_{ij}, have one origin state S_i and one destination state S_j, and such transition may occur with some probability p_{ij}. In some cases, stochastic transitions can be combined or replaced by deterministic conditions, so that in this PhD. Thesis a PFSM is undestood as a finite state machine with at least one stochastic transition. PFSM states are associated to robot behaviors, such as random walk, explore, or stay (in most studies the name of the state self-explains the associated behavior). Moreover, transition probabilities can be fixed from the beginning of the execution or dynamically modified based on some external variable, such as the number of neighbors (see e.g. [49]). PFSMs have been mostly used to solve aggregation [50, 51, 52, 49, 53] and foraging [54, 6, 55, 56, 57] tasks.

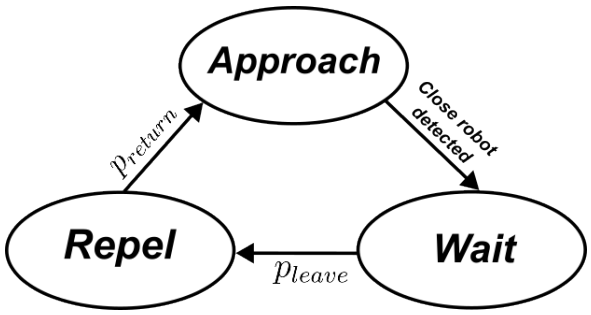


Figure 2.2: Example of a PFSM used in an aggregation task, adapted from [50].

The PFSM used in an aggregation task [50] is provided as representative example (see Figure 2.2). It has three states corresponding to the robot behaviors (i) approach to other

robots, (ii) move away from nearby robots, and (iii) wait. It has two stochastic state transition probabilities, p_{leave} and p_{return}, and a deterministic transition when another robot is detected very closely.

2.4.2 Automatic design methods

In contrast to manually designed controllers, automatic desing methods rely on optimization algorithms to find a suitable and effective robot controller solutions. The optimization mechanism operates on some parametrized model that is commonly based on artificial neural networks. However, optimized robot controllers based on artificial physics models [58, 59] and PFSM architectures [6, 55] also fall into this category. The most used optimization algorithms in SR are evolutionary algorithms (e.g. [60, 61, 62, 58, 63, 60, 64, 65, 66, 55, 67]), reinforcement learning (e.g. [68, 69, 59]), and swarm intelligence algorithms (e.g. [70, 71, 47]). In contrast to the manual design methods, the automatic design of robot controllers follows a *top-down* approach. Specifically, the researcher establishes the loss or fitness functions that describe the behavior of the robots globally (swarm level), and it is the optimization algorithm the mechanism responsible for finding the appropriate behavior of each individual (robot level). In most cases, furthermore, the optimized solution is seen as a black box, and, in the best case scenarios, researchers can only gain some insight about the resulting robot behaviors after an exhaustive analysis and deciphering.

Evolutionary robotics

Evolutionary Robotics (ER) [72] refers to the use of evolutionary computation to optimize the parameters of some robotic controller. It is also closely related to the term *neuroevolution* [73], which is strictly used when the evolutionary algorithm is applied to neural controllers (based on ANNs). In such case, the ANN acts as the robot controller, being fed by the sensor measurements and returning actions that directly operate on the robot's actuators. Initially, the neural controller is an arbitrary and randomly initialized model, so that it is the evolutionary algorithm the process that is responsible for defining the final individual behavior of the robots. ER has been successfully applied to most SR tasks, such as aggregation [74, 60, 61], flocking [63, 62, 58, 60, 65, 64, 66], or foraging [55, 75, 76, 67]. Moreover, it will be described in subsequent chapters that ER is a fundamental optimization technique for reaching the emergence of communication in robot swarms from initially non-communicative behaviors (see e.g. [77, 78, 79, 80, 81, 1, 82], and Chapter 3 for more details).

Within the family of evolutionary computation algorithms, the most widely explored in SR are standard Evolutionary Algorithms (EA) [74, 60, 64], Genetic Agorithms (GA) [7, 55, 75, 76], NeuroEvolution of Augmenting Topologies (NEAT) [61, 67], and Evolution Strategies (ES) [63, 62]. In all of these methods, a population of candidate solutions (genotypes) is iteratively evolved in order to reach a final robot controller that satisfies the requirements of the SR task. The genotypes encode the parameters of the robotic controller being optimized, which can be the weights of an ANN, the transition probabilities of a PFSM, or the weights of artificial physics models. Depending on the optimized model, the number of parameters, and the task constrains, such coding can be binary-coded (e.g. in [74, 60]) or real-coded (e.g. in [7, 64]).

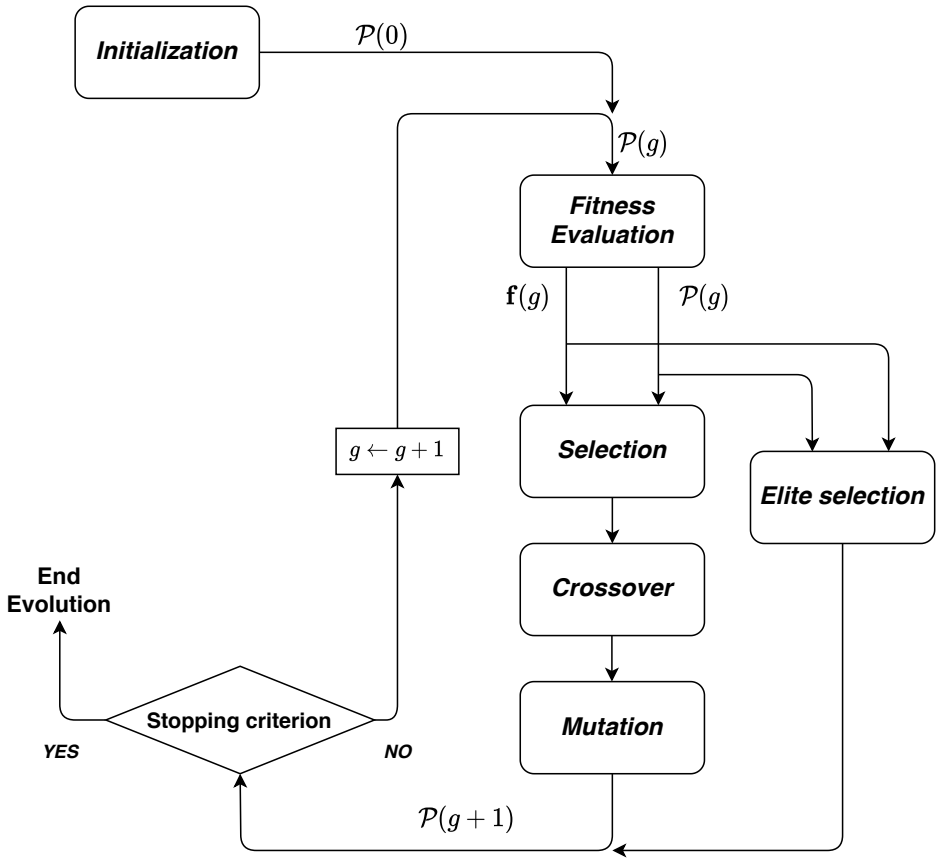


Figure 2.3: General diagram of a genetic algorithm.

Most evolutionary algorithms can be characterized by the following stages (see also Figure 2.3):

- *Evaluation*: each genotype is evaluated in the corresponding task, and the overall robot swarm performance is quantified with a fitness score. The evaluation process requires an sufficient number of large duration robot executions, which is not a feasible option in most is experiments with real robots. Therefore, it is generally performed using realistic robotic simulations, which are transferred to the real robots after the artificial evolution is ended.

- *Evolutionary operators*: Every generation, a set of operators is applied to the population of candidate solutions in order to give rise to the next generation. Even though each algortihm uses a specific operator pool, the most common ones are *selection*, that filters out the fittest individuals to be preserved, *recombination*, that merges pairs of genotypes (parents) to create new combined solutions (offspring), and *mutation*, that slightly alters some genes of the population to trigger the exploration of unseen areas of the search space.

- *Stopping criteria*: the evolution typically ends when (i) the evolution reaches a predefined number of total generations, (ii) a sufficiently good average or maximum fitness score of the population is reached, or (iii) the average fitness score of the population becomes stagnant or consistently decreases.

- *Deployment*: in some studies, the evolved robot controller is finally tested in real environments with real robots. In these cases, the simulations used during the evolution process must be sufficiently realistic in order to minimize the reality gap.

A highly relevant evolutionary algortihm for this PhD. Thesis is NEAT [83]. NEAT is an extension of standard GAs exclusively oriented to the evolution of neural controllers. Its most outstanding feature is that it evolves both the parameters and the topology of the ANN itself. Among many other characteristics, it proposes novel recombination operators to combine the neurons and synapses from two different ANNs. The reader is redirected to Appendix A for further theoretical details on GA and NEAT, which will be repeatedly used during this PhD Thesis.

Reinforcement learning

Reinforcement Learning (RL) [84] is a learning paradigm in which agents learn a desired behavior through trial and error and interaction with the environment.

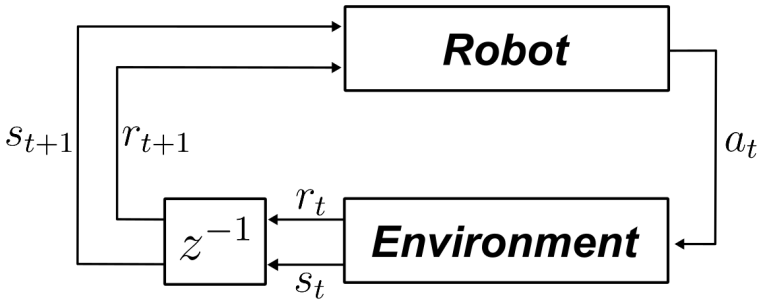


Figure 2.4: General diagram of RL algorithms.

Figure 2.4 shows the general diagram of RL learning paradigm. Its most simplified version contains two building blocks, the robot, and the environment where it is embodied. The current state of the robot (s_t) refers to its current situatedness within the envirnoment (e.g. its position, orientation, ...). Such state can be a holistic view of the whole environment, or a partial observation biased by the agent's location and perception capabilities (which is the most frequent scenario in SR). In RL terms, the agent is defined by some *policy* (π), that transforms the current state into actions to interact with the environment. In essence, the *policy* is analogous to the robot controller and defines the behavior of the agent. At each control cycle (t), the autonomous agent interacts with the environment via some action (a_t). The influence that such action has on the environment leads to modifications on the agent's state (e.g. changing its location), producing a next state (s_{t+1}). Moreover, the performed action potentially produces some future reward (r_{t+1}). Such state-reward pair (s_{t+1}, r_{t+1}) will be perceived by the robot the next control cycle, influencing subsequent actions.

The last aim of the agent is to maximize the total discounted return (G_t) at any time instant

of an episode (task execution). The computation of G_t is shown in Eq. 2.6,

$$G_t = \sum_{k=t}^{T} \gamma^k r_{t+k+1}, \tag{2.6}$$

where $\gamma \in [0, 1]$ is a discount factor, and it is assumed that episodes have a finite duration T.

In order to learn a policy that maximizes the long-term discounted return, it is further extended to the *value function* $V^\pi(s)$ (see Eq.2.7a), and to the *action-value function* $Q^\pi(s, a)$ (see Eq.2.7b),

$$V^\pi(s) = \mathbb{E}\left[G_t | s_t = s\right] \tag{2.7a}$$

$$Q^\pi(s, a) = \mathbb{E}\left[G_t | s_t = s, a_t = a\right] \tag{2.7b}$$

Considering an agent that follows policy π, $V^\pi(s)$ is the expected discounted return that results from being in state s at time step t and following policy π thereafter. Similarly, $Q^\pi(s, a)$ is the expected discounted return resulting from being in state s at time step t, performing action a and, subsequently, following policy π.

Reinforcement learning algortithms are generalized to the multi-agent scenario as Multi-Agent Reinforcement Learning (MARL), which is the variant that is normally used in SR. In MARL, rewards are typically computed at the swarm level, so that robot policies have to consider states and actions both from itself and from the other agents [85]. Moreover, other examples of additional challenges are the communication among the robots, or whether the robot policies are homogeneous or heterogeneous (see also [85]). MARL has been used to optimize the robot controller of SR systems in flocking [68, 69, 59], formation [86, 87], foraging [88], and path planning [89]. These studies employed diverse RL algorithms, such as SARSA [87, 89], Q-learning [68, 86, 88], and DDPG [59], among others.

2.5 Swarm robotics behaviors

Inspired by collective behaviors in nature, the SR paradigm has been used to solve a wide variety of problems and tasks. The collective behaviors exhibited by the swarm to solve the target tasks can be divided into *primitives* and *composed* behaviors (hereafter, the term *primitive* is indistinctly used for both the robot behavior and the task that it tackles). Firstly, *primitives* are basic tasks that are generally solved by means of simple and specific behaviors. The concept of primitive is based on the levels of competence in [90] and on the behavior primitives in [91] or in [92]. Representative examples of primitive behaviors are obstacle avoidance, *phototaxis* (approach to a light source), explore the environment, or approach to other robots. Secondly, composed tasks are broader problems that require the composition of multiple primitive behaviors in order to be solved. A clear exponent of this category are the Reynolds' rules [93] designed to model flocking behaviors of birds and fish schools in nature. Reynolds proposed three primitive behaviors, (i) cohesion, (ii) separation, and (iii) alignment, that when combined together give rise to efficient flocking movement patterns in swarms of robots.

2.5.1 Primitive tasks

Aggregation

Swarm aggregation is a basic SR primitive behavior in which robots reunite in a common location (either arbitrary or agreed). Aggregation is a fundamental primitive in SR as it can be used in most complex behaviors, such as formation, flocking, or foraging, among others. It essentially avoids swarm disconnection and dispersion, which is a fundamental initial requirement in most of tasks.

Probabilistic models, mainly based on PFSM architectures, have been widely explored in the context of aggregation behaviors. These studies normally combine the microscopic and macroscopic aspects of the aggregation primitive. Microscopic behaviors refers to the control of a single robot in the swarm, whereas macroscopic analysis refers to predictive models that estimate the final outcome and evolution of the overall aggregation dynamics. Probabilistic robot controllers are mostly inspired by cockroach aggregation behaviors in the real world [94, 95].

The authors of [50] used a probabilistic behavior-based approach to tackle aggregation of mobile robots. Their controller was a subsumption architecture with two levels of competence and a total of four behaviors: obstacle avoidance, approach, repel, and wait. The lower level of the subsumption model was uniquely defined by the obstacle avoidance routine, whereas the higher level was a PFSM with remaining three behaviors (see Figure 2.2). The robots were equipped with sound sensors, that provide an estimate of the direction with highest concentration of robots. Robots in the *approach* state tend to navigate towards the direction of loadest sound, expecting to aggregate with other swarm members. A transition to the *wait* state occurs when the distance to a neighboring robot is estimated to be lower than some threshold. In such case, agent becomes static until, with some probability p_{leave}, a transition to the *repel* state is produced. Robots executing the *repel* behavior move in the opposite direction of the highest measured sound. Finally, robots can randomly return to the *approach* state with some probability p_{return}. Soysal and Şahin also studied probabilistic aggregation in [51]. The aggregation controller was a PFSM with three states corresponding to the behaviors *random walk*, *approach*, and *wait*. The general structure of the PFSM was similar to the one previously proposed in [50]. The main differences were using a random walk, instead of the *repel* behavior, that is accomplished with some proability p_{leave}. Additionally, robots transition from *random walk* state to either *approach* state, if some robot is perceived, or to the *wait* state, when some time was elapsed without perceiving other robots. Moreover, they proposed a macroscopic model used to estimate the final outcome and evolution of the aggregation behaviors.

A PFSM-based controller was also proposed in [52] to reach emergent aggregation behaviors in physics-based simulations. The authors used a very simple aggregation PFSM with two states and behaviors, random walk and wait. They also included a guard period when a robot transitions to the random walk state, so that the next transition cannot occur before such period is elapsed. An important contribution of this study was that, unlike many probabilistic aggregation models, the robots lack of sensing capabilities to detect the number of aggregated robots. In [49], the authors addressed the aggregation task using a subset of *informed* robots

in the swarm. The only difference between the *informed* robots and the remaining agents was that the former ones know the target aggregation locations. Specifically, there were two types of aggregation areas, in white and black, and the *informed* robots were capable of distinguishing in which type of area the swarm should agregate to correctly solve the task. As in previously reviewed aggregation studies, the robot controllers were based on PFSMs with three states, namely, random walk ($\mathcal{RW}$), stay ($\mathcal{S}$), and leave ($\mathcal{L}$). The probabilities p_{stay} and p_{leave} involved in the stochastic transitions $\mathcal{RW} - \mathcal{S}$ and $\mathcal{S} - \mathcal{L}$, respectively, were not fixed. Instead, these probabilities depended on the number of other robots in the surroundings. The PFSM used by the *informed* robots was extended using the privileged information about the target location. The authors did not only assess whether it was possible for a subset of *informed* robots to guide a complete swarm towards desired aggregation zones, but they also studied the minimum proportion of *informed* agents required to accomplish such task. Their results revealed that successful aggregation in the correct spot was achieved when (i) at least 20% of the robots were informed in symmetric scenarios (same amount of black and white areas), and (ii) between 20% to 30% of the robots (depending on the swarm size) were informed in assymmetric experiments (more number of white areas).

An aggregation study that is highly relevant for this PhD Thesis is one published by Cambier et al. in [53]. The authors used a minimal naming game to discover and propagate suitable sets of parameters of PFSM models. The PFSM had two states, *walk* and *stay*, and dynamic transition probabilities that depend on the number of neighbors. Candidate parameters of the PFSM were stored as words in the robots' lexicon. Words were negotiated and culturally shared along the swarm through the engagement in language games, until all the lexicons contain only one (and the same) set of parameters. Moreover, the aggregation dynamics also influenced the course of the language games in order to obtain high quality solutions. For instance, only agents in the *stay* state were allowed to act as speakers during the minimal naming games. This paper will be revisited in Chapter 3, under the frame of language evolution and emergence of communication.

In addition to probabilistic approaches, several authors have also applied virtual physics to the design of aggregation controllers. The study in [96] explored the use of Lagrangian-based models in a swarm of point masses, in the absence of obstacles. Agents were attracted to the Center of Mass (CoM) of the swarm, while being repelled by other individuals. Their simulations revealed that different behaviors arise from their proposed dynamics when the model parameters are adjusted. One of these behaviors was an aggregation of the swarm while preserving the location of the CoM. Gasparri et al. proposed in [97] a physics-based approach in which robots aggregate through their interaction via attraction and repulsion forces. The authors also included obstacle avoidance by treating nearby obstacles as virtual robots, and including additional repulsion forces when required. In [44], the authors used the k-nearest neighbors algorithms to detect the neighbors that contribute to the computation of attraction and repulsion forces dictated by physics-based dynamics. They expored three different machine learning distance metrics (i.e. cosine, angular and Minkowski distances) to compute the k nearest neighbors. Physics-based dynamics among the computed k nearest neighbors were based on the Voigt model, using virtual viscoelastic connections.

Finally, evolutionary robotics have been also used to obtain aggregation behaviors. Trianni et

al. used in [74] a generational EA that optimzed the weights of a simple perceptron ANN. The weights were enconded as binary numbers in the genotypes of the evolutionary algorithm. The fitness function was computed using the distances of the robots to the CoM of the swarm. Specifically, the fitness was higher with the increase of the average distance to the CoM. Robots were equipped with a speaker that generated a constant sound during the entire execution time. Therefore, a robot could perceive neighbors either from the emitted sound using microphones, or through proximity sensors. After the evolution processes, the authors found two different aggregation behaviors: a static aggregation in clusters, and aggregation in dynamic clusters, that resulted in a collective motion after aggregation was accomplished. One of the tasks addressed in [60] was the aggregation of *s-bots*. They used an evolutionary algorithm with binary-coded genotypes that encode the weights of a simple ANN (phenotype). The ANN inputs were sensory information from eight proximity sensors and three sound sensors, whereas the outputs of the neural controller were responsible for controlling the motor wheels. The fitness function that guided evolution towards the desired aggregation behaviors was a trade-off between an aggregation quality term, and a motion quality component. The authors of [61] evolved aggregation behaviors using the NEAT algorithm and novelty search. Unlike most evolutionary algorithms, that use the fitness score to guide the evolutionary process, novelty search allows the guidance of artificial evolution also through the discovery of new behaviors. Novelty search is a highly powerful evolution method to deal with local optima, deceptive solution, and stagnation during the evolution process. They separately used three measures of the novelty of behaviors in novelty search: (i) a vector with the average of distances to the CoM at different time steps, (ii) a vector with the number of clusters at different time steps, and (iii) a concatenation of the two previous vectors.

Leader election

The leader election is an interesting primitive in swarm robotics as it can be used in a wide variety of tasks, such as aggregation, formation, or foraging, among others. The problem can be formulated with either one or many leaders of the robot swarm. However, the selection of the leader must be decentralized, and all the robots must initially have the same opportunities to assume the leadership. The leader election is a fundamental primitive that can be used to solve much more complex SR tasks, such as flocking [98] or goal navigation [99], among others. In [24], the authors designed a voting algorithm based on local communication in a static swarm for the leader election, among other cooperative tasks. The results of their experiments successfully showed consensus in the selection in most cases. The leader selection procedure in [100] avoided the usage of direct communication among robots as it only employed the positions of other agents in the neighborhood. A robot became leader whenever all the other robots in its vicinity lied on the same quadrant, considering the robot's position as the origin of coordinates. Alternatively, one of the multiple experiments in [91] was the leader election. They employed Wave Oriented Swarm Paradigm (WOSP) techniques in order to trigger the emergence of collective behaviors, such as leader election, with local binary information exchange. The controllers were also handcrafted as outstandingly simple and compact finite state machines. Neither evolution nor neural controllers were explored in these papers. On the contrary, in [101], the leader selection and role allocation problems were faced using neural controllers and evolutionary computation. The swarm members communicated locally

through a communication system and a robot could assume the role of leader by directly maximizing its communication output.

Heading orientation consensus

Heading orientation consensus or, simply, robot alignment, is a primitive task in which all the robots in the swarm have to point to the same direction. That is, upon convergence, all their heading orientations must tend to the same consensus value. It is one of the pillars of flocking behaviors (see Chapter 2.5.2), according to Reynolds' rules [93]. Consequently, most of the related SR studies have addressed the problem of heading orientation alignment in the context of the emergence of flocking. In [102], the alignment task was addressed with the aim of solving a self-organized flocking in mobile robot swarms. In their experiment, the robots used a virtual heading sensor through a digital compass, so that they were able to perceive and broadcast their own global orientation relative to a north reference. Similarly, heading alignment behavior was also addressed in [103]. There, within the swarm, there was a subset of informed robots that could directly access the direction of a common and global objective, using a light source as the global reference. The goal was that all the robots must alter their headings and point to the objective. The informed robots broadcast the acquired object direction, while the uninformed agents play the role of communication relays. The swarm members knew an absolute reference throughout measuring the light intensity emitted by a light source. The orientation consensus was evolved in [7], where an EA and a recurrent neural network were used. They focused on the emerged communication, showing that a situated communication emerged as a result of evolution. It was situated because only the physical information and the context about the environment attached to the message was relevant and communicative.

2.5.2 Composed tasks

Formation control

The formation control is a collective behavior in which robots have to self-organize themselves spatially in order to create a target swarm topology. Such goal topology can be a specific geometric structure (e.g. a triangle, square, or circle), or a repeating pattern. In both cases, robots are not associated (initially) to specific positions in the formation, and the convergence to the target geometry is purely obtained using a decentralized control of the robots.

The most widely explored family of robot controllers are based on physics-based approaches (see Section 2.4.1). Spears et al. [40] assessed their *physicomimetics* framework in formation tasks. However, due to the nature of the virtual physics approach, their formations were mainly based on regular patterns such as hexagonal and square lattices. In [104], robots were aggregated inside a specific geometric shape. An attraction force was applied to the agents outside the shape, while robots already gathered inside the target area were affected by repulsion forces. Both attraction and repulsion forces were nullified when an agent was in the shape boundary. Moreover, inter-robot repulsion forces avoided collisions among the robots, while maintaing a proper swarm diffusion inside the target shape. In [105], the authors also addressed formation control based on potential fields. They extended the

study in [104], so that the robots had to fit inside the perimeter of a goal shape in order to fulfill the desired formation. Qin et al. improved the similar formation control in [104] by fixing some of its limitations. For that, all their forces were based on exponential functions with two controllable parameters. Moreover, the authors dealt with some important issues such as navigation preserving the formation, collision avoidance, narrowing of the shapes to bypass complex obstacles like corridors, or formation change. Barnes et al. proposed in [106] a swarm formation in parametrized ellipse shapes. They used artificial potential fields combined with limiting functions to attract the robots towards a target elliptical band. In addition to the ellipse formation control itself, the authors also tackled obstacle avoidance, scalability, and heterogeneous swarms. In [42], formation control based on virtual spring-damper meshes was applied to Spacecraft swarms. Their virtual physics approach was a Virtual Spring-Damper Mesh (VSDM), subject to the effect of a gravitational field. Additionally, the authors of [43], obtained decentralized swarm formations circumscribed in a circle using virtual viscoelastic forces based on the Voigt model. Consequently, particles in the Voigt model (representing each robot) were located along the perimeter of a circle. Links between robots were modeled as virtual Hookean elastic spring and virtual Newtonian damper connected in parallel. They assessed the performance of the formation in real environments, using e-puck robots [107] equipped with eight IR proximity sensors and range and bearing technology [108]. A *physicomimetics* model based on liquid spheres was proposed in [48] to address formation control. In order to emulate liquid sphere phenomena, the authors used virtual spring networks and *physicomimetics*, so that robots were seen as point masses that interact among them through spring connections. Formations were not restricted to circular topologies, allowing arbitrary shaped formations through parameter variations. Moreover, they addressed collective navigation in formation, through tracking of a virtual target, and liquid-inspired obstacle avoidance. Their proposed formation control was scalable and flexible, so that new robots could be easily attached to the formation. Elkilany et al. [109] addressed formation control with collision avoidance and trajectory tracking. They used artificial potential field to accomplish this task, so that attraction forces were applied to the robots to converge to the target formation, and to navigate towards the goal location dictated by the trajectory being tracked. Complementarily, repulsion forces were applied to avoid obstacles and to guarantee a minimum separation between the robots. Moreover, an ANN with two hidden layers was optimized using the backpropagation algorithm with the aim of finding a suitable set of parameters of the potential field method.

Some authors have also faced the formation control using behavior-based approaches. In behavior-based approaches, multiple primitive behaviors (such as avoid obstacles, move towards goal, and so on) were implemented separately and combined using motor schemas or subsumption models. Balch and Arkin proposed in [110] a behavior-based approach to formation control and collective motion. Their model was based on several primitives (i.e. avoid static obstacles, avoid other robots, move towards goal, and maintain formation) that were combined using motor schemas. They tackled line, column, diamond and wedge (V-shaped) formations, that were preserved while agents performed collective motion and avoid obstacles. In [111], the authors also obtained formation behaviors using behavior-based approaches that make use of motor schemas. They implemented similar primitives as in [110], with the addition of a *wall-following* behavior that was required to bypass complex and large

obstacles. Moreover, the authors adressed the convergence to an initial target formation in large scale robot swarms, using classification-based searching algorithms. This method allowed the mapping between positions in any arbitrary target shape and robots of the swarm.

Another relevant method to solve formation control is through virtual structures [112, 113]. In [113], the authors used a virtual structure for moving multi-robot systems in formation with a high-precision. Agents formed a rigid geometric relationship with each other with respect to a common reference frame. In order to achieve the desired formation, the positions in the virtual structure and the physical positions of the robots influenced each other. In [114], the authors proposed a virtual rigid body scheme for achieving multi-robot coordination and formation. Specifically, they constructed a virtual space with its basis, boundaries, and constraints defined by real-world state variables. Moreover, they defined the goal that the robots must achieve as some coordinates in the virtual construct. Thereafter, they simplified multi-robot tasks to finding a possible path that connects the initial robot coordinates with the goal coordinates without entering into the constraint regions. In [115, 116, 117], the virtual path planning procedure with constraints as virtual obstacles was extended, and referred to as *motion planning on a Representation Space (RS)*, for solving diverse problems. Askari et al. proposed in [118] a formation control for swarms of UAVs, based on virtual structures, classical control theory, and inverse dynamics.

Flocking

Flocking is a fundamental behavior in SR that is based on the collective and coordinated motion of the individuals that compose the swarm (see e.g. [119]). The robotic controllers that generate flocking movement patterns are highly influenced by multiple collective behaviors in nature, such as flocks of birds, or fish schools. One of the earliest models of flocking was proposed by Reynolds in [93], called *boids model*. It was composed by three local and simple rules:

- *Cohesion*: the individuals tend to be as compact as possible, by moving towards the CoM of their neighbors.
- *Separation*: collisions between the individuals are avoided, and a minimum security distance is maintained.
- *Alignment*: the agents modify their heading orientation or direction of movement to match the orientation of their neighbors.

Reynolds showed that these three rules are sufficient to generate stable flocking behaviors. Additionally, Couzin et al. [120] defined three distinct circular zones centered at each individual i:

- *Repulsion zone*: is the area immediately close to agent i. Other individuals in this zone get repelled by the influence of i.
- *Orientation zone*: is an intermediate area in which the distance to agent i has reached a stable point (neither close nor distant). Therefore, agents in this zone tend to converge to the same heading orientation.

- *Attraction zone*: agents in this area are too far away from i, so an attraction mechanism is applied to reduce such distance.

An alternative family of flocking models is considered under the field of distributed control over graphs, being the Cucker-Smale model [121] the greatest representative. The Cucker-Smale flocking model considers a group of N self-propelled particles with three-dimensional positions x_i and velocities v_i. Thereafter, for some agent i, it proposes the continuous-time distributed control system in Equation 2.8,

$$\begin{cases} \dfrac{\partial x_i}{\partial t} &= v_i \\ \dfrac{\partial v_i}{\partial t} &= \displaystyle\sum_{j \in \mathcal{N}_i} \psi(\|x_i - x_j\|_2^2)(v_i - v_j), \end{cases} \tag{2.8}$$

where $\mathcal{N}_i$ is the neighborhood of agent i, and ψ is a positive decreasing function as in Equation 2.9,

$$\psi(d) = \frac{K}{(1-d)^\beta}. \tag{2.9}$$

They mathematically showed that convergence to a consensus velocity is guaranteed when $\beta < 1/2$, and dependent on the initial conditions $(x_i(0), v_i(0))$ when $\beta \geq 1/2$.

The Cucker-Smale model has been extended by multiple authors to consider collision avoidance [122, 123], finite-time control [124], noisy environments [125, 126], and hybrid models with swarms where continuous-time dynamic agents and discrete-time dynamic agents coexist [127].

Even though the Cucker-Smale flocking model and its extensions propose a mathematical formalism and guaranteed convergence, it may be limiting when real robots with constrained sensing, computing and communication capabilities are considered. Therefore, in the field of SR, other modeling paradigms, such as physics-based approaches or optimization techniques, have been more popularized to implement flocking behaviors (see e.g. [128] for a recent survey on the topic). Considering Reynolds' model [93] as the initial point, many authors have modeled the flocking behavior in artificial robots by means of physics-based approaches (see Section 2.4.1).

The coordinated motion of a group of robots using social potentials was addressed in [129]. Each agent was simultaneously influenced by repulsion from obstacles and attraction to goal coordinates, aggregated using a motor schemas approach. Moreover, they modeled flocking using a variety of possible geometric formations (e.g. line, or diamond). In [130], the authors performed decentralized control to model flocking and formation behaviors using Artificial Potential Functions (APF). They allowed flexible migratory coordinated movements with obstacle avoidance, based on the application of attraction and repulsion forces. The authors of [131] also addressed the emergence of flocking using APF, so that collective flocking movement, obstacle avoidance, and motion towards a defined goal position were simultaneously accomplished by the robot swarm. In order to compute the repulsion and attraction forces, the authors considered the three behavioral zones of [120], described above. Stranieri et al. proposed in [132] an approach to self-organized flocking in swarms of mobile robots based on artificial physics. A key aspect of their research lied on the behavioral heterogeneity of

their robot swarms, so that the swarm was split into two sub-groups with distinct behaviors. More precisely, the agents of one group, called *aligning robots*, were controlled using all the three Reynolds' rules of flocking, whereas the members other group, denoted as *non-aligning robots*, did not use the alignment rule. Their results revealed that, even though flocking is still possible with the behaviorally heterogeneous robot swarm, the overall performance was affected by the proportion of *aligning robots*. Several authors have used Active Elastic Sheet (AES) model to obtain the flocking motion [45, 46, 47]. The model proposed in [45] treated the robot swarm as a semi-rigid formation of particles joined by linear springs. In this way, they reached flocking without the explicit application of any alignment or velocity matching rule. In [46], the authors implemented collective goal-oriented motion behaviors with AES models in simple robots with limited sensing capacities. In line with [45], they showed that, with this setup, an autonomous flocking can emerge without using alignment rules. Bahaidarah et al. [47] also used AES models to tackle the flocking task. However, they used optimization techniques, based on Particle Swarm Optimization (PSO), on the viscoelastic links between robots, to mitigate undesirable swarm fluctuation, slow alignment, and excessive energy consumption.

In addition to the physics-based approaches, the use of optimization techniques to fine-tune the parameters of flocking models has been drastically popularized in the recent years. Specifically, algorithms from the fields of SI [70, 71], evolutionary computation [63, 62, 58, 60, 65, 64, 66], and RL [68, 69, 59] are the ones that stand out the most.

The authors of [70] used an noise-resistant extension of PSO to optimize the parameters of an ANN with one hidden layer. The ANN model received the range and bearing readings and the average heading orientation, and its outputs were used to directly control the two wheels of the robots. They compared two learning scenarios: (i) a homogeneous swarm with centralized learning, and (ii) a heterogeneous swarm with distributed learning. In both cases, Reynolds' rules were used to define the fitness scores that guided the learning process. They showed that flocking behaviors can successfully emerge in both scenarios. Additionally, [71] proposed a multi-objective pigeon-inspired optimization algorithm, improved with hierarchical learning, which was applied to a swarm of UAVs. The resulting optimized model allowed a coordinated motion with obstacle avoidance through complex environments.

In [63], the work in [102] was extended, obtaining self-organized flocking behaviors in a swarm of mobile robots using evolution strategies. They used the robotic platform *Kobot* [133], that was specifically created for SR research. The authors of [62] used evolutionary optimization to generate a tunable flocking model for real aerial robot swarms. Specifically, they used the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) to fine-tune the parameters that maximize flocking rules while minimizing the number of collisions. In [58], a GA was combined with a physics-based approach, with the aim of optimizing an AES model. Consequently, the GA directly found three parameters of the AES, so that virtual forces among robots were minimized and alignment was maximized. Dorigo and et al. addressed in [60] flocking as one of the tasks to test the *swarm-bot* system, a swarm of mobile robots (*sbots*) that can either attach to each other to give rise to more complex robotic structures. They used a standard EA that optimizes the parameters of a simple ANN in both an aggregation task and a coordinated motion task. In contrast to other reviewed studies, the authors of [65]

studied flocking behaviors in an ecosystem of plants, predators, and herbivores. More precisely, predators were fed on herbivores and herbivores fed on plants. Each agent (both predators and herbivores) were controlled by ANNs optimized using a GA. However, no explicit fitness function was given to the GA, so that evolution was purely driven by survival of the fittest and reproduction. The authors of [64] successfully evolved neural controllers for the emergence of flocking behaviors. They designed the fitness function of the EA based on the Reynolds' rules, namely, cohesion, separation and alignment terms. The robots had access to an alignment sensor, responsible for measuring their orientation relative to the average orientation of the neighbors. In [66] a flocking model was also evolved using genetic optimization. It was capable of preserving the swarm formation, the robot alignment and the avoidance of collisions with convex and non-convex obstacles.

RL has been also recurrently used to optimize flocking controllers. In [68], the authors used the Q-learning algorithm to obtain flocking controllers that operate in discrete time. Each agent perceives one neighbor i at each control cycle and, consequently, executes one discrete action among a set of four possible options (e.g. be either attracted or repelled by the perceived agent). They accomplished the learning process in multiple scenarios, with and without predators that should be avoided. The study in [69] applied a tabular Q-learning algorithm to a multi-agent system in order to achieve flocking behaviors. Each agent perceived its environment in the form of a normalized average velocity of its neighborhood. Such information was used to generate an action that either maintains the current direction of motion or performs a rotation movement. They assessed their system in two scenarios: (i) one with a learning agent assisted by a group of teachers (whose flocking controller is handcrafted), and (ii) a second scenario without teachers. The authors showed that, in both cases, flocking behaviors based on polar velocity alignment strategies emerged. Additionally, in [59] deep RL was applied on an AES model to obtain flocking controllers in a swarm of homogeneous robots. Specifically, the deep RL algorithm that they used was the Deep Deterministic Policy Gradient (DDPG) algorithm, which is a suitable option for continuous action spaces. Under the frame of their DDPG implementation, the authors employed an actor network that generates three parameters used in the AES model. However, they accomplished two experiments, considering *individual* and *collective* learning, respectively. Firstly, during the individual learning each agent was responsible for its own discovery of the parameters of the AES, so that multiple learning processes coexisted within the same robot swarm. Secondly, collective learning implied that the same parameters were shared by all the agents.

Foraging

Foraging [28] is a fundamental task in the SR field that takes direct inspiration from the foraging processes of ant colonies in nature [134, 31]. Its most basic description poses a scenario with a nest area, and one or multiple resource areas scattered across the environment. The aim of the task is that the swarm individuals, with no prior knowledge about the environment or the food location, must

(1) explore their surroundings and discover the location of the food sources,

(2) navigate towards the known food areas and transport the obtained resources back to the nest, and

(3) repeat (2) until there is no more food in the known sources, while some of the swarm agents continue to explore the surroundings for unknown and potentially richer food locations.

On top of its most basic definitions there are multiple extensions or concepts that have been included to the foraging task to make it more complex and realistic. Some recurring examples are battery consumption [135], physical transportation of the resources [136], or "*poisonous*" areas [81], among many others.

One of the most popular and well-consolidated approaches to tackle the foraging problem is *stigmergy*, which is the indirect communication through the environment. In the context of foraging in SR, stigmergy is commonly implemented using *artificial pheromones*, taking direct inspiration from chemical pheromones emitted by ants to communicate. It has been observed in nature that through the deposit of pheromones in the environment, ants can create chemical trails that connect to the richest food source through the shortest path [30, 31, 32].

Some authors have implemented artificial pheromones in real robots using emission and sensing of actual chemical compounds [137], or through its emulation [138, 139, 140]. In [137], a fully autonomous foraging based on chemical pheromones was proposed. The robots used ethanol to physically generate the pheromones, that can be perceived by other near robots using alcohol sensors. The agents deposit the chemical compound during its way to the nest, when a food source is detected.

Nonetheless, the emission and sensing of chemical substances in real robots is a challenging problem and, therefore, some studies have emulated chemical pheromones using other physical communication methods, such as light, heat, and odor. In [138], heat trails were considered as means to emulate the chemical pheromones. They used heater actuators and pyroelectric sensors for laying and tracking the pheromone trails. The authors of [139] emulated pheromones in a foraging task using graphics projected on the floor through a LC projector. They used a CCD camera, located above the arena, that can detect the positions of the robots, so that colored graphics can be projected accordingly. Robots can sense the color and brightness of graphics projected on their top. Additionally, Garnier et al. [140] implemented artificial pheromones using light trails projected on the environment. They assessed the system with ant-like robots that aimed at finding the best route between a nest area and a target area inside a network of corridors with multiple configurations (e.g. symmetrical and asymmetrical bifurcations).

The simplest, yet most used, method to implement artificial pheromones in robot swarms is using explicit exchanges of messages between robots (decentralized) or with a central computing node (centralized). Campo et al. [5] used virtual pheromones for the path selection between food sources and a central nest in a foraging task. Their experiments assumed that the chain formation that joins the nest with the food sources was already established, so that the focus of the study lied on the selection of a common route. For that, they introduced the concept of *virtual ant*, messages that are spread across the chain of robots that joins the nest with the food areas. Therefore, the density of virtual pheromones would be higher in the route that joins the nest with the closest food location. In order to amplify this process, the authors included a relaying mechanism that preferably transmits the virtual

ants towards the node with highest density of artificial pheromones, when a bifurcation in the route is produced. The authors of [6] addressed foraging using virtual pheromones in groups of three robots, both in simulation and real environments. The food sources to be foraged were Radio-Frequency IDentification (RFID) tags scattered in the arena. The robots' behavior was based on an agent-based model optimized using a GA. They used virtual pheromones to communicate others about known locations of RFID tags through a central server. Specifically, when a tag is found the robot deposits virtual pheromones during its return to the nest. In [54], foraging behaviors were obtained using virtual pheromones in large-scale robot swarms of up to 100 memoryless kilobots [23], making use of the Augmented Reality for Kilobots (ARK) framework [141]. The robots were controlled using FSM models, with (i) states such as random walk, turn back, go to depot, or follow pheromone trail, and (ii) state transitions with conditions such as detecting a pheromone, finding a source area, or being within the nest. Their kilobots were unable to engage in a direct communication with other agents and, furthermore, they could not even sense their presence. Thus, their only way to communicate and become aware of other neighbors was via the deposit and sensing of virtual pheromones. The virtual pheromones were completely handled by the ARK system. Robots notified to the ARK system when they want to deposit a new pheromone trail, and, conversely, the ARK system informed those kilobots with a pheromone in their current location. In [142], the authors used an ANN to model pheromone diffusion. The ANN was a one-layer perceptron connected in a lattice-like topology, so that each neuron uniquely represents a specific area in a two-dimensional arena. Thus, the activation of each neuron represented the density of pheromones in the corresponding zone of the environment. The input of each neuron was composed by the weighted average of output activities from "nearby" neurons, and from an external input current that depended on the agents' perception of the environment. Specifically, the agent could produce two types of virtual pheromones: (i) an attracting pheromone (positive injected current) when a food source is found, (ii) a repellent pheromone (negative injected current) when either obstacles or previously explored areas are encountered. Robots used indirect communication to gain holistic access to the virtual pheromone field.

Hecker and Moses proposed in [55] the Central-Place Foraging Algorithm (CPFA), for obtaining foraging behaviors with a unique central nest and multiple resource areas. CPFA is based on a FSM model with the following states (see Figure 2.5):

- *Set search location*. Robots are initialized inside the nest area, and, in order to start the foraging process, each one randomly generates a preferred direction θ to start searching for resources.
- *Travel to search site*. The robots navigate away from the nest towards the search direction θ. Transitions from this state lead to either *Search with uninformed walk* or to *Search with informed walk*. The former transitioning may occur at any time with a fixed probability. Alternatively, the latter transition is performed when one of the following conditions is met: (i) a pheromone trail is found, or (ii) with some probability the robot decides to return to some location where food was previously found (site fidelity).
- *Search with uninformed walk*. The agent is not aware of the food location and randomly patrols around the vicinity of some search area. The navigation is based on a correlated

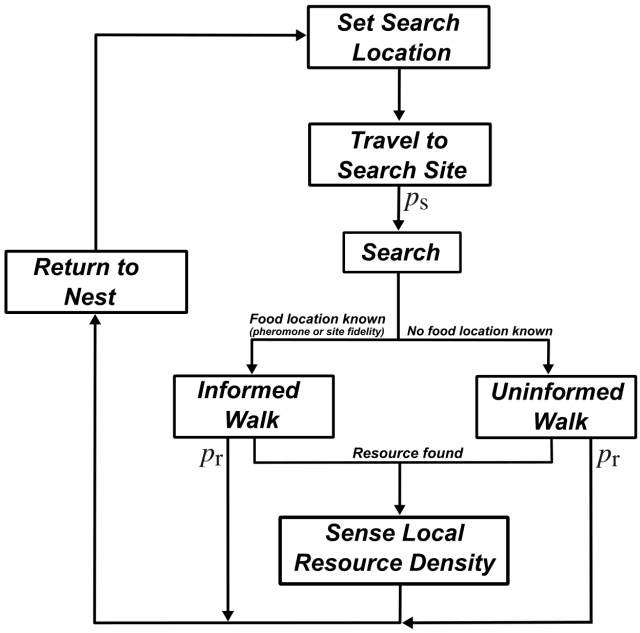


Figure 2.5: Diagram of the CPFA. The figure was adapted from [55].

random walk, so that Gaussian noise is dynamically added to the search direction at every control cycle ($\theta_t \sim \mathcal{N}(\theta_{t-1}, \sigma)$. At any time, with some predefined probability, the robot may abort its search mission and return to the nest.

- *Search with informed walk.* The robots in this state know the location of potential resources and perform an informed search. It is similar to the correlated random walk described above, but directed towards the estimated food location. At any time, with some predefined probability, the robot may abort its search mission and return to the nest.
- *Sense local resource density.* When a robot finds a resource the following actions are taken: (1) the food is collected, (2) it is added to a list of collected items (memory), and (3) the density of food in the current location is estimated.
- *Return to nest.* Finally, the robots that found resources return to the nest. The whole process is initiated again. During its route to the nest, the robot can deposit a pheromone indicating the original resource location. The emission of such pheromone is randomly determined using a Poisson cumulative distribution function.

The authors evolved the parameters of the CPFA using a GA with a population of 100 homogeneous robot swarms and 100 generations.

In [56], the Distributed Deterministic Spiral Search (DDSA) was proposed. It is a foraging algorithm with a central nest that implements a resource search strategy based on spiral trajectories. This spiral-based search guarantees a complete search of the whole arena, avoiding redundancy during the resource search process. Its performance, efficiency, robustness, and

scalability were assessed and compared with the CPFA. They showed that DDSA can solve the foraging problem (collect all the resources) in much less time than CPFA, mainly due to the stochastic search of the latter one. However, they also found some strengths of CPFA, such as its better performance for large robot swarms (more than 20 robots). The improved scalability properties of CPFA are associated to its better ability to deal with high congestion and accumulation of robots in the around nest. Lu et al. [143] also compared CPFA and DDSA, both in simulation and real robots. Surprisingly, they showed that DDSA performed better in simulations, but CPFA outperformed DDSA in experiments with real robots. The authors of [75, 76] proposed an extension of CPFA, called Multiple-Place Foraging Algorithm (MPFA), to mitigate its main limitations, namely, the high travel time and the collisions near the central place due to high accumulation of robots. In MPFA there are multiple nests scattered across the arena, so that robots return to the nearest nest when some resource is collected. Robots are uniformly initialized in the different nests for an efficient and balanced search. Moreover, they used a GA to optimize the parameters of the algorithm. MPFA was compared with CPFA, showing better overall performance, less number of collisions, and decreased forage time.

In contrast, several authors have applied Lévy search strategies, which has been used to explain and model foraging behaviors of some species in nature [144, 145], to robotic foraging. Lévy search is a stochastic search pattern that uses Lévy distributions to model the step size of the random walks. Consequently, smaller steps are more likely to occur, whereas the probability of larger steps decreases exponentially with the size of the step. Generally, the Lévy search itself defines only the size of the step, while the direction of the motion is randomly determined according to a uniform distribution in $[0, 2\pi)$. Lévy search strategies have been used in swarms of robot foragers with genetic algorithms to define some of its parameters [146], with Gaussian mixture models that incrementally learn the spatial distribution of the environment to sample resource locations [147], and combined with artificial pheromones and the Keller–Segel model [148].

Social odometry was proposed in [149, 150, 57] to address collective foraging and localization problems. Each robot in the swarm has its own estimate of its own position in the environment. Moreover, the robots also stored a confidence level associated to the position estimate. Such confidence level decreases as time and distance navigated grow, due to odometry errors. Therefore, *social odometry* is a distributed online learning algorithm that uses the estimates of other neighboring robots to reinforce their confidence levels, and improve their estimations about self and goal locations. Social odometry was applied to central-place foraging tasks with a single resource area [149, 150], and with multiple resource areas so that robots have to find the path to the closest one [57].

In addition to manually designed robot controllers, optimization algorithms have been also widely explored in foraging. Genetic algorithms have been used to evolve the parameters of the CPFA [55], the MPFA [75, 76], and Lévy search strategies [146]. Besides, Ferrante et al. [151] addressed foraging from a task partitioning perspective. Their experiments consisted of a group of mobile robots that had to collect resources and deposit them in a nest area, in both flat and sloped environments. Therefore, they proposed to partition the overall problem into a set of sub-tasks or primitives: (i) *dropper* robots that collect the resources and deposit

them in a temporary depot, and (ii) *collectors* that transport the stored items from the temporary depot to the central nest. They compared scenarios in which sub-tasks were either pre-defined or evolved from zero, using only very low-level behaviors (e.g. phototaxis and random walk). In the latter case, they used grammatical evolution to evolve grammar rules that define compositions of the basic primitives. Additionally, in [67] the authors used the NEAT algorithm to evolve the parameters and topology of the ANN that defined the foraging behavior of the robots. They compared the results using NEAT to the CPFA and the DDSA, showing equivalent performance of their evolved model.

Chapter 3

Emergence of Communication

3.1 Definitions

C. E. Shannon proposed in 1948 a mathematical formulation of a communication system [152]. Its model relied on three fundamental parts: the transmitter, the communication channel, and the receiver.

(1) *Transmission*: the transmitting entity (i.e. a person, an animal, or a machine) aims to forward some message with an associated meaning grounded on some information source (e.g. an external or internal event, or a previously received message). The message is converted into a signal and transmitted via some actuation mechanism, such as vocalization, gesturing, locomotion, or radiofrequency transmission.

(2) *Communication channel*: the signal in which the message is encoded is transmitted through a physical medium (e.g. air in the case of sound-based communication, or radiofrequency spectrum in the case of RF communication), and it is corrupted by some noise source associated to the communication channel. Here, one can consider a wide variety of sources of noise, depending on the communication system itself. Some examples are RF or sound interferences, distractions of different kinds if the agents involved in the communication are living beings, or distortions due to darkness or because of the observer's perspective in non-verbal communications.

(3) *Reception*: the receiver perceives the noisy version of the emitted signal, that has to be decoded into the correct message. Such decoding is ideally the inverse of the encoding message-to-signal mapping at the transmission side. The decoded message is fed to the destination (i.e. person, animal, or machine) who interprets and actuates accodingly.

In this PhD. Thesis, this formulation is adapted to the diagram shown in Figure 3.1A, where the organisms involved in the communication are the *speaker* and *hearer*, and the communication channel is more broadly refered as *environment*. The environment not only encompasses the communication medium, but it is also where agents perceive external events (such as threats or resources), and materialize their actions as *behaviors*. Moreover, agents perform the encoding and decoding of messages and signals through a *cognition* module (e.g.

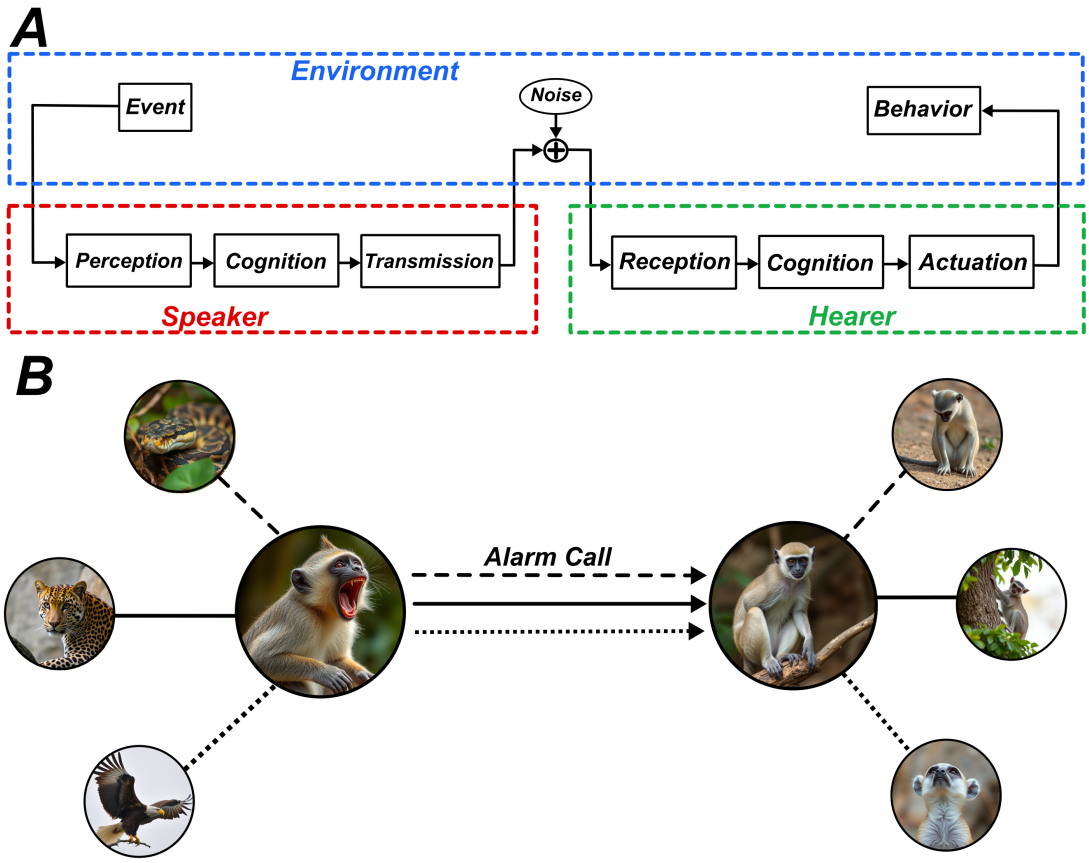


Figure 3.1: Basic definition of a communication system. (**A**) Generic diagram of a communication system between embodied and situated agents. (**B**) Diagram of the communication mechanics of vervet monkeys. All the images of animals were generated using AI from https://deepai.org/machine-learning-model/text2img

a biological or artificial neural network), which may also encompass working memory, decision making, or reasoning.

Oliphant also provided a basic formalism of a communication system in [153, 154]. There, the mapping between a meaning μ and a signal σ is modeled as a probability function $s(\mu, \sigma)$. Similary, the decoding of some signal σ into a meaning μ is modeled as the probability function $r(\sigma, \mu)$. Therefore, encoding and decoding probabilities can be arranged as probability matrices S and R, that map M meanings onto N signals, and viceversa (see Equations 3.1a and 3.1b).

$$\mathbf{S} = \begin{pmatrix} s(\mu_1, \sigma_1) & \cdots & s(\mu_M, \sigma_1) \\ \vdots & \ddots & \vdots \\ s(\mu_1, \sigma_N) & \cdots & s(\mu_M, \sigma_N) \end{pmatrix} \quad (3.1a) \qquad \mathbf{R} = \begin{pmatrix} r(\sigma_1, \mu_1) & \cdots & r(\sigma_1, \mu_M) \\ \vdots & \ddots & \vdots \\ r(\sigma_N, \mu_1) & \cdots & r(\sigma_N, \mu_M) \end{pmatrix} \quad (3.1b)$$

A representative example of a communication system in nature is presented in societies

of vervet monkeys, that warn other individuals about the presence of predators through alarm calls. Moreover, such calls differ depending on the type of predator sighted, and other listening monkeys react with differentiated response behaviors accordingly [35]. A simplified representation of this animal communication system is depicted in Figure 3.1B, where vervet monkeys perceive three different threats (leopards, snakes, and eagles), produce three alarm calls (signals) specific to each predator, and accordingly respond with three different behaviors (climb, look down, or look up).

Under the frame of communication systems in nature, Wilson provided in [38] the following general definition of biological communication:

> "Biological communication is the action on the part of one organism (or cell) that alters the probability pattern of behavior in another organism (or cell) in a fashion adaptive to either one or both of the participants. By adaptive I mean that the signaling, or the response, or both, have been genetically programmed to some extent by natural selection. Communication is neither the signal by itself nor the response; it is instead the relation between the two." ([38], p. 253)

Additionally, some authors have distinguished three types of social interactions depending on the purpose of the communication and the agent or agents that benefit from such interaction [155, 153]:

- *Exploitation*: the *hearer* exploits some signal or behavior produced by some *unconscious speaker*, so that the listening organism is the only individual that benefits from the interaction.
- *Manipulation*: the *speaker* intentionally generates some signal or behavior that triggers some desired response or behavior in an *unconscious hearer*. The *speaker* is the only individual that benefits from the interaction through manipulation.
- *Communication*: "true" communication is produced when both agents involved in the communication, the *speaker* and the *hearer*, knowlingly benefit from the interaction.

With this taxonomy in mind, Oliphant[153] defined communication as follows:

> "An act of communication is a causal chain of events, whereby one individual, the sender, exhibits a behavior in response to a particular situation, and a second individual, the receiver, responds to this behavior. Such an interaction is communicative if it involves manipulation on the part of the sender and exploitation on the part of the receiver." ([153] p. 14).

In the context of SR and AI, communication systems in societies of artificial organisms are divided into three categories [156]:

- *Indirect communication*: is characterized by a communication through physical contact, via the sensor readings [77, 157, 158, 64].
- *Stigmergy*: denotes indirect communication that uses the environment as the communication medium [159, 140, 55].
- *Direct communication*: requires explicit transmission and processing of signals and

symbols whose meaning is correlated to the specific task. There are multiple works in the literature that employ direct communication to solve a wide variety of SR tasks, with diverse communication technologies and completely different emergent or designed communication semantics (see, e.g., [50, 8, 7, 9, 10, 25]).

3.2 From signaling to language

What are the main differences between signaling-based interactions, common in most animal communications, and complex languages like human language? Before diving into the more intricate language-based communications, the most basic form of communication, namely, *signaling*, is defined. A signaling-based communication is understood in this PhD Thesis as an interaction between one *signaler* (or *speaker*) and one or more *receivers* (or *hearers*), so that the *signaler* notifies the *receivers* about some event, or condition ocurring in the environment where all of them are embodied and situated. Such transmitted signal must, moreover, trigger some response or behavior on the individuals listening. Broadly speaking, a signaling communication can be manipulative, explotative, or both.

In contrast, a language is a much more complex an elaborated symbolic communication process that is characterized by the following characteristics that signaling systems lack [15, 16]:

- *Symbolic communication and compositionality*: language is based on symbols, commonly refered as words or morphemes, that are collected as a *lexicon* or *vocabulary*. Moreover, symbols can be manipulated and combined through rules, i.e. grammar, to give rise to other symbols with different or modified meanings. As a part of grammar, words are divided into *syntactic categories* (e.g. nouns and verbs) that, when combined, give rise to properties such as predication, deixis, or referential communication.
- *Duality of patterning*: meaningful symbols (e.g. words) are composed by other lower-level units (e.g. phonemes or syllables) that lack of meaning by their own.
- *Arbitrariness*: the relation between the words in a language and the external referent on which they are grounded is arbitrary. This implies that words do not resemble or correlate with the entity that they represent.
- *Cultural transmission*: even though there can be a strong evolutionary component that gives rise to language pre-adaptations; language, lexicons, and grammar rules are culturally transmitted among language users.
- *Displaced communication*: language can be used to refer to entities or events that are currently not observable (neither in time nor in space). That is, it is possible to speak about an elephant even though no elephant is visible by the speaker at that moment (and in some cases the speaker has never seen an elephant in the past).

Language-based communications should include all or most of these complex features in order to be clearly differentiated from lower-level forms of communication, such as signaling. Some communication systems in the animal realm present some of these properties in isolation, and, however, we clearly classify them as signaling communications that are utterly simpler and less intricate than human language. Two interesting examples are the waggle dance

of honey bees and the alarm calls of vervet monkeys (see Section 2.3). In the former case, the dance-based signaling of honey bees exhibits some features of displaced communication, because workers signal infomation about distant resources that are not observable from the hive, neither temporally nor spatially. However, the communicated signals of the waggle dance are not arbitrary because the signal explicitly encodes the meaning being communicated. Specifically, the distance and direction from the hive to the resource are encoded in the length and orientation of the straight run of the waggle dance. In contrast, the mapping of meanings and signals of the vervet monkeys can be conceived as an arbitrary relation. That is, the alarm call is not correlated to the sighted predator. Nonetheless, the signaling of vervet monkeys is not a displaced communication (it cannot express "There was a snake yesterday"), and it is mostly innately defined.

3.3 Origins and conditions for the emergence of communication

M. H. Christiansen and S. Kirby suggest and categorize the origins of language and communication as one of the hardest problems of science [160]. It is an unsolved and open problem that requires the interdisciplinary collaboration from many fields of knowledge [161, 16], such as linguists, cognitive scientists, roboticists, or biologists, among others. However, in order to unveil the origins of human language it is crucial to focus firstly on the origins of the most basic forms of communication, i.e. *signaling*.

In [162], it is distinguished between *evolution of communication*, related to simple and usually signaling-based communication systems in animals, and *evolution of language*, understood as human-like language. Moreover, they state that "the study of language is inherently connected to the evolution of animal communication". The focus of this subsection lies on the evolution of animal-like signaling communications, as the basis for Chapter 3.5 where the pillars of language are revisited.

Phylogenetic problem of communication

One of the main issues and open questions when studying the emergence and evolutionary origins of signaling (and communication in general) is the *phylogenetic problem of communication* (see [163]). Essentially, it states that the emergence of communication requires the co-evolution of both signal emission by an agent and its corresponding response (in terms of behavior or communication) of listener agents. This *paradox* has been mentioned in many studies about the origins of communication (see e.g. [77, 164, 81, 163, 2]) and can be summarized by the words of Maynard Smith in [165]:

> "It's no good making a signal unless it is understood, and a signal will not be understood the first time it is made" ([165] p. 208).

Nonetheless, from an evolutionary point of view, it is utterly unlikely that simultaneous mutations give rise to both speaker and hearer roles at the same time. And, furthermore, if either speaker or hearer roles are evolved first, it is also improbable that these traits persist long enough through the evolution history. The reasoning behind this statement is that

the evolution of only one of the communication roles would not result in a fitness increase or higher survival chances until the complementary skill is present too. As explained in [77], there are two possible hypotheses to solve this *paradox*:

> (**1**) some agents tend to "*unintentionally*" produce signals collaterally to their non-communicative behaviors. The collateral signal can be the behavior itself, that is potentially observed by other individuals, or some dedicated signal generated through common neural activity patterns. In any of these scenarios, other observers are evolved with the ability to interpret and react to the pre-existing signals.
>
> (**2**) some individuals develop the tendency to react to many incoming stimuli from the environment and from other organisms. Therefore, evolution can use these arbitrary and initially non-communicative responses to elaborate proper signals that elicit the expected behaviors on the reception side.

Similarly, Mirolli and Parisi define in [163] two different types of emerging communications:

- *Exploitation of a receiver bias*: some individuals are sensitive and responsive to certain environmental stimuli that causes behavioral effects. Thus, other organisms exploit this reaction to *"consciously"* emit a similar signal that evokes the desired behavior of the listener. In other words, the communication is based on *manipulation* strategies, where only the speaker benefits from the communication.
- *Exploitation of a producer bias*: some individuals develop and repeat behaviors that *"unintentionally"* correlate with some information or event of the environment. Consequently, observers exploit this bias and interpret it as a communicative signal. As it was discussed in [163] and reviewed further in this section, this kind of exploitation can explain the emergence of *altruistic* communications in biological and artificial societies, where only the hearers benefit from the communication.

Ritualization

There is a general consensus about the origins of basic communication, suggesting that it may have evolved on the basis of pre-existing non-communicative behaviors. This statement has been observed and studied under the frame of both animal communication [166, 167] and simulated artificial systems [77, 82]. In [38], this *semanticization* process in evolution is called *ritualization*. Wilson defines the concept of *ritualization* as follows:

> "(...) ritualization, the evolutionary process by which a behavior pattern changes to become increasingly effective as a signal. Commonly and perhaps invariably, the process begins when some movement, anatomical feature or physiological trait that is functional in quite another context acquires a secondary value as a signal" ([38], p. 317).

Ritualization has been used to explain the origins of multiple communicative behaviors in nature. For instance, Scott et al. showed in [166] that the vibratory signals that are generated by caterpillars to communicate their ownership of some territory may have originated from previous locomotion behaviors. Ritualization is also a plausible hypothesis to explain the evolutionary origins of waggle dance communication in honey bees [168], which resembles

a miniaturized version of their actual foraging, and *stigmergic* communication in ants via pheromones [167].

Altruism

Another important issue that arises is the emergence of *altruistic* communication. A communication system is said to be *altruistic* if only one of the communicating individuals, the *hearer*, is benefited from the interaction. Such altruistic communication is not only observed in humans, but also in animal communication systems such as the alarm calls of vervet monkeys to warn about the presence and type of predators to other monkeys, at the cost of exposing the caller [35]. How is it possible that *altruistic* signaling traits are preserved and developed when the signaler has no direct increase of fitness, survival chances, or reproduction success? One possible explanation is, indeed, ritualization, because the *listener* could potentially be exploiting some non-communicative behavior of an *unconscious speaker* (exploitation of a producer bias). However, this phenomenum is only plausible either in very simplistic signaling communications, and only to explain the origins of the *altruistic* communication (not its preservation). Recalling the example of the vervet monkeys, the alarm calls not only warn about the presence of predators, but also encode the type of predator detected (e.g. pythons, leopards, and eagles), so that listeners can react accordingly [35]. Such *altruistic* signaling may have originated through ritualization, but it has clearly not evolved into an *unconscious* call. Another powerful explanation is the *kin relatedness* between speaker and hearer, so that both individuals involved in the communication are genetically similar. The reasoning behind this hypothesis is that both speaker's call and hearer's response traits are enconded in the same genome, which is shared in both communicating individuals due to their kinship. The importance of kin relatedness to explain the origins of communication has been widely explored using computer simulations (see e.g. [169, 80, 170, 164, 81]).

Artificial life and computer simulations

Studying the evolutionary origins of communication is an utterly complex task due to the large amount of theories and scarce, or even non-existent, empirical basis [162]. More importantly, the evolutionary and cultural origins of a communication system is a non-repeated event that cannot be observed directly [171]. In order to figure out the origins and evolutionary trajectories of communication, researchers can barely rely on archaeological evidence and fossil records [162, 172], and on other species that share a common ancestor in the evolution history. Computer simulations and the evolution of artificial life is a relatively novel tool that can bring some light to the study of the origins of communication.

Artificial life presents many limitations, and it cannot be used to demonstrate and unify the origins of animal communicaiton or human language. However, it can be used to partly assess certain hypothesis and ideas in a virtual scenario with a fully customized and controllable artificial evolution. In other words, it can be used to complement and reinforce other disciplines devoted to language evolution (e.g. phylogenetics, linguistics, biology, or paleoanthropology). In [161], the authors collected three main roles and use cases of computational models and simulations in the field of evolution of language and communication:

- *Evaluation*: due to its rigurosity, computational models and simulations can be used to assess whether or not some event or condition is the cause of some phenomenum (in this case the emergence of communication), at least in the simulated world.
- *Exploration*: computer simulations can guide researchers to elaborate new theories and hypotheses.
- *Exemplification*: computational simulations can be also used to explain more easily some aspects of the evolution of communication.

M. H. Christiansen and S. Kirby also highlight in [161] that computational simulations should not be used in isolation. They should be used jointly with theoretical considerations, mathematical modeling, experimentation and data collection.

3.4 Emergence of signaling in swarm robotics

The emergence of low-level signaling communications in societies of robots is mostly addressed from the paradigm of evolutionary robotics. That is, communication semantics emerges as a result of an evolutionary process. Before delving into the extensive literature review, it is important to define the following types of evolved communications, depending on the emerged semantics [173]:

- *Abstract communication*: only the abstract message transmitted between the agents is communicative. That is, all the semantics lie on the message content, and it can be understood and processed in the abscence of its context within the environment.
- *Situated communication*: both the message and its context and situatedness within the environment are communicative. The message content cannot be understood without the associated environmental context. A great example of situated semantics would be the *stigmergic communication* through pheromones, because the pheromone trail cannot be interpreted without the location where it was deposited in the environment.

In [77], Quinn studied the origins of communication from a group of initially non-communicative agents. He explored the hypothesis that communication emerges during the evolution process from previous non-communicative behaviors. Moreover, he stated that to properly study the origins and evolutionary paths of communication using artificial life simulations, dedicated communication channels should be discarded. His main arguments were that dedicated communication channels are specific and functionally isolated to send and receive communication signals, that are only potentially useful in a strictly communicative context. The sole act of designing dedicated communication channels and deciding the potential signals or symbols that can be transmitted is an indirect imposition to communication. Furthermore, the use of dedicated communication channels precludes the incremental evolution of communication from non-communicative behaviors. Quinn proposed a proof-of-concept experiment to demonstrate that communication can be successfully evolved without using dedicated channels. The featured task was a collective navigation in which two mobile robots had to move away from their initial positions while keeping in sight (staying within sensor range). Each experiment simulation lasted 10 seconds and the initial positions and orientations of the robots were

selected randomly among a set of possible configurations. In order to succeed in the task, some sort of communication was required between the pair of robots. However, his robots did not have access to dedicated channels, so that they could only communicate through motion patters via motor wheel control and using eight IR sensors. Quinn's results revealed that it is possible to evolve communication without dedicated and functionally isolated communication channels. The evolved behavior was the following: (1) An agent rotates counter-clockwise until it faces the partner. (2) Once both agents face each other, one of them moves away (backwards) from the other, resulting in the latter moving towards the former (forward) as a consequence. (3) This coordinated back and forth "*dancing*" is repeated until the experiment concludes. Notice that not only the robots communicate in the form of a non-trivial "*dancing*" that was evolved from zero, but also they autonomously allocate the *leader* and *follower* roles in the "*dancing*" without any bias or predisposition towards none of these roles. Moreover, the author also analyses the evolution history, showing an incremental evolution starting from very basic non-communicative behaviors, such as straight line navigation and collision avoidance, up to the previously described evolved role-based communication.

In [78] and [79], the authors investigated the origins and emergence of communication in artificial evolution in a collective navigation task. In their experiment, there were two goal ground areas and a team of four homogeneous robots, so that the aim of the agents was to aggregate in groups of equal size in the two areas (i.e. two robots in each target area). In contrast to [77], robots used dedicated channels to communicate with other neighbors. After the evolution of the robot neurocontrollers, in [79], the authors discovered that the emerged communication was an interesting signaling process with a *zoo* of non-trivial signals. They also verified that the signals properly triggered important modifications in the listener robots, not only in terms of motor behavior but in the communication itself. For instance, one of the evolved signals, *signal B*, was produced when a robot was located alone inside a goal area. In turn, the reception of *signal B* caused both a tendency to approach the source of the signal and the production of a second signal, *signal D*. The semantics of the signals were grounded both on the current states of the robot and its team mates, and on their situation within the environment. As an example, the emerged *signal C* caused different motor reactions depending on the location of the agent listening (i.e. inside or outside the target area). The work in [78] deepened more into the origins and evolutionary pathways of the emerged signaling. By exploring the different behaviors of the robots across generations, they found out that the evolution of communication can be characterized as a co-evolution of motor and communicative behaviors. In other words, motor behaviors are the basis to evolve new communicative and signaling skills and, in turn, the new communicative are used to produce novel and more complex motor actions. Therefore, they highlighted that the semantics of the evolved signals are not only grounded on sensorimotor signal, but also on previously acquired robot behaviors.

In an earlier work, Marocco et al. [80] studied the emergence and origins of communication in a completely different task. They proposed a collective categorization of objects of different shapes. Agents, modelled as three-link arms with six degrees of freedom, could sense the objects through touch sensors. The robots were controlled by ANNs whose parameters were evolved using a GA. Even though the experiments involved a single robot, the agents could communicate with other individuals of the GA population (e.g. with the parents) in order to

share information about the categorized objects and actions. They studied the emergence of communication in two separate experiments devoted to: (*i*) the emergence of basic signaling and signals grounded to nouns, and (*ii*) the evolution and emergence of verbs. In the first experiment, they assessed the emerged communication under the following specific conditions and constrains:

- `Parent_Speaker`: Communication is only possible with parents in the GA population.
- `All_Speaker`: Communication is possible with any individual in the GA population.
- `From_0`: Communication is available from the first generation of GA evolution.
- `From_500`: Communication is only available from generation 500 of GA evolution.

Comparing the results of the evolved solutions under conditions `Parent_Speaker` and `All_Speaker`, the authors found that communication between kin individuals (i.e. communication with parents) led to clear benefits and improvements in the emerged signaling. According to their findings, one of the main reasons for this cause was the use of stable and reliable signals in the communication between the two relatives. Furthermore, the sole action of transmitting a meaningful and useful signal to other agents did not imply a fitness increase. Thus, there was an *altruistic behavior* in the communication, up to some degree. According to the authors, this altruistic behavior can be owed to the kinship relationship and genetic similarity between the sender and the listener (see also [169]). They observed another important finding resulting from the comparison between `From_0` and `From_500` evolution results. Specifically, they concluded that the evolved communication is more efficient and superior when the communication is enabled from generation 500 (`From_500`). This finding suggests that the emergence of signaling communication is more likely after the evolution of more basic non-communicative behaviors, that pave the way for more complex and specific cognitive behaviors such as communication. These conclusions match with the previously described results observed in [77] and in [78] (posterior work).

The authors of [81] studied the conditions for the evolution and emergence of communication and signaling in populations of initially naive and non-communicative robots. They proposed a foraging task with one food area and one poison area, both of them emitting a red light. Thus, even though the robots were able to perceive the areas from the distance, they could only distinguish the type of area in the short range. Robots could potentially emit blue light to communicate with other robots, and their behavior was determined by a feed-forward neural network subject to artificial evolution. Their results revealed two main conditions for the emergence of a beneficial and cooperating communication:

- *Kin structure*: the robots are genetically similar and related,
- *Level of selection*: selection during the evolutionary process applied at the colony level.

They repeated the evolution of communication in scenarios that fulfill the previous conditions in 20 different trials. Beneficial signaling emerged in all of the experiments. Nonetheless, they observed two types of signal semantics: (i) senders signal the presence of nearby food and listeners react by approaching the food area (in 12 out of 20 experiments), and (ii) senders signal the presence of nearby poison and listeners react by escaping from the poison area (in

8 out of 20 experiments). In the experiments where any of these conditions were not met, the emerged communication was deceptive (e.g. the robots emitted signals when they are far away from both food and poison zones). Moreover, they surprisingly found that listeners navigated towards the source of the signals instead of ignoring or avoiding them. Another finding was that once an stable signaling communication emerges it becomes stagnant, with no evolutionary improvements. They hypothesized that such stagnation was because any alteration of the signaling, meaning or motor response, would result in a noticeable fitness decrease before finding another fitness maximum with a more complex communication system (similar to the analogy of *"a needle in a haystack"* posed in [174]).

Mirolli and Parisi also studied in [170] the emergence of altruistic communication and the importance of kin relatedness of the agents. They proposed a task with simulated agents in which communication only favours the hearer, bringing no benefit to the signal producer. The robots were controlled by ANNs evolved using a GA, and the environment where the experimients took place was a corridor with discrete positions (tiles or cells). In each simulation, the *speaker* was positioned on the leftmost cell and the *hearer* was located on the right side of the corridor. Moreover, a "*mushroom*", that can be either edible or poisonous, is instantiated nearby the speaker. Agents can only perceive mushrooms that are located in their immediate surroundings (within one cell), but they can communicate with the other agent in the long range. More precisely, the speaker, that is initially close to the mushroom can potentially signal its type (edible or poisonous) to the distant hearer. However, *eating* a mushroom implies benefits or consequences depending on its edibility: eating an edible mushroom increases the individual's fitness score, whereas a poisonous mushroom implies a decrease of the fitness value. Therefore, signaling the type of mushroom to the other robot is an *altruist* act because only the hearers benefits from the communication. The results revealed that efficient and stable altruist communication could not be evolved successfully. They found that unstable evolution of communication was caused by the existence of *mutant* speakers, that generated bad signals that misled the hearer agents. The authors overcame this problem using two important hypotheses of the emergence of *altruistic* communication: (1) kin relatedness between the communicating agents, and (2) cultural transmission. For the former case, they included an additional parameter that determines the probability that both speaker and hearer are genetically similar. The results in terms population average fitness and language quality revealed that, indeed, kin relatedness increases the success of *altruistic* communications. The authors also assessed the importance of cultural transmission, exploring the concept of *docility* [175]. In this context, it refers to the willingness of an individual (mostly infant) to learn the communication semantics and mechanics from others, through cultural evolution. In order to accomplish this experiment, Mirolli and Parisi completely reduced the genotype of the agents to a single gene that encodes the individual's *docility*. Weights of the ANN controllers were randomly initialized at the beginning of each simulation, and were culturally optimized through a back propagation algorithm using the agents' parents as teachers. The *docility* value represented the number of learning loops in which each agent engages during its infancy period.

In [1], a sound signaling was evolved in a task that does not necessarily require communication to be solved. The task was a categorization task in which mobile robots were requested to discriminate between two possible environments, and act accordingly. The environment was

composed by a light source surrounded by 3 different circular crown areas of different colors. The colors of the circular crowns corresponded to different tones of grey, being the innermost area colored in black. The black circular crown was a *danger* area that cannot be crossed by a robot, while the grey tone of the outermost areas indicated to the agents proximity to the *danger zone*.

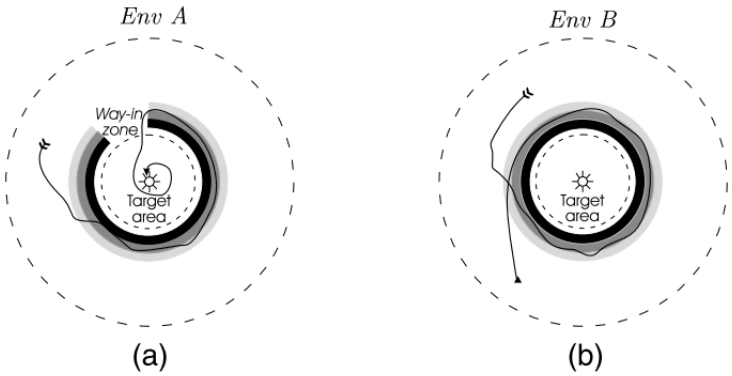


Figure 3.2: Environments in the categorization task in [1]. (a) *Env A*: the target area can be accessed through a *way-in zone* safe area. (b) *Env B*: the target area cannot be accessed by the agents. The source of the figure is [1].

The two possible environments, namely, *Env A* and *Env B*, are depicted in Figure 3.2. *Env A* presents an entrance to the target light source without *danger* area, referred by the authors as *way-in zone*. Alternatively, the danger circular crown in *Env B* completely surrounds the target area, so that robots can by no means access the light source. The aim of the task was that robots seek the light source in *Env A* while avoiding it in *Env B*. After the evolution of the CTRNN controllers, the results disclosed that the robots were able to correctly solve the task in 13 out of 20 evolutionary runs. Moreover, the authors also observed that, among those 13 successful evolutions, sound signaling emerged in 9 runs. They verified that signaling is more relevant and notable when *Env B* is shown to the agents, whereas it is almost negligible in *Env A*. Furthermore, when signaling did emerge it produced clear effects and alterations on the behavior of the robot listening, and an increse in the overall group performance. Specifically, they found out that the perception of a sound signal led to a switch from *phototaxis* to *antiphototaxis* behaviors. In accordance with previous works studying the origins of communication [77, 80, 78], in [1] it was also found that the most basic forms of communication, i.e. signaling, seem to arise after the acquisition of non-communicative behaviors. In their experiment, Ampatzis et al. observed that the first emitted signals in the evolution history carried no apparent meaning and produced no effect at the reception side. Actual communication, understood as the combination of an emitted signal and its behavioral or communicative response, started to emerge only after the first individual and non-communicative categorization behaviors.

E. Tuci studied in [2] the *phylogenetic problem of communication* (see Section 3.3) in a categorization task that requires the communication between two robots. The robots were endowed with a simple sound-based communication system, although their initial behavior was

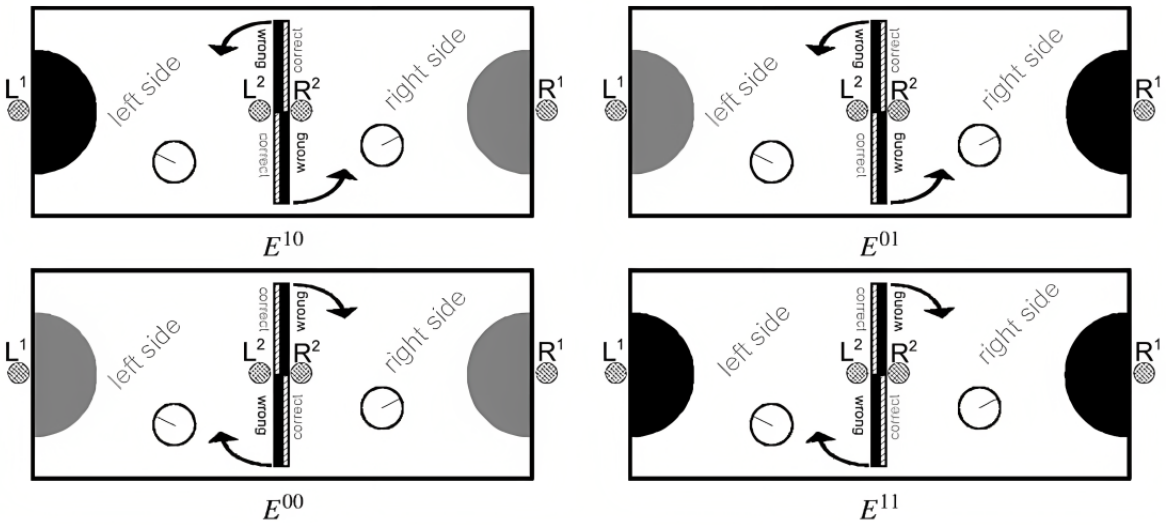


Figure 3.3: Environments considered in [2] for the emergence of reciprocal communication. The source of the figure is [2].

non-communicative. It was an ER algorithm the building block that guided a CTRNN-based robot controller towards effecive communicative behaviors. The author proposed the following task (see Figure 3.3 for a graphical illustration of the task setup): Two robots were placed on each side of a rectangular arena constrained by walls. The arena was divided into two equiareal portions by a revolving door, that can be rotated either clockwise or counter clockwise by the robots to cross to the opposite side. Notheless, the door can only be rotated if both agents simulatenously push it in opposite directions from each arena side. A ground area is located in the outermost section of each side of the arena. Tuci establishes four different variations of the environment (E^{00}, E^{01}, E^{10} and E^{11}), depending on the colors (gray or black) of the ground areas. Moreover, each specific environment dictates the forces that the robots should apply to the door in order to rotate it (see Figure 3.3). Each robot only knows the color of one of the ground areas and, therefore, some sort of communication is compulsory to categorize the type of environment. The results of this study revealed that rewarding the categorization capabilities of the robots in the fitness function had a possitive impact on the evolution of signaling. Specifically, it was relevant for preserving early and simple non-communicative behaviors related to categorization, that eventually could give rise to effective reciprocal signaling.

Wischmann et al. studied in [176] how historical contingency can notably affect the evolution of different signaling communications under the same selective pressure. They addressed this problem using simulated robots, controlled by neural controllers, devoted to solve a foraging task. Specifically, the agents must discover and aggregate inside an initially unknown food area. Robots could potentially emit blue and green lights, so that communication can emerge either across one or two communication channels. They repeated the evolution process in 20 independent runs and, surprisingly, they showed that two different kinds of signaling behaviors emerged. In some cases, the robots only used one of the colors to notify the presence of food, whereas in the other evolutionary runs the robots exploited both signals (green

and blue) in their communication process. When comparing both emerged communications, they authors discovered that the (less complex) signaling that uses only one color leads to higher performance than its more elaborate counterpart. They also highlighted that the first important traits characterizing each signaling, and leading to its emergence, occurred at the early stages of the artificial evolution, having a high impact on the whole evolutionary path traced. They reasoned that the emergence of these different signaling systems under the same environment and selective pressure is due to historical contingencies. That is, the evolutionary divergences are caused by the stochasticity in the evolutionary process and due to the randomness of historical events and genetic operators. Moreover, the authors also pointed out that, once emerged, the signaling becomes stable and stagnant, unable to optimize to other more refined and better performing communication systems. In a second experiment in [176], the impact of historical contingency in competitive skills was studied. Instead of setting a fully cooperative group of foragers, as in the first experiment, two groups of robots from different populations competed for accessing limited food resources. In contrast to the first experiment, they showed that the group whose communication was based on the two colors performed better. They attributed this performance swap to the high robustness of the more complex signal.

The authors of [82] explore the emergence of displaced communication in an abstract foraging task. Displaced communication refers to communication systems and signaling processes where the subject of the communication is not immediately present or visible. Displaced communication not only is always part of human communication but it has also been observed in animals (being the honeybee waggle dance one of its main exponents). In order to study this complex type of communication, Bernard et al. proposed in [82] a simple abstract foraging task with multiple food areas an a nest located along a circumference (world is one-dimensional). The agents can visit the different areas by moving along the circumference, and food is only present in one of the food areas simultaneously, so that agents can use communication to notify others about the location of food. Nonetheless, owing to the fact that robots can only communicate inside the nest, some sort of displaced communication is required for an effective signaling. The results of the evolution revealed that the information and semantics of the communication were not modulated by the amplitude of the signals. Instead, the information about the food location lied on either the timing of onset-delay, the duration of the signal, or both. Amplitude modulation did also emerge in some evolutionary runs in which encoding via onset-delay and duration was prohibited. Surprisingly, they experimentally verified that onset-delay and duration encoding did emerge over amplitude encoding regardless of the fact that the latter resulted in a better performance. Their confirmed hypothesis was that signaling encoding food location via amplitude of the signal was much more difficult and takes more generations to evolve. Bernard et al. argue about the findings of [77], stating that evolutionary origins of communication should be effectively studied with the use of dedicated and functionally isolated channels. They explain the evolutionary predilection of onset-delay and duration encondings because amplitude encoding cannot be easily evolved from previous a priori non-communicative behaviors. Instead, the signaling via duration or onset-delay can be incrementally evolved from previously acquired behaviors. For instance, the distance to the target correlates with the time taken by an agent to return to the nest from the correct food source. This behavior, that was not initially interpreted and communicative,

did evolve into a transmitted signal and its corresponding behavior response at the reception side. In general, they associated this incremental evolution of signaling with the concept of *ritualization* [177, 155, 178], in which previously established behaviors in an organism lose or extend its functionality to develop new communicative skills. Besides, another finding in [82] that matches with the observations of [81] and [176] is that once some sort of effective communication system has emerged, it is highly improbable that evolution is guided towards a novel and improved communication. As Floreano et al. [81] also hypothesised, this is likely caused by the immediate abrupt decrease in fitness that results from leaving one communication system to develop a new one.

3.5 Emergence of Language

Human language is the greatest exponent of an utterly complex emergent communication system that exhibits intricate properties such as compositionality, arbitrariness, recusiveness, or displaced communication, among many others [15, 16]. Most of these properties have not been clearly observed in any other communication mechanism in the animal realm. The transition from signaling-based communication mechanics to symbolic and language-like communication is a complex mystery yet to be solved and understood. A traditional line of research and current of thought, which greatest exponent being Noam Chomsky, proposes that language is mostly innate and acquired through biological evolution [179, 180]. This claim is motivated by the *poverty of stimulus* and the fast learning of new languages by children. Chomsky's nativism suggests that there is a cognitive module, genetically determined and transmitted, called the Language Acquisition Device (LAD) [180]. The LAD encodes the set of invariant and universal language rules, known as *Universal Grammar*. Therefore, children's language learning starts from this innate initial state and it is further specialized to the taught language in a process of *parameter setting*. Additionally, Fodor's *Language of Thought* [181] proposes a distinction between the natural language used to communicate and the mental language, or *Mentalese*, where thinking and cognition takes place. Words, sentences, and, compositional semantics of natural language are associated with an equivalent in the *mentalese*. In line with Chomsky's nativism, Fodor argues that the language of thought is innate and genetically determined.

The nativist thought school drastically diverges from the behaviorist theory of language acquisition, being B. F. Skinner and his book *Verbal Behavior* [182] the most important representative. The behaviorist theory claims that the language acquisition is mainly produced through a process of learning, reinforcement, and imitation, rather than being genetically determined. Nonetheless, Skinner's work was heavily criticized by Chomsky in [183].

More recently, several authors have proposed an intermediate hypothesis, stating that the emergence, origins and acquisition of language derives from three interacting and interfering processes, namely, (*i*) individual learning, (*ii*) biological evolution, (*iii*) cultural transmission [184, 161, 185]. These three dynamic processes and their main interactions, depicted in Figure 3.4, are the following:

- *Biological evolution* (phylogeny) conditions the individual's cognitive and linguistic capabilities. Evolution imposes a learning bias that affects the language acquisition and

comprehension of individuals.

- *Individual learning* (ontogeny) involves the acquisition and comprehension of language at the agent's lifetime scale. Learning is conditioned by the individual's environment and the cognitive resources genetically determined by the biological evolution. The learning causes a great impact on language transmission to new generations and on the language change through cultural dynamics.
- *Cultural transmission* (glossogeny) refers to the propagation, cultural evolution, and language change across generations (at a historical time scale). Cultural transmission in turn affects biological evolution through an indirect fitness increase and alteration of the fitness landscape.

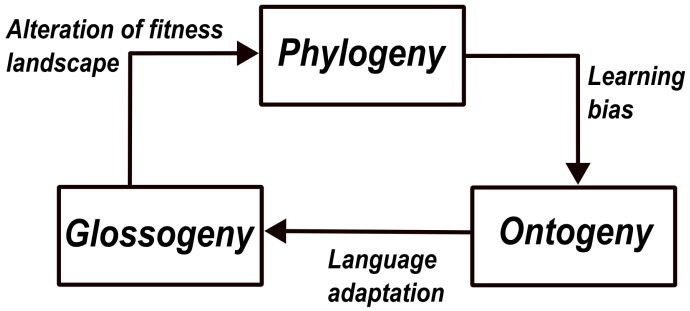


Figure 3.4: Processes involved in the emergence of language. Figure adapted from [184].

According to this decomposition, the three processes occur at different time scales and affect each other in a closed loop. Therefore, it seems reasonable to believe that all of them are of essence to fully unveil the origins of language. In contrast to the previous section that reviewed the artificial evolution of communication, this section is focused mainly on the cultural substrate of language emergence.

3.5.1 The Symbol Grounding Problem

Cognition, human thinking and intelligent systems have been traditionally hypothesized to be based on symbols, composition of symbols, and rules to manipulate those symbols [181, 186]. However, symbols, instead of being purely abstract representations, must be somehow linked to entities, events, or experiences about the real world. Far from being trivial, this requisite is know as the *Symbol Grounding Problem* (SGP) [13], usually considered as one of the fundamental problems in AI, robotics and, more abstractly, philosophy. The SGP poses the question on how symbols are grounded on physical and external referents in the world, and how they acquire meaning. Harnad distinguishes three main phases that should be satisfied in the grounding of symbols (see [13, 14]):

- *Iconisation*: or *categorization* is the process of transforming analog sensory signals, which are projections of the external world perceived by the agents via a sensory apparatus, into *iconic representations*. These *icons* are sub-symbolic representations that generally resemble sensor stimuli (belonging to the same input space).

- *Discrimination*: is a process characterized by the ability to determine whether two input signals are the same or not, and, in the latter case, how different the signals are. Discrimination is highly related to the iconisation process, because the formation of solid iconic representations, that reflect invariant and distinctive properties of the external object on which they are grounded, is of essence for an efficient discrimination.
- *Identification*: is a complex process that implies the assignment of a symbol or *meaning* that uniquely abstracts a given iconic representation. Discrimination and identification should not be understood as the same operation, as the outcome of the former are categories that are non-symbolic representations. In the words of Harnad, "I could be discriminating things without knowing what they were" [13]. Besides, as pointed out in [187], there is a major distinction between an icon and a symbol. Specifically, an icon resembles and looks like the object it represents, whereas a symbol shares no direct or indirect relation with such object. The *meanings* resulting from the identification should be symbolic and agents should be able to operate with them systematically. Moreover, the associations among iconic representations and meanings are commonly considered as an *arbitrary* mapping [13, 14], which is learnt from experience or by convention. Even though the identification is an arbitrary process, the out-coming meanings should be symbolic and exhibit systematic properties, such as recursiveness and compositionality. Harnad mentions in [13] the example of the word `zebra`, which results from the combination of the words `horse` and `stripes`.

3.5.2 The Physical Symbol Grounding Problem in embodied and situated agents

The *Principle of Embodiment* [188, 189] states that the meaning of external objects and symbols is defined by the physical interactions of the agent with the environment. It has been, moreover, argued that *intelligence* cannot be considered without embodiment and physical interaction [189]. In contrast to symbol systems [186, 13], Brooks suggested in [190] that assuming that the *Physical Grounding Hypothesis* is met, there is no longer a requirement to define *intelligence* as a symbol system. He claims that, provided that an agent is embodied and situated (i.e. it is endowed with a set of sensors to perceive the real world and a set of actuators to interact with it), *intelligent systems* can be designed and understood as purely reactive behavioral patterns ultimately based on sensorimotor control. Under Brook's hypothesis the SGP is effectively and readily solved, due to the absence of symbols.

In [14], Vogt combines Newel's *Physical Symbol Grounding Hypothesis* [186] and Brook's *Physical Grounding Hypothesis* [190], stating the *Physical Symbol Grounding Problem*. Vogt hypothesizes that even though purely reactive systems can fully define specific low-level behaviors (e.g. phototaxis, or obstacle avoidance), symbol systems are required to reach higher-level cognitive and linguistic abilities. The *Physical Symbol Grounding Problem* converts the SGP into a *technical* problem instead of a *fundamental* one by defining the concept of *semiotic symbol*. Similar to *signs* in Peirce's semiotics [191], a *semiotic symbol* is composed by three fundamental parts, *referent*, *meaning* and *form*, and their interactions.

These three elements compose the *Semiotic Triad* or *Semiotic Triangle*, and they are defined

as follows:

- *Referent* is the entity, event or circumstance of the world (external to the agent) on which the *meaning* and *form* are grounded.
- *Meaning* is the internal representation grounded on the external *referents* or objects inside agent's cognitive system. The *meaning* is a private representation of each agent that is shaped by its own experience and interaction with the world.
- *Form* refers to the linguistic symbol or *word* used to refer to external *referents* and private *meanings* in a communication among multiple agents. In contrast to the *meaning*, it is a public representation, and it is negotiated and shared across the societies of cognitive agents.

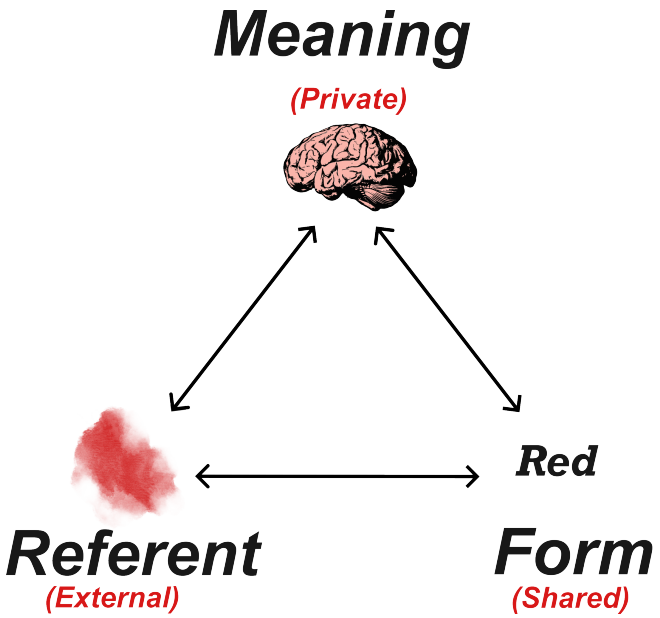


Figure 3.5: Interacting parts of a semiotic symbol.

As an example of a semiotic symbol, the red color present in the real world would be the *referent*, the *meaning* associated to such referent can be understood as the neural activity elicited by the perception and processing of the referent. Finally, the *form*, i.e. the name in a language lexicon to refer to the *referent*, would be the word `red` (considering a hypothetical English-speaking agent). Figure 3.5 depicts the concept of a semiotic symbol, considering also the mentioned illustrative example.

The distinction between *meaning* and *form* and the dichotomy between natural and mental symbols is clearly in line with the language of thought proposed by Fodor [181]. It also resembles the I-Language (internal to the individual) and the E-Language (externalized to others) defined by Chomsky.

The *referent*, *meaning* and *form* terminology is defined in [192] as an alternative terminology to the *representamen*, *interpretant* and *object* defined by Pierce. Pierce's terminology is, in turn, an extension of the *signified* (meaning) and *signifier* (form) of De Saussure's semiotics [193]. As pointed out by Vogt [14], a semiotic symbol is, by definition, grounded on entities in

the real world. Therefore, the SGP reduces to the *technical* problem of finding the *meaning* and *form* for a given *referent*, and establishing the interactions among the three (*semiosis*). According to Vogt, the process of semiosis still requires the *categorization*, *discrimination*, and *identification* processes [13]. The processes of *categorization*, *discrimination* have been tackled in the literature using a wide variety of techniques and algorithms, most of them related to the topics of pattern detection, unsupervised learning and clustering. One of the most popular approaches are *discrimination games* [194, 14, 195, 196]. During a discrimination game, agents aim to find distinctive categories or *prototypes* that belong to a *conceptual space* [197, 198] and solidly and ubiquitously capture invariant properties of external objects. The obtained prototypes are used to create a partition of the conceptual space, so that new analog sensory inputs are categorized using the closest prototype in such space. The word "game" in the algorithm name denotes the conversational games played by two agents to share their knowledge and experience. In addition to discrimination games, other studies have used *category games* [199, 200], Self-Organising Maps (SOM) [201, 202], extensions SOM such as Evolving SOM [203], or Hopfield networks [204], among others.

3.5.3 Social Grounding and language games

In addition to solving the SGP at the individual level by means of creating solid and properly grounded associations among *referents* and *meanings*, there is a much broader problem to be tackled when societies of agents are considered. In these scenarios, in order to fully address the process of *semiosis*, the agents should collectively negotiate and agree a common lexicon, or set of *forms*, that is shared across the entire population. As pointed out in [205], the meaning or internal representation is private to the agent through a *private grounding* process. This private grounding, which is the same as the *physical grounding*, is highly conditioned by the experience of the agent with the external world. Therefore, two embodied and situated agents would generally end up with different sets of meanings of the exact same referents in the real world. These discrepancies are even present in the scenarios where the two agents are homogeneous and share a kin-relationship. Thus, a new dimension is added to the SGP when societies of interacting agents enter into the scene. In short, the following new questions arise. How is it possible that a group of agents ground the same symbols on the same physical entities? What are the social dynamics that lead to the sharing of the same symbols grounded to the same objects to reach an effective communication? Returning to the concept of *semiotic symbol*, the *meaning* can be compared to the internal or mental representation of external entities in the individual's brain or cognitive system. Alternatively, the *form* is related to word in a lexicon that is used to name the corresponding *referent* and to externalize the associated private *meaning*. Therefore, the problem can be regarded as finding the bidirectional mapping among established *meanings* and *words*. The solution to this issue is non-trivial because *meanings* are private and relative to agents and, by extension, so are their mappings to *words*. It implies a proper production, comprehension and sharing of language based on these previously acquired *meanings*. Cangelosi proposes in [206] the division of the SGP into two parts: an individual component that corresponds to the *Physical Symbol Grounding Problem* (how a physical object acquires its meaning), and a social part referred as the *Social Symbol Grounding*.

In the literature, the most common approach to deal with the social symbol grounding are *language games* [207]. Language games encompass a family of algorithms in which populations of communicating agents interact and exchange parts of their vocabularies, according to some cultural dynamics, with the last aim of converging to a common lexicon. Each game is usually regarded as a conversation between two agents, namely, a *speaker* and a *hearer*, that negotiate on some word in the *speaker*'s lexicon. The cultural dynamics that lead to the desired lexicon convergence depend on the specific language game being used. One of the most widely used language game is the *naming game* [208], that has been the most popular in the context of embodied and situated agents [209, 192, 210, 203, 211, 196]. During *naming games*, the *speaker* transmits a word in its lexicon, associated to some context or referent, which is received by some *hearer*. The hearer responds to the incoming message stating whether the word is already stored in its lexicon to refer to the same context or not. Each agent stores a list of word candidates to name a given referent. Depending on the response, the outcome of the game can be either *success* or *failure*. The precise strategy applied by the interacting agents depends on the particular implementation of the naming game (see [212] for a complete overview). Some of the most common strategies are the *Minimal Naming Game* (MNG) [200, 53, 213] and the lateral inhibition strategy [209, 196, 203]. Upon success, applying the minimal strategy implies removing all the words from the list, except the received one. Alternatively, the lateral inhibition proposes a less strict strategy by increasing the importance of the given word while diminishing the relevance of the competing ones. Another highly relevant language game in the context of social and physical symbol grounding is the Iterated Learning Model (ILM) [214, 184, 215, 195], in which the *speaker* and *hearer* roles are engaged by *adult* and *learner* agents, mimicking cultural evolution of language. *Learner* agents are inexperienced language users that acquire new vocabulary and grammar by means of being instructed by *adults*, which are in turn expert language users. The cycle repeats by replacing the adults with the learners of the previous generation, and substituting the previous learners by completely novice individuals. In addition to language games, other studies have used hybrid models that combine artificial evolution and cultural transmission [216, 184], MARL [217], and, combinations between MARL and language games [218].

The mentioned language games by their own can be exclusively used to spread and agree a common lexicon. However, it does not address the broader issue of using the same words associated to the same private meanings and, by extension, grounded on the same external referents. A traditional solution to this problem is the co-optimization of transmission and reception matrices that encode Saussurean one-to-one mappings among *meanings* and *words*. It may be argued that agents should firstly develop a solid basis of meanings and, subsequently, perform the Saussurean mapping among the meanings and some words in a lexicon. It is, nonetheless, important to create *meaning-word* associations in parallel to the lexicon and language formation and propagation [219, 220, 197, 195, 196], which results from the combination of cultural evolution and individual learning [184, 161]. Recalling the definition of *semiosis* and *semiotic symbols* in [14], it can be summarized as simultaneously finding all the three components of the *semiotic triad*. Notice that the co-evolution of *meanings* and *forms* is positioned against innateness of the internal language (I-Language or mentalese) of Chomsky and Fodor. As pointed out by Steels:

> "Thus language and cognition co-evolve. Each one pushes the other up towards

> more complexity and they become tightly co-ordinated with neither a central co-ordinator nor prior innate design." [196]

Steels and Loetzsch proposed in [211] the Grounded Naming Game, an extension of the naming game with situated and embodied humanoid robots aimed at developing a shared lexicon about geometric objects of diverse colors. The robots create a lexicon from scratch by discovering new *categories* (or *prototypes* in their terminology), associating them with words, and communicating those words to reach a shared lexicon. In order to properly associate the words, meanings and referents correctly, the *speaker* robots should somehow be able to draw the attention of the *hearers* (typically in a non-verbal way) to the referent object on which the transmitted word is grounded. That is, both agents involved in the communication should be aware of the subject of the communication. These attention-based language games are usually called *observational games* [153, 195, 221]. In [211], the *speaker* humanoid robot draws the attention of the *hearer* by directly pointing to the targeted geometric object with its finger. Joint attention through direct pointing is not possible or practical in most robotic scenarios, specifically in SR because of the large number of robots and their simplicity. In these cases, a simplified joint attention can be implemented by directly communicating some additional information (e.g. the *iconic representation*) to the *hearer*. It results in the previously mentioned *discrimination games* [194, 14, 195, 196], that combine individual learning and categorization with language games. In addition to *observational games*, joint attention can be also implemented through *guessing games* [192, 195]. In a guessing game, the *speaker* produces an utterance that should be used by the *hearer* to guess the topic of the communication. Thereafter, the *speaker* receives this guess and provides feedback to the *hearer*, which updates its lexicon and *meaning-word* mappings accordingly.

One of the earliest and most relevant experiments addressing the physical and social symbol grounding problems as a whole is the *Talking Heads Experiment* [196, 222, 192]. The *talking heads experiment* consists of two pan-tilted cameras oriented towards a whiteboard with multiple geometric shapes of varying sizes and colors (see Figure 3.6). The robots can sense and process the environment and the objects in it principally via computer vision. Agents in a population are selected to play roles of *hearer* and *speaker* in naming games, being embodied in each of the cameras. Only two agents can engage in a language game by "teleporting" and being instantiated in the robots' body during the game time span. The agents are initialized with an empty vocabulary and category set, so that they emerge as the robots participate in the language games and interact with the environment.

The overall experiment combined discrimination games, for categorization and discrimination of sensory stimuli, naming games, for lexicon formation and propagation, and guessing games, with joint attention based on pointing to the referent figure in the whiteboard. As explained in [196], some of the main hypotheses addressed alongside the Talking Heads research line are the origins and emergence of language through decentralized self-organisation, the co-evolution and incremental complexification of forms and meanings, or the non-innateness nature of grammar (as opposed to Chomsky's LAD and universal grammar). In order to tackle these great concerns, the talking heads experiment was divided into the following layers or levels of abstraction:

- *Perceptual layer*: involves all the low-level processing, segmentation and feature extrac-

Figure 3.6: The Talking Heads Experiment. The source of the figure is [196].

tion of sensory stimuli.

- *Conceptual layer*: is responsible for the categorization and discrimination of the processed stimuli with the aim to create solid *prototypes* or *categorical representations* that capture invariant and distinctive features of external objects.
- *Lexical layer*: devoted to the lexicon management and storage, mappings among meanings and forms, and utterance production.
- *Syntactic layer*: is a more abstract layer concerned with the syntactic categories of words, grammar rules and schemas, and the compositionality of language.
- *Pragmatic layer*: is related to the mechanics and interactions among the robots engaging in the language games.

The concept of *semiotic triad* was also of essence in the talking heads experiment.

In [192], the authors proposed the concept of *Semiotic Landscape*, which has been highly influential in multiple posterior works [14, 195, 196, 221]. The *semiotic landscape* (see Figure 3.7) or *semiotic square* provides a complete picture of the dynamics involved in a grounded language game during a process of *semiosis*. It depicts the *speaker* on one side and the *hearer* on the opposite side, showing the *referents*, *meanings* and *words* (communicated as some utterances) of both agents. It also depicts the diverse processes that take place in order to coordinately build the different mappings (and their inverses) among the components of the *semiotic symbol* of each agent.

Aside from the talking heads experiment, the authors of [205] addressed the problem of grounding a lexicon on geographic and spatial areas, and sharing it across a population of mobile robots. They applied cognitive maps and language games to achieve this complex

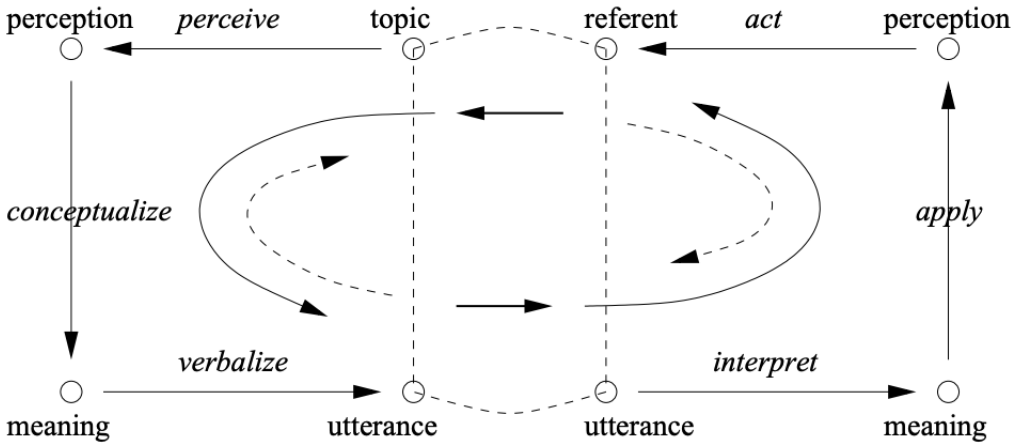


Figure 3.7: Semiotic landscape. The source of the figure is [221].

problem, calling it *Lingodroids*. They used a semiotic triangle similar to Vogt's *semiotic symbols*, with their own terminology: referent, internal representation (meaning), and symbol (form). Using this triad, robots individually learn private internal representations that are grounded on spatial objects or zones. These internal representations were modeled as cognitive maps, that are graphs that interconnect agent's experiences based on locomotion data. In order to negotiate a common lexicon based on the privately grounded internal representations, the mobile robots played language games composed of two types of interactions, namely, *where-we-are* conversations, and *go-to* games. Furthermore, in order to perform the mapping between internal representations an social symbols, they proposed a novel *distributed lexicon table* to tie cognitive map experiences (nodes in the cognitive maps) to words in a lexicon.

Additionally, in [203], the authors evolved a grounded lexicon using a language game. They combined conceptual spaces (see [197, 198]) and a mental simulation framework. Conceptual spaces were used to dynamically cluster and categorize sensorimotor experiences into a set of initially unknown prototypes. The conceptual space was partitioned into a set of *semantic regions* using Evolving Self-Organizing Maps (ESOM) and Voronoi tessellations. A name was assigned by the agents to each of the emerged prototypes in the form of an arbitrary three syllable word. The aim of the language game was that all the agents in the population converge to the same lexicon grounded to the same referent objects. During the language game, a manager orchestrates the iterative process by randomly selecting pairs of agents to engage in the game. The agents "dialog" in order to converge to the same name to refer to the context object proposed by the manager. They assessed the cultural evolution process with three alternative naming strategies, namely, the minimal naming game, the basic lateral inhibition, and the interpolated lateral inhibition. Even though the interpolated lateral inhibition provided the fastest convergence, all the three strategies resulted in communicative success.

Cangelosi, Harnad and colleges addressed the physical and social symbol grounding from a different perspective, away from semiotic symbols and language games [224, 216, 225, 226, 227, 201, 228, 223]. They tackled the SGP using connectionist models (see [228]). In their

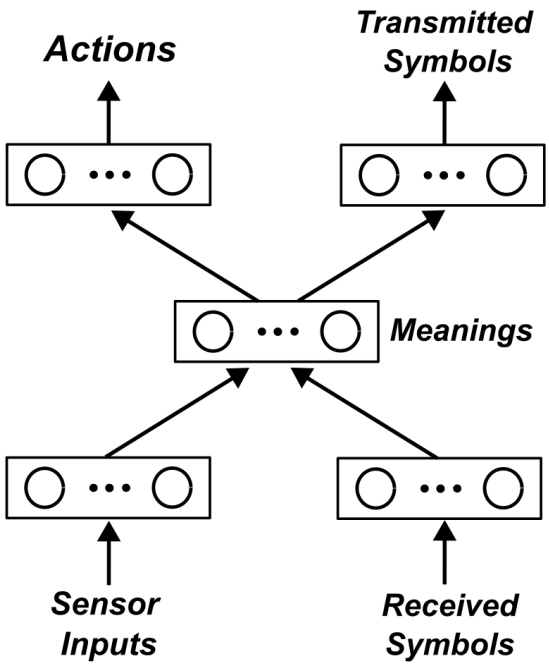


Figure 3.8: Artificial neural network to address the SGP. Figure adapted from [223].

various studies they used ANNs as the unique model for categorization, meaning creation, and language formation and production. Fig. 3.8 illustrates a representative example of the ANN used in [223], where it can be observed that both vision stimuli and received symbols or words are combined and encoded in an internal hidden layer. It is this latent representation what they interpreted as *meaning*. Similarly, the *meaning* was used to define both the behavior of the agent (its actions) and its language production (its utterances).

3.5.4 Compositionality and the emergence of grammar

Compositionality is a distinctive feature of complex languages, such as human languages, that has not been clearly observed in animal communication [229, 230, 231]. There are multiple definitions of compositionality that reflect different aspects of language. In [232], compositionality is understood as the property of mapping similar signals into similar meanings. Besides, we can also consider compositionality from the point of view of grammar and the combination of different syntactic categories (e.g. nouns, verbs, adjectives, etc) to generate new meanings [216, 201]. Cangelosi addressed in [216] the problem of compositionality using a combination of genetic algorithms, artificial neural networks and cultural transmission in a foraging task. Robots were first evolved using a GA and a Multi-Layer Perceptron (MLP) to forage some food items while avoiding others, based on some properties such as edibility and color. After some generations, the agents were allowed to communicate and propagate a lexicon using parent-child interactions. In addition to aiding the foraging task itself, the aim was to develop a compositional lexicon with simple verb-noun syntactic combinations. The naming and compositionality of words was based on an imitation process where children's neural controllers were trained to replicate the words produced by the parents. The artificial

evolution successfully achieved verb-noun rules, albeit there were still some limitations in Cangelosi's work. Firstly, the compositionality performed by means of two clusters of winner-take-all neurons, one for verbs and the other for nouns. Thus, the maximum number of nouns, verbs and compositions constrained by this structure. In addition, it only considers two-word compositions, instead of hierarchical and multi-layered compositional lexicon.

This PhD. Thesis is more concerned with the following definition provided by Vogt (which is similar to the definitions considered in [233, 234, 184]):

> " *Compositionality* refers to a representation (e.g. an utterance) where the meaning of the whole is a function of the meaning of its parts and the way they are combined. " [235].

An example of compositionality under this premise is the word `zebra`, which can be interpreted as the combination of the words `horse` and `stripes` (see [13]). An important aspect of such word composition is that the agents should first develop a solid basis of *elementary* symbols that acquire meaning holistically, and that are directly grounded on the real wold. Built on top of this basic set of symbols, new composed and increasingly more complex symbols arise from the composition of the others. Then, these composed symbols inherit the perceptual grounding of the elementary ones [13].

In [234], the author proposed a computational model to study compositionality and syntax under a cultural evolution basis. In the experiment, an initially non-communicative population of agents obtained a compositional grammar based on the $\langle$`Agent`, `Patient`, `Predicate`$\rangle$ combinations. These three basic components had different syntactic categories, corresponding to nouns in the cases of `Agent` and `Patient`, and verbs in the `Predicate`. The combination of these three terms gave rise to composed expressions, such as the following example provided by the author:

$$\langle \texttt{Agent=Zoltan}, \texttt{Patient=Mary}, \texttt{Predicate=Knows} \rangle.$$

The grammar emerged through a process of cultural transmission and observation of other's utterances. Based on these observations, agents developed new linguistic knowledge by means *invention* and *induction* mechanisms.

One of the lines of research of Vogt [236, 195, 237, 238, 235, 239] has addressed the emergence of compositionality in language through cultural evolution. The agents solved the social and physical symbol grounding using *discrimination games*, *observation games*, and *guessing games* combined with the ILM to produce a shared lexicon perceptually grounded on a simulated *Talking Heads* world. Compositionality was addressed using private grammar rules or *schemas* (holistic and compositional) that were combined to produce complex expressions and new meanings. The grammar rules were learnt incrementally from scratch through sophisticated mechanisms such as *induction*, *invention*, *generalization*, *interpretation*, or *exploitation*, among many others. For instance, if an agent receives a compositional expression $R_1 \circ R_2 \circ R_3$ and it is only capable of understanding some of its parts (say rules R_1 and R_2), then the hearer *invents* a new rule to fill the gap of the unknown component R_3. Alternatively, compositional expressions that are completely unknown for the hearer agent result in the *exploitation* and *invention* of a new holistic rule.

In [215], Brighton studied the role and importance of the cultural evolution and transmission

of language in the emergence of structural languages, specifically the compositionality of language. He hypothesized that, even though biological evolution of the LAD [180] is relevant up to a certain degree, compositionality in a language arises as a result of cultural evolution mechanisms. In his work, he supported this claim by means of the ILM. Abstract objects in an environment were conceptualized as meanings, which were in turn mapped onto signals in a dictionary by means of agent hypotheses. It was shown that the evolution of language occurs when there is a transmission bottleneck, so that agents are not completely exposed to the whole language from previous iteration. Moreover, it was also found that the structural complexity of the meaning space, used to conceptualize the objects, also plays a crucial role when determining the advantages of compositional languages over holistic ones.

Riga et al. [201] extended the work in [225, 227] to address the symbol grounding transfer and the compositionality of symbols to create new concepts. A single agent firstly learns to categorize and name basic concepts directly tied to elemental properties of the referent objects, such as shape and color. Subsequently, new words were learnt in a supervised way by connecting previously formed concepts (e.g. `"Red & Square = DAX"`). Their system was composed by two neural networks and two coexisting learning algorithms. Firstly, a SOM was optimized through unsupervised learning to categorize the input stimuli and build a feature map. Thereafter, supervised learning, based on the standard backpropagation algorithm, optimized a MLP by showing the agents the new abstract symbols and the list of previously acquired concepts.

3.5.5 Emergence of language in Swarm Robotics

In the field of SR, studies are limited either to using language games as an optimization tool to solve certain SR tasks (e.g. aggregation) [240, 241, 53], or to the convergence of lexicons of few words [213].

In [53], the authors explored the application of MNG to solve an aggregation task in a swarm of kilobots. They proposed a combination of probabilistic aggregation and cultural evolution of language games to efficiently achieve an adaptive aggregation behavior. The aim of the MNG was to dynamically find a proper tuple of parameters of the PFSM that defines the robots' control. When convergence was reached in the cultural evolution process, all the agents ended with the same word to refer to the same set of PFSM parameters. Moreover, they proposed a variant of the MNG in which only listeners update their lexicon and robots can only engage in the speaker role when their PFSM is in a specific state correlated with a correct behavior.

Miletitch et al. proposed in [213] one of the few applications of language games in large scale SR, for uniquely naming relevant parts of the environment. In thier work, language games were executed on top of a foraging task. The robot foraging behavior was based on the Nest-Site Selection (NSS) algorithm, while the MNG was the language game that they used. Robots enganged in the foraging task while they played the MNG in parallel with nearby neighbors, with the objective of naming two food areas in the arena with the same words, upon convergence. The authors applied two variants of the MNG for SR, namely, the *classic game* and the *spatial name*. These two variants differ on the way that robots create

new words, being the former the regular word creation process in the MNG, and the latter a creation of new words based on food source discovery through exploration. As it is common in works combining SR and language games [241, 53], the speaker broadcasts a word of its lexicon to all its neighbors in communication range, that act as listeners. Moreover, only the listeners update their lexicons on the basis of the received word depending on the outcome of the game.

Lexicon formation and sharing has not been sufficiently addressed in SR from the perspective of semiotics, making a distinction between private meanings and shared forms. Moreover, compositionality has not been addressed with swarms of minimal, situated and embodied robots with limited communication and perception coverage. Therefore, we believe that there is a considerable research path to be explored in SR, to pave the way for more complex and elaborate language-based communications beyond signaling [11].

Chapter 4

Evolution of communication in swarm robotics primitives

The evolution of communication is one of the most challenging problems in multi-agent systems and artificial life. It requires the co-evolution of both signal emission and its corresponding response, processing, and identification (see Chapter 3 for more details). Put simply, it is like *searching for a needle in a haystack*. This problem becomes significantly more complex if the aim of the evolution of communication is to solve complex and intricate tasks. Therefore, dividing the whole problem into low-level primitive behaviors may be essential to substantially simplify the evolutionary process. In a context of evolution of communication in the animal realm, it has been widely hypothesized that communication mechanics and semantics originate from pre-existing non-communicative primitive behaviors (see e.g.[166, 167]). Furthermore, the division into basic primitives is also a recurring factor when modeling complex collective behaviors. For instance, flocking behaviors are commonly decomposed into three low-level rules, (i) cohesion, (ii) separation, and (iii) alignment, that are combined in a motor schema to produce the desired collective motion (see Chapter 2.5.2). Additionally, foraging behaviors can be decomposed into low-level primitives (e.g. explore, stay in nest, forage from known area, or deposit pheromones) that can be combined using subsumption or FSM models. Therefore, this chapter is mainly focused on the evolution of low-level SR primitives that are the basis of many high-level behaviors.

Besides, an important question when designing communication systems in SR, guided by an evolutionary process, is whether or not it is possible to solve remarkably different tasks using the same robot controller. In Chapter 1, this feature was introduced as *transferability*, which can be either *weak* or *strong*. In this Chapter, the investigation focuses on SR systems with *weak transferability* properties, so that the same robot controllers and communication systems can be used in multiple tasks with minor re-design. However, in this level of transferability, the systems must be re-evolved for every task being addressed. Another driving motivation was to assess whether or not artificial evolution can lead to the emergence of different communication semantics depending on the task to be solved, using the same robot controller.

In this chapter, these concerns are addressed. For this aim, a minimal IR-based communication

system is proposed, and CTRNN-based robot controllers that can harness such communications are optimized using GAs. The system is evaluated in three different SR primitive behaviors. In all of them, robots are endowed with the same communication capabilities, dictated by a highly constrained and minimal communication. The communication system establishes the communication mechanics (e.g. the communication range, morphology of the message, and context variables that can be received), but the communication semantics and the way in which the robots communicate are purely guided by the artificial evolution. The same robot neuro-controller is separately evolved in three low-level SR primitives, assessing whether *weak transferability* is fulfilled.

4.1 Methodology

4.1.1 The robots and the environment

In all the experiments of this chapter a simulated environment is used. Specifically, the robots can move within a 10m×10m square area without any obstacle other than the boundary walls. The robot swarm is denoted as $\mathcal{R}$ and, hereafter, we refer to a generic robot as $r \in \mathcal{R}$. A robot r is abstractly represented with a position $\mathbf{x}_r$ and a heading orientation θ_r. Additionally, even though the environment is a three-dimensional world, the motion of the robots is restricted to the XY plane. Therefore, the formulation of the positions and heading orientation is conveniently simplified to $\mathbf{x}_r \in [-5, 5]^2$ and $\theta_r \in [0, 2\pi)$. The simulated robots take inspiration from the e-puck robot [107], and are modeled as simple mobile robots with a differential drive system, a set of sensors, and a set of actuators. Figure 4.1 illustrates the most basic arena used in the experiments, which can be subject to some variations depending on the specific requirements of each task. Moreover, the figure also shows a zoom to one of the simulated robots.

4.1.2 The communication system

In all the experiments and SR tasks of this chapter a simplistic IR-based communication system is used. Such communication system is based on the range and bearing technology [108], that operates on top of the IR transmitters and receivers of the simulated mobile robot. The range and bearing module allows the robots to acquire information about (i) messages modulated on the transmitted IR signal, (ii) the estimated distance of the transmitting robot, and (iii) an estimation of the relative orientation from where the message was received. In this chapter, the emergence of communication is evaluated under constrained and difficult communication conditions, and, therefore, a minimal and simplistic communication system is proposed. Its main characteristics are the following:

- The communication is local, with a small communication range, which implies that two robots can communicate if their distance is lower than a threshold. The communication range is fixed to 80 cm (see [108]).
- A robot can only receive and process a single simultaneous message, regardless of the number of concurrent transmitters in range. Consequently, in order to process multiple messages from many simultaneous senders, some message selection mechanism

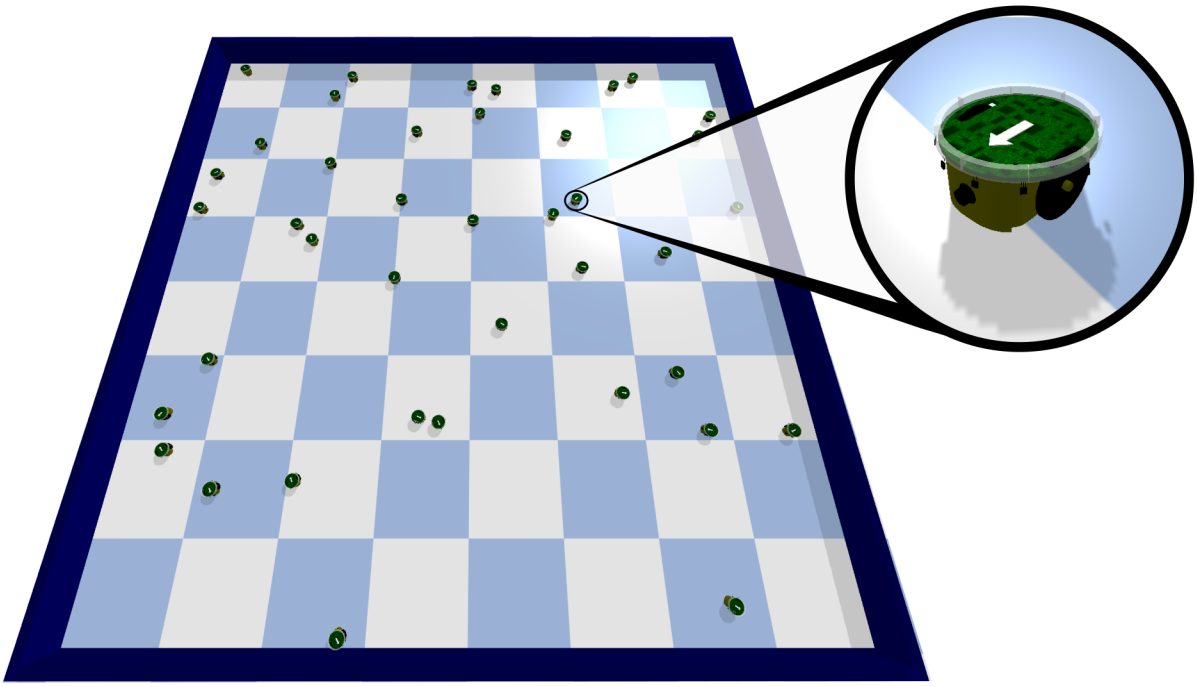


Figure 4.1: Basic configuration of the arena used in the experiments of Chapter 4.

is designed to pick one messages according to some established criteria. A detailed description of the message selection mechanism is provided below in this subsection. Receiving a single message also entails that the robot is only aware of one neighbor at each control cycle.

- Even though the robot is equipped with eight IR devices positioned along its perimeter, the number of IR sectors are deliberately restricted to four. This is a hard constrain that makes the communication mechanics and the task solving much more challenging. Specifically, the increased complexity lies on the fact that most of the tasks proposed in this chapter require the emergence of a situated communication that makes use of the relative orientation from where messages are received.
- The context of the communication that is received and processed corresponds uniquely to the current perceived message.

Transmission

The communication system endows the robots with the capability to encode a potentially meaningful signal $\mathbf{a}_{tx}$. This signal is broadcasted simultaneously by all of the four IR communication transmitters.

The signal $\mathbf{a}_{tx}$ is constrained to the hypercube $[0, 1]^M$, being M its dimension to be specifically designed for each task. $\mathbf{a}_{tx}$ is originally produced by the robot controller and, subsequently, it

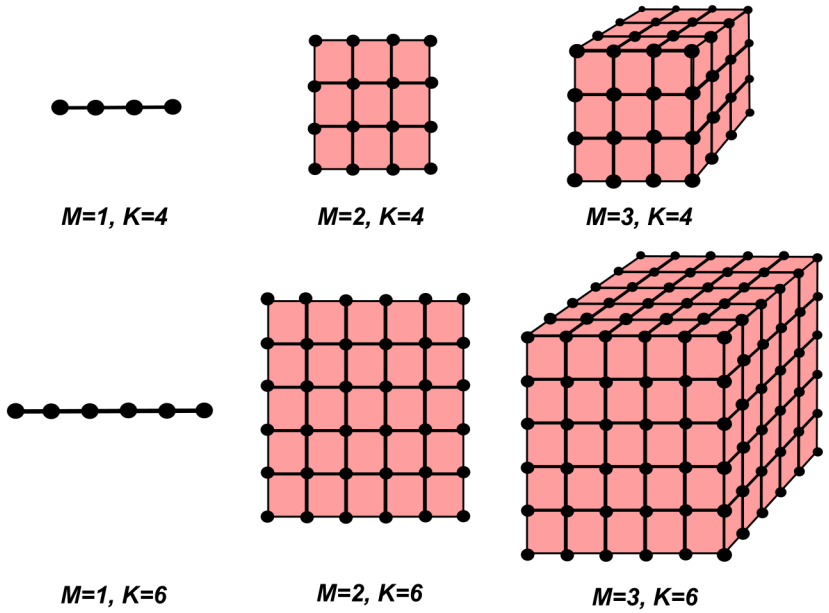


Figure 4.2: Illustration of the constellations in $\mathcal{C}$ for different values of M and K. The symbols are depicted as black dots.

is subject to a quantization process $\mathcal{Q}$. The function $\mathcal{Q}$ is defined in Equation 4.1,

$$\begin{array}{rccc} \mathcal{Q}: & [0,1]^M & \longrightarrow & \mathcal{C} \\ & \mathbf{a}_{tx} & \longmapsto & \mathbf{m}_{tx} \end{array} \tag{4.1}$$

where $\mathcal{C}$ is the set of possible symbols. The set $\mathcal{C}$ is designed as a hypercubic lattice constrained in $[0,1]^M$ (although it is not the only feasible constellation). Equation 4.2 formalizes the definition of $\mathcal{C}$ (see Figure 4.2),

$$\mathcal{C} = \left\{0, \frac{1}{K-1}, \frac{2}{K-1}, \ldots, 1-\frac{1}{K-1}, 1\right\}^M, \tag{4.2}$$

considering K as the number of rows in the lattice.

Algorithm 1 Message Quantization

procedure $\mathcal{Q}(\mathbf{a}_{tx})$
  for $\mathbf{c}_i \in \mathcal{C}$ **do**
    $\mathbf{d}_i \leftarrow \|\mathbf{a}_{tx} - \mathbf{c}_i\|_2$
    $\mathbf{z}_i \leftarrow 1 - \mathbf{d}_i / D_{\max}$
  end for
  $\mathbf{p} \leftarrow \text{Softmax}(\mathbf{z}_i), \forall i$
  $\mathbf{m}_{tx} \sim \text{Cat}(\mathcal{C}, \mathbf{p})$
end procedure

The dimension of $\mathcal{C}$ (M) and the number of rows (K) can be used to compute the total number of symbols in the constellation (K^M). Figure 4.2 illustrates the set of symbols in $\mathcal{C}$ for different combinations of M and K.

Algorithm 1 summarizes the list of operations that take place during the quantization process. Firstly, the distances from $\mathbf{a}_{tx}$ to every possible symbol in $\mathcal{C}$ are computed and collected in a vector $\mathbf{d}$. Thereafter, the distances in $\mathbf{d}$ are transformed to the vector $\mathbf{z}$, which takes values in the interval $(-\infty, 1]$. This transformation assigns smaller distances to higher values of $\mathbf{z}$. The vector $\mathbf{z}$ is used as the input of a Softmax function (see Equation 4.3), which assigns a probability to each component of the input vector.

$$\text{Softmax}(\mathbf{z})_i = \frac{e^{\beta \mathbf{z}_i}}{\sum_{j=1}^{J} e^{\beta \mathbf{z}_j}},\ \forall j \in \{1, \ldots, J\} \tag{4.3}$$

Therefore, the softmax operation results in a probability vector $\mathbf{p} \in [0, 1]^{K^M}$ that represents the probability of transmitting each symbol in $\mathcal{C}$. Clearly, those symbols that are closer to the raw message $\mathbf{a}_{tx}$ would have higher probability of being sent. Finally, a specific symbol $\mathbf{m}_{tx}$ is selected using the categorical distribution (Cat) using the probabilities in $\mathbf{p}$ on the set $\mathcal{C}$. The parameter β in Equation 4.3 is a temperature constant that controls the balance among the element probabilities. When $\beta \to 0$ then the probabilities tend to the same value (random communication), whereas as β grows the communication becomes increasingly more sparse and less random (concluding in a purely deterministic message selection when $\beta \to \infty$).

Reception

In addition to the four IR communication transmitters, each robot is equipped with four IR communication receivers that can be used to perceive the messages broadcasted by others neighbors in communication range. Specifically, the IR devices of each sector are positioned at the orientations in $\{0, \pi/2, \pi, 3\pi/2\}$ relative to the heading orientation θ_r. Each sector can perceive an incoming message only from the closest neighbor inside its coverage area. We denote $\mathbf{m}_{rx}^{(\alpha)}$ as the message received by the IR device located at sector with orientation $\theta_r + \alpha$, being $\alpha \in \{0, \pi/2, \pi, 3\pi/2\}$.

Even though each communication sector receives information independently, only one message is ultimately processed and provided as input to the robot controller. The final message, $\mathbf{m}_{rx}$, is the result of some selection mechanism, and it is one of the messages received each IR device. In this PhD Thesis, the message selection mechanism is formulated as in Equation 4.4,

$$\mathbf{m}_{rx} = \underset{\langle \mathbf{P}_{rx}, \mathbf{D}_{rx} \rangle}{\text{Selection}} \left(\mathbf{m}_{rx}^{(0)},\ \mathbf{m}_{rx}^{(\pi/2)},\ \mathbf{m}_{rx}^{(\pi)},\ \mathbf{m}_{rx}^{(3\pi/2)} \right), \tag{4.4}$$

where $\mathbf{m}_{rx}$ is the final message after the selection process and $\mathbf{m}_{rx}^{(\alpha)}$ is the message received from the sector positioned at $\theta_r + \alpha$ radians. The selection mechanism is based on a Probabilistic Finite State Machine (PFSM) with time delays, with four states corresponding with each sector. The transition probability matrix is denoted as $\mathbf{P}_{rx}$, and the time delays of each transition are gathered as the matrix $\mathbf{D}_{rx}$. The components p_{ij} (see Equation 4.5a) represent the probability of being in state i (listening to messages of *sector* i) and transitioning to state j (selecting messages from *sector* j). Similarly the time delays τ_{ij} (see Equation 4.5b) are the time elapsed to transition from state i to state j.

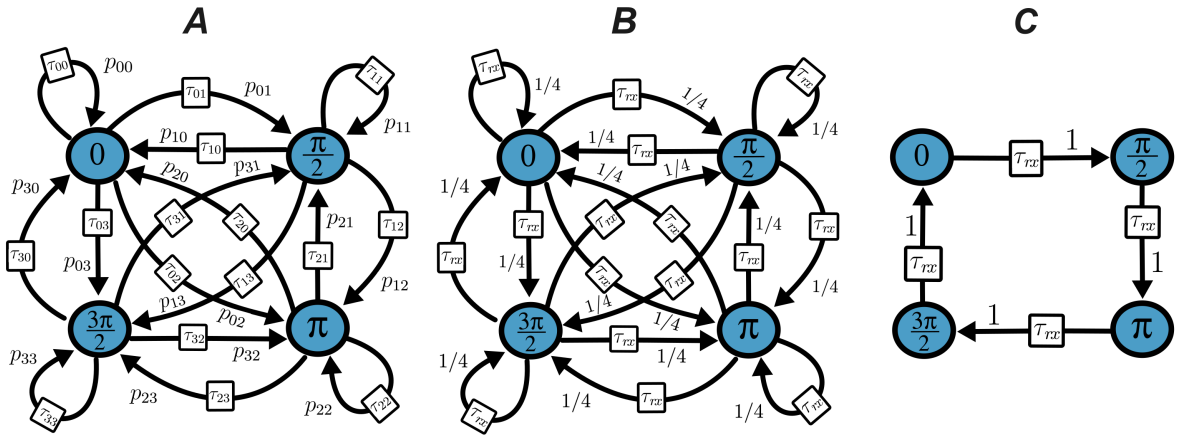


Figure 4.3: Diagram of the message selection mechanism. (**A**) General diagram of the message selection process. The blue circles represent the PFSM states, corresponding to each communication sector and the arrows illustrate the state transitions, with their corresponding transition probabilities. The white squares attached to each arrow depict the time delays of the state transitions. (**B**) Random selection with time delays and uniform probabilities. (**C**) Cyclic selection with time delays.

$$\mathbf{P}_{rx} = \begin{pmatrix} p_{00} & p_{01} & p_{02} & p_{03} \\ p_{10} & p_{11} & p_{12} & p_{13} \\ p_{20} & p_{21} & p_{22} & p_{23} \\ p_{30} & p_{31} & p_{32} & p_{33} \end{pmatrix} \quad (4.5a) \qquad \mathbf{D}_{rx} = \begin{pmatrix} \tau_{00} & \tau_{01} & \tau_{02} & \tau_{03} \\ \tau_{10} & \tau_{11} & \tau_{12} & \tau_{13} \\ \tau_{20} & \tau_{12} & \tau_{22} & \tau_{23} \\ \tau_{30} & \tau_{31} & \tau_{32} & \tau_{33} \end{pmatrix} \quad (4.5b)$$

Figure 4.3A illustrates the diagram of the PSFM, showing the states as the blue circles, the probabilities p_{ij} attached to each transition arrow, and the time delays of each transition as white squares.

It is important to notice that time delays do not alter the destination of the transition and they are only effective after the next state is determined. The reason behind including the delays in $\mathbf{D}_{rx}$ is to avoid constant switching of the selected messages, and to allow the robots to listen to the same sources for a guaranteed duration. Besides, in the case that no message is received from some sector i positioned at $\theta_r + \alpha$ radians, then we fix $\mathbf{m}_{rx}^{(\alpha)} = \emptyset$ and assign $p_{ji} = 0,\ \forall j \in \{0, \ldots, 3\}$.

In this PhD Thesis, the following selection schemes are considered:

- **Random selection with time delays**: the communication sector is selected randomly among the four possible alternatives, with equal probability. Moreover, each connection has some fixed delay τ_{rx} in order to guarantee that the robot listens to a sector before randomly switching to another one. Figure 4.3B shows the diagram of the random selection with time delays. Additionally, the matrices $\mathbf{P}_{rx}$ and $\mathbf{D}_{rx}$ are shown in Equations 4.6a and 4.6b,

$$\mathbf{P}_{rx} = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix} \quad (4.6a) \qquad \mathbf{D}_{rx} = \begin{pmatrix} \tau_{rx} & \tau_{rx} & \tau_{rx} & \tau_{rx} \\ \tau_{rx} & \tau_{rx} & \tau_{rx} & \tau_{rx} \\ \tau_{rx} & \tau_{rx} & \tau_{rx} & \tau_{rx} \\ \tau_{rx} & \tau_{rx} & \tau_{rx} & \tau_{rx} \end{pmatrix} \quad (4.6b)$$

- **Cyclic selection with delay**: Alternative to the random selection scheme, the cyclic selection method follows a cyclic transitioning pattern, starting at communication sector $\alpha = 0$ and, thereafter, endlessly following the sequence $\{0, \pi/2, \pi, 3\pi/2, 0, \pi/2, \dots\}$. To ensure a minimal listening duration in each sector, each transition adds a time delay of τ_{rx} seconds. Figure 4.3C illustrates the diagram of the random selection with time delays. Additionally, the matrices $\mathbf{P}_{rx}$ and $\mathbf{D}_{rx}$ of the cyclic selection are shown in Equations 4.7a and 4.7b,

$$\mathbf{P}_{rx} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (4.7a) \qquad \mathbf{D}_{rx} = \begin{pmatrix} 0 & 0 & 0 & \tau_{rx} \\ \tau_{rx} & 0 & 0 & 0 \\ 0 & \tau_{rx} & 0 & 0 \\ 0 & 0 & \tau_{rx} & 0 \end{pmatrix} \quad (4.7b)$$

In addition to the listed ones, the proposed communication system admits many other selection schemes. For example, an "*intelligent*" scheme may consider some communicative value of each sector, computed as the likelihood of receiving messages that are more relevant for solving the task. Nonetheless, these advanced selection schemes are left as future work, and in this PhD Thesis only the random and cyclic selection mechanics are considered in the experiments of this chapter.

The range and bearing technology [108] allows the acquisition of not only the abstract message but also the estimated distance relative orientation of the sender. Thus, in addition to $\mathbf{m}_{rx}$, the robot also perceives the environmental context associated to such message and, consequently, the artificial evolution is endowed with the opportunity to obtain either abstract or situated emerged communications. In this chapter, the context variables are

(1) the estimated distance, ρ_{rx}, to the transmitting robot,

(2) the relative orientation, θ_{rx}, from where the message was received, and,

(3) the relative orientation, θ_{tx}, from where the message was transmitted by the sender robot.

Notice that θ_{rx} and θ_{tx} are computed relative to the heading orientation θ_r of the receiving and transmitting robots, respectively. Essentially, θ_{rx} is the orientation of the sector receiving the message ($\mathbf{m}_{rx} = \mathbf{m}^{\alpha}_{rx} \Rightarrow \theta_{rx} = \alpha$), whereas θ_{rx} is the orientation of the sector from where the message was originally transmitted. Consequently, θ_{rx}, θ_{tx} belong to the finite set $\{0, \pi/2, \pi, 3\pi/2\}$.

For the estimation of ρ_{rx} (distance estimation), a similar computation as in the IR proximity

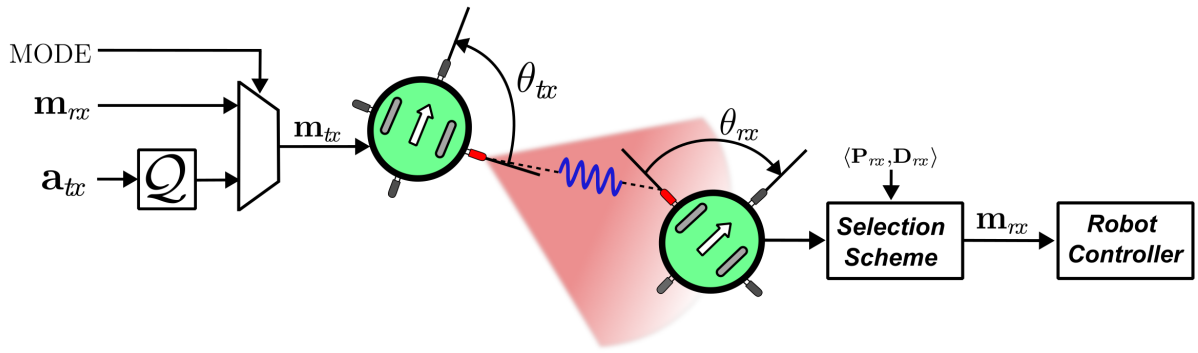


Figure 4.4: Overall diagram of the communication system

sensor is used. In short, the values of ρ_{rx}, ranging in $[0, 1]$, decay exponentially with the distance to the target object, so that a value of one indicates that the transmitting robot is very close to the IR receiver.

Communication state

We propose the incorporation of a communication state that alters the transmitted message of each agent depending on its value. The state of the communication is defined as a binary variable MODE, that can can equal to the values *relay* (0), or *send* (1). Send mode implies that the communication transmitter broadcasts the new message generated by the robot controller, based on the incoming message from its neighborhood. On the contrary, in relay mode, the transmitter ignores the message elaborated by the robot controller and, instead, transmits the input message perceived by the receiver (acts as a communication relay). The commutation between these states will be performed by the robot controller logic. Moreover, a maximum number of hops is established so that old messages eventually are discarded, avoiding information overflow. Each message has an associated identifier of its original sender, which is used to filter out incoming messages with its own identifier.

Overall communication system

After the selection process explained above, the robot controller receives as input the tuple in Equation 4.8,

$$\phi_{rx} = \begin{pmatrix} \mathbf{m}_{rx} \\ \rho_{rx} \\ \theta_{rx} \\ \theta_{tx} \\ \text{MODE} \end{pmatrix} \tag{4.8}$$

that gathers the abstract message, its associated context, and the current communication state. In addition to other sensor readings, these variables are used to generate the next message to be transmitted, and the next value of the communication state (MODE) of the receiver robot. Figure 4.4 illustrates the overall diagram of the communication system showing the transmitting robot on the left side and the receiving robot on the right side.

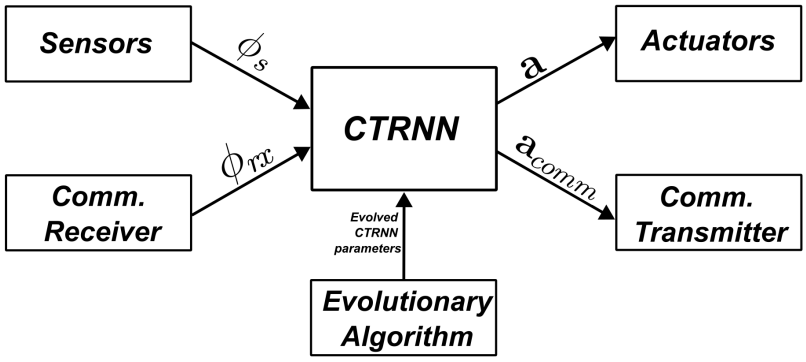


Figure 4.5: Generic diagram of the robot controller used in the experiments of Chapter 4

4.1.3 The robot controller

A robot controller can be described as a black box that maps the measurements read by the robot's sensors into the actions used to control the robot's actuators. Its logic entirely defines the behavior of the robots, and it can be implemented using multiple algorithms and models. In general terms, the robot controller of this Chapter is defined as in Equation 4.9,

$$\mathbf{a},\, \mathbf{a}_{comm} = \text{Controller}\left(\phi_s,\, \phi_{rx}\right), \tag{4.9}$$

which is fed with the sensor readings, ϕ_s (e.g. measurements of the proximity sensor), and with the information received by the communication receiver, ϕ_{rx} (see Equation 4.8). Additionally, its output is composed by the actions that control the robot actuators (e.g. the motor wheels), $\mathbf{a}$, and the actions that control the communication transmitter (e.g. the message to be transmitted), $\mathbf{a}_{comm}$.

In all the experiments of this chapter we use neural controllers based on CTRNN models, which is a type of ANN with feedback connections that operates in continuous time (see Appendix B for detailed theoretical background). Moreover, the parameters of the CTRNN controller (i.e. synapse weights, neuron biases, and membrane time constants) are evolved using a standard genetic algorithm (see Appendix A). The robot swarm is homogeneous, so that a copy of the same evolved CTRNN is used in all the robot controllers. Nonetheless, the evolution process is independent for each of the treated primitives of this chapter. Figure 4.5 illustrates the general diagram of the robot controller, which is a generic architecture used as the basis for the primitive tasks.

4.2 Leader election

4.2.1 Description

The leader election primitive is a negotiation and coordination process among the robots with the final aim of selecting a unique leader of the whole swarm. The robot swarm is fully homogeneous and decentralized, so that, at the beginning of a simulation, all the robots have the same opportunities to claim the leadership. A robot becomes a potential leader when it

turns its LED on. The control of the LED actuator is accomplished by the robot controller of each agent. The leader election primitive is depicted in Figure 4.6, showing how the robot swarm converges to a unique leader of the group.

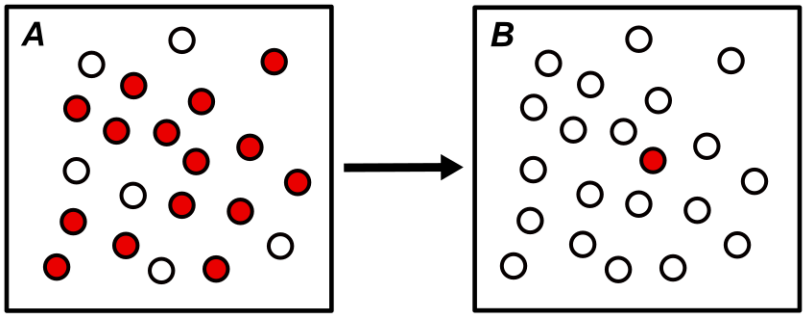


Figure 4.6: Illustration of the leader election primitive. (**A**) Initial state. (**B**) Correct final behavior. Red circles represent the robots claiming leadership, while white circles represent the *follower* robots.

In order to avoid excessive leadership switching between two or more robots, behaviors leading to the convergence of the same stable leader are highly rewarded. However, it is also critical that the robots develop the ability of reacting to unexpected events or perturbations during the execution time. Specifically, the robot swarm must be robust enough to detect and react to the failure of the current leader by selecting a new candidate. To guide artificial evolution towards this behavioral feature, any robot becomes faulty when it has been the unique leader for 50 consecutive time steps. Furthermore, a faulty robot is defined as an agent that can no longer become the leader, and that it is only capable of relaying incoming messages but not to elaborate new ones. Robots are static, so that they cannot alter neither their positions, $\mathbf{x}_r$, nor their orientations, θ_r. Consequently, the initial conditions of $\mathbf{x}_r$ and θ_r, determined randomly, are the same during the entire simulation episode. Moreover, the random initialization of the positions ensures that there are not isolated robots in the swarm.

4.2.2 Robot controller

The CTRNN architecture used as the robot controller of the leader election task is depicted in Figure 4.7, showing a topology with two hidden layers of 10 neurons each (colored in blue). There are five input nodes (green circles), that are fed with the received message $\mathbf{m}_{rx}$, the associated value of θ_{rx}, and the current communication state MODE. Additionally, the architecture is composed by 4 output neurons, colored in red. Specifically, the output a_L controls the LED actuator, to either claim or reject leadership, the $\mathbf{a}_{tx}$ is the new raw message to be posteriorly quantized and transmitted, and MODE is the next communication state of the robot. Notice that there is a post processing stage that transforms a_L and MODE, which belong to the set $[0, 1]$, to binary values using a step function. The weights of all the synapses, and biases and time constants of all the neurons are evolved using a GA. All the neurons use a sigmoid activation function.

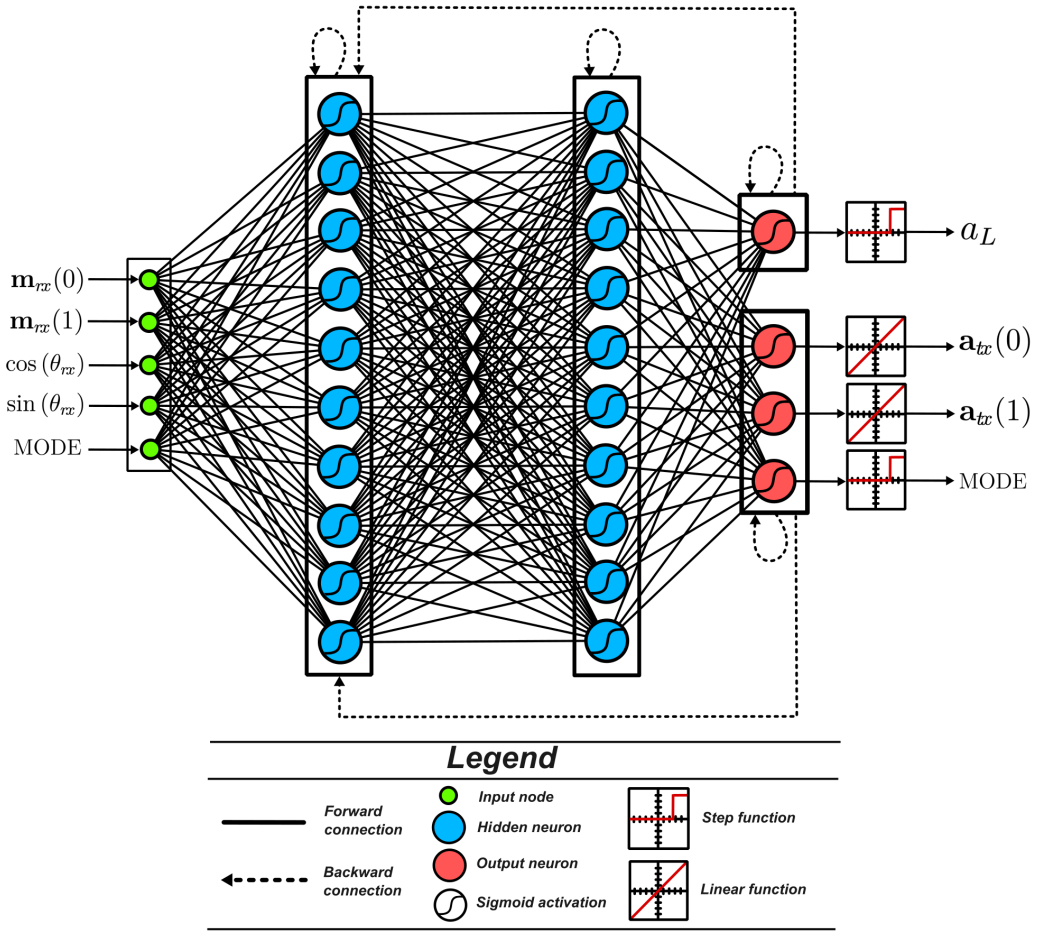


Figure 4.7: Artificial neural network architecture used in the leader election and frontier identification tasks.

4.2.3 Fitness function

The fitness function used in the leader election task is denoted as f_A, and it is formulated in Equation 4.10,

$$f_A(t) = \begin{cases} \min\left(f_A(t-\Delta t) + 0.1,\, 5\right), & \text{If } \|\mathbf{a}_L(t)\|_1 = 1 \text{ and } \mathbf{a}_\text{L}(t) = \mathbf{a}_\text{L}(t-\Delta t) \\ 0.1, & \text{If } \|\mathbf{a}_L(t)\|_1 = 1 \text{ and } \mathbf{a}_\text{L}(t) \neq \mathbf{a}_\text{L}(t-\Delta t) \\ 0, & \text{Otherwise} \end{cases} \tag{4.10}$$

where $\|\cdot\|_1$ is the L^1 norm of some vector, and Δt is the period of the control cycle. Moreover, $\mathbf{a}_\text{L} \in \{0,\, 1\}^R$ is a binary vector that collects the LED actions of all the agents in the robot swarm. Thus, the i-th component of $\mathbf{a}_\text{L}$ is the action that turns either on or off the LED of the i-th robot. For instance, in a group of three robots with $\mathbf{a}_\text{L} = (0, 1, 0)^\top$, the first and third agents have their LED turned off whereas the LED of the second robot is turned on. The first case of the equation is met when there is only one leader ($\|\mathbf{a}_L(t)\|_1 = 1$), and such leader

is exactly the same as in the previous control cycle ($\mathbf{a}_{\mathrm{L}}(t) = \mathbf{a}_{\mathrm{L}}(t - \Delta t)$). In case of fulfilling this condition, the value of the fitness is increased by 0.1, up to a maximum value of 5. Thus, the fitness grows proportionally to the number of consecutive time steps with a unique and stable leader. Alternatively, the condition in the second case of Equation 4.10 is also met when there is only one leader. Nonetheless, it differs from the previous condition because it is only fulfilled when the current leader is not the same as the one at $t - \Delta t$, implying that there is a new leader. Finally, the third case is fulfilled when there is either zero or more than one leader, resulting on a fitness of zero.

The fitness score of a whole trial is computed as the temporal average of the instantaneous fitness value of each evaluation time step (see Equation 4.11),

$$FF_A = \frac{1}{T} \sum_{k=0}^{T} f_A(k\Delta t), \tag{4.11}$$

where T is the number of control cycles of an episode, and Δt is sampling period of the robot control. It means that the robot controller, and thus the fitness calculation, is executed every Δt seconds. In this experiment we fixed Δt as 50 ms and T as 500 cycles. Finally, each individual is evaluated in five independent trials, so that multiple fitness samples are used to construct the final expected fitness estimate. The final estimate is the sample mean of the fitness values obtained in each trial.

4.2.4 Results

Behavior and overall performance

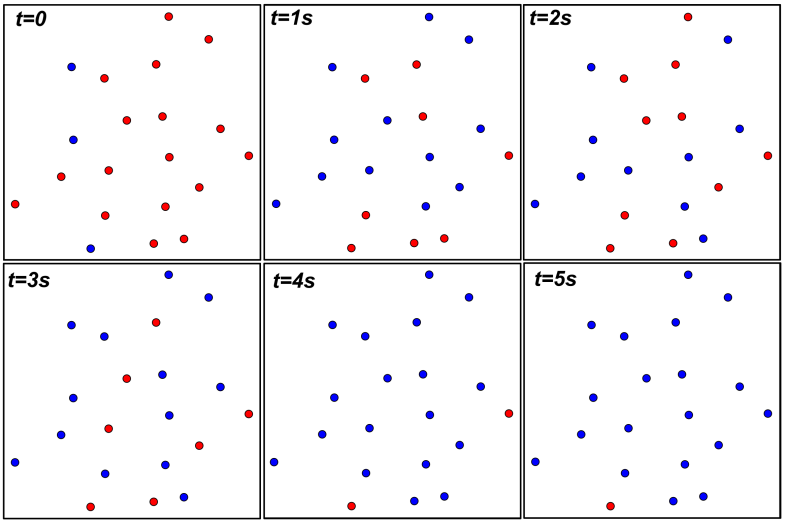


Figure 4.8: Results of the leader election task during a single simulation, showing the leaders at different time instants. Red circles represent leader robots, and blue circles represent follower robots.

Figure 4.8 illustrates the evolved behavior during a simulation of 5 seconds. It shows six frames of the trial at different time instants of the simulation, highlighting the robots that claim leadership as red circles. It can be observed that at the early time instants of the

simulation most of the robots claim the leadership of the swarm. However, as time elapses, the robots communicate and engage in a negotiation process that results in a decrease of the number of leaders. Such negotiation ultimately converges to a unique leader of the group.

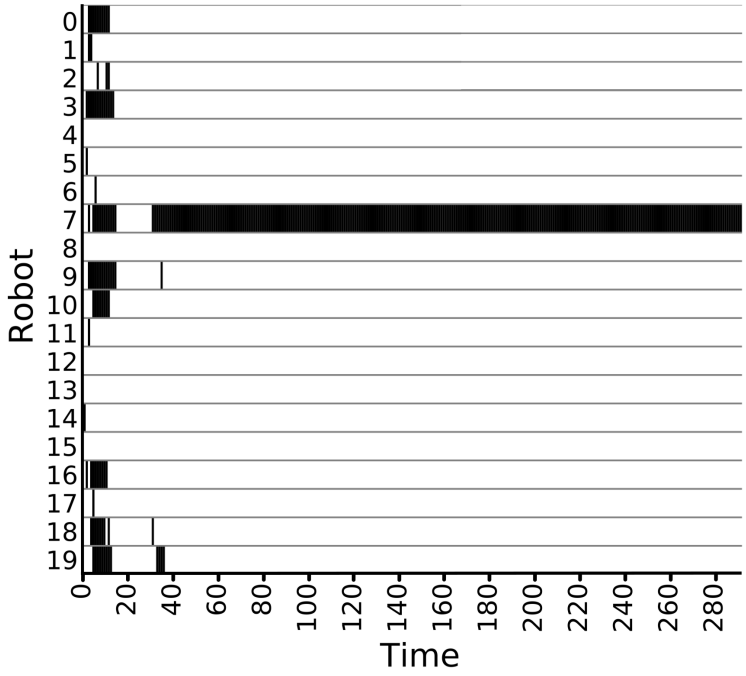


Figure 4.9: Results of the leader election task during a single simulation, showing the LED actions of each robot across simulation time. For each robot (X axis), it is shown the LED binary action either in black (LED turned on) or in white (LED turned off).

The negotiation process can be observed in Figure 4.9, that shows the evolution of the LED actions of the robots in a swarm of size 20. The horizontal black bars indicate that the LED of the corresponding robot in the ordinate axis is turned on. The negotiation phase, starting at time step 0 and concluding approximately at control cycle 40, is a *winner-take-all* process until only a single robot has its LED turned on. Figure 4.9 additionally shows a short time window (between time steps 16 and 30) in which all the robots have their LED are turned off and, thus, there is no leader of the swarm. Nonetheless, this unsuccessful negotiation is rapidly solved by a much shorter second negotiation process, involving only four robots, that successfully concludes with a unique leader. Such elected leader is stable and preserved along the rest of the simulation.

Scalability

The scalability of the solution is assessed by simulating the experiment with different swarm sizes. Moreover, instead of depicting the results for a single trial, the results are collected among 50 independent simulations, with random initialization, to provide a statistically significant number of observations. Figure 4.10 shows the impact that the swarm size has on the performance in the leader election task. It is a box plot that illustrates, for each swarm size, the percentage of the total simulation time in which just a single robot claims

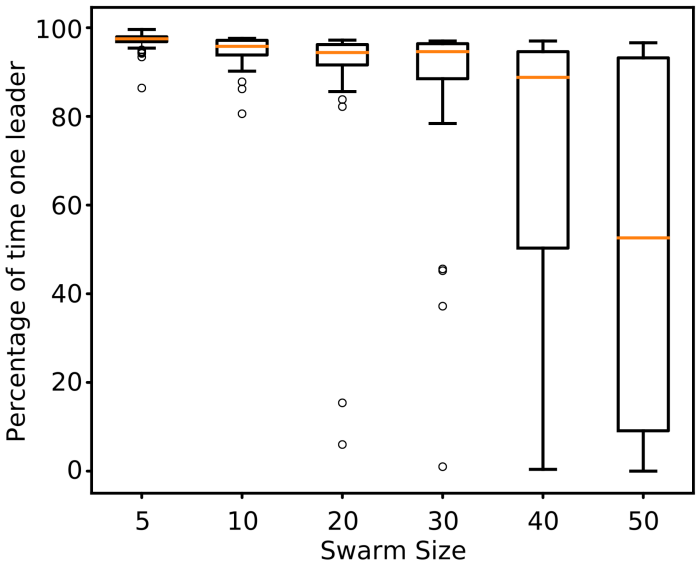


Figure 4.10: Scalability assessment of the leader election primitive. It shows different box plots indicating the percentage of the total simulation time in which just a single robot claims leadership. Each box plot is constructed using data from 50 independent simulations.

leadership, across 50 independent trials. Even though there is not a significant degradation up to swarm sizes of 30, the percentage of time with a unique leader drastically decreases for 40 and 50 robots. Such worsening is not only in terms of median value but also in interquartile range growth. However, in the box plots corresponding to swarm sizes of 40 and 50, it was observed that in more than 96% of the simulations there was a leader elected at some point of the simulation. However, there is a high variability in the time elapsed before leadership allocation.

Robustness

In order to assess the robustness of the robot swarm, any robot that has been the unique leader for fifty consecutive control cycles immediately becomes faulty. More precisely, a faulty robot is no longer able to become the leader of the swarm and, consequently, the rest of agents have to be robust enough in order to overcome such erroneous behavior. Moreover, a defective agent cannot claim leadership nor send its own messages. It can only relay incoming messages from its neighborhood, to preserve graph compactness and avoid isolated robots. Figure 4.11A shows the evolution of the LED actions with faulty robots. Black bars denote activated LEDs while blank spaces mean that the LED is turned off. Red bars indicate that the given robot is faulty during the corresponding time window. It can be observed that robots successfully react against the leader failure, overcoming the undesired scenario by electing a new leader. Specifically, a new negotiation process of short duration is accomplished after the previous leader becomes faulty. It should be noticed that in Figure 4.11A there are a total five robots that become faulty at different time cycles of the simulation. Thus, the robot swarm correctly overcomes robot failure in five consecutive times, demonstrating good robustness.

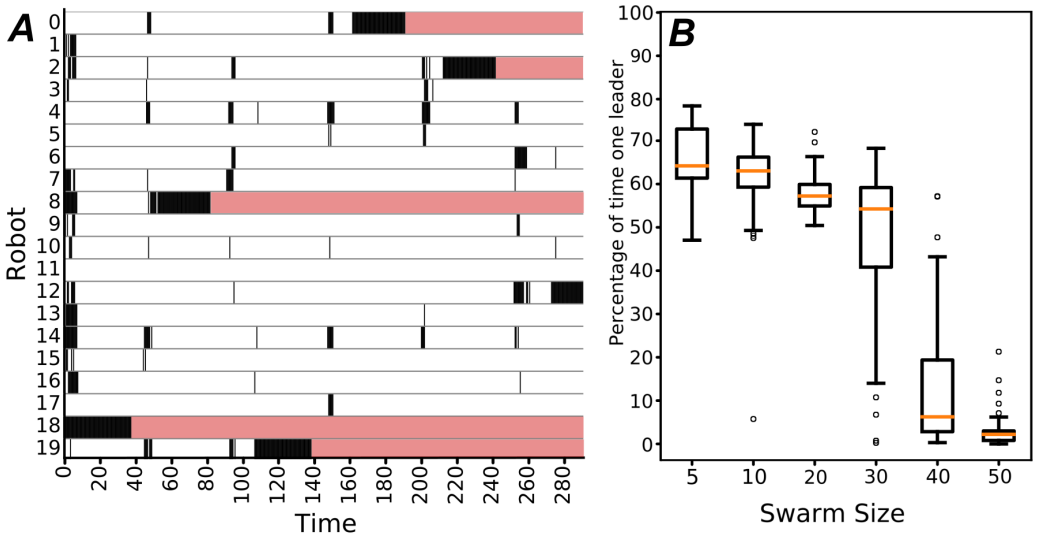


Figure 4.11: Results of the robustness assessment. (**A**) Evolution of the LED actions of the robots in a swarm 20 robots as time elapses. Horizontal black bars denote that the corresponding robot's LED is turned on, and the red bars indicate the given robot is faulty during the corresponding time window. (**B**) Box plot indicating the percentage of the total simulation time with a single leader in simulations with faulty robots. Each box plot is constructed using data from 50 independent simulations.

Figure 4.11B depicts, for multiple swarm sizes, the percentage of the total simulation time with a unique leader, when the robot swarm is subject to the leader failure constrain. It can be observed that there is a overall performance degradation with respect to the scenario without leader failure (see Figure 4.10). However, such performance decrease is unavoidable because of the multiple negotiation phases that are caused by the failure of the leader robots (Figure 4.11A serves as a good example of this issue).

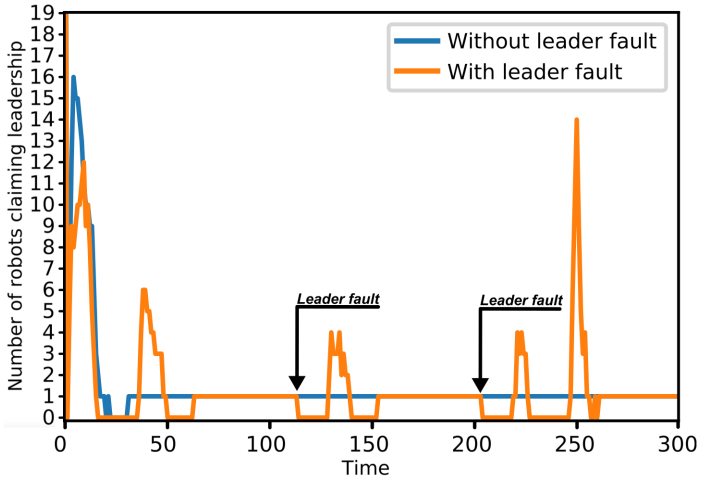


Figure 4.12: Temporal evolution of the number of robots claiming leadership for a trajectory in normal conditions and with leader failure.

Additionally, Figure 4.12 illustrates the temporal evolution of the number of leaders with and without leader failure (in blue and orange, respectively). In the trial without leader fault (under normal conditions), the robot swarm converges to one stable leader at about control cycle 30, which is preserved for the whole duration of the simulation. Alternatively, when leader fault is included the robot swarm is correctly capable to converge to a unique leader (after a negotiation stage) every time that failure is produced.

Communication

To complement the overall behavior, scalability, and robustness of the system, the communication that emerged from the artificial evolution is shown and analyzed. Firstly, a single simulation with 20 robots using the evolved robot controller is studied in detail (Figure 4.13). Moreover, the results displayed in the figure corresponds to the same simulation previously shown in Figure 4.9. Figure 4.13A illustrates the undirected graph that represents the static robot swarm, where the graph nodes correspond to each robot and the graph edges indicate that there is a communication channel between two robots. Notice that the positions of the robots are randomly defined at the beginning of each simulation and, therefore, the depicted graph is only representative in the specific trial under study. Figure 4.13B shows the evolution of the communication state, MODE, of each robot (*relay* in white and *send* in black). A clear correlation between the LED action (a_L) and the communication state (MODE) stands out when Figures 4.13B and 4.9 are compared. Specifically, it can be observed that only those robots that are leaders enter the *send* state, while the others remain in *relay* state. Figure 4.13C details the messages that are transmitted by each robot at each control cycle of the simulation, where the values of each symbol S_i are obtained according to Equation 4.12.

$$S_i = 0.33 \cdot \begin{pmatrix} \mathrm{mod}(i,4) \\ \lfloor i/4 \rfloor \end{pmatrix} \tag{4.12}$$

Provided that Figure 4.13B and C are analyzed jointly, the following communication mechanics can be highlighted:

- If an agent claims leadership, its communication state settles as *send* state. On the contrary, non leader robots enter into *relay* mode.
- Agents in *send* mode mostly emit symbol S_{15}, corresponding to $\mathbf{m} = (1, 1)$.
- The message sent by the potential leaders is spread around the entire swarm, resembling a wave-like propagation. In fact, Figure 4.13A shows that the agents sending symbol S_{15} more frequently are those with less number of hops to the leader (node 7).
- Other symbols apart from S_0 and S_{15} are transmitted only during the negotiation.

In order to validate the previous statements, Figures 4.14 and 4.15 show the results using 50 independent simulations with random initial values. Figure 4.14 shows a boxplot of the percentage of the simulation that there was a unique leader, when different communication variables are inhibited. More precisely, it compares simulations under the following conditions:

(i) normal conditions without inhibition,

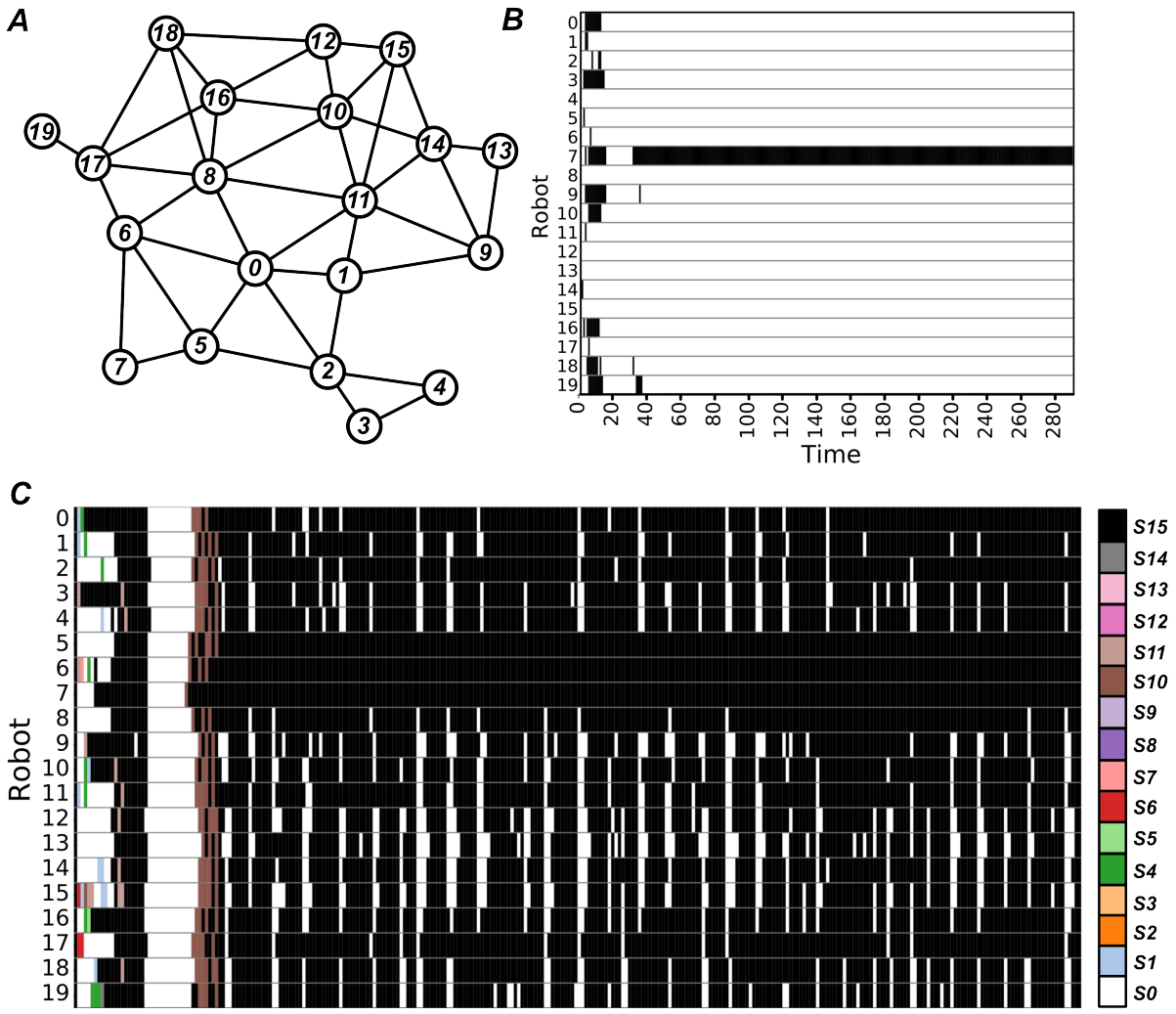


Figure 4.13: Evolved communication in the leader election primitive in a single simulation with 20 robots. (**A**) Spatial graph of the static robot swarm. Nodes represent the robots and edges indicate that two robots can communicate. (**B**) Evolution of the communication state (MODE) of the robots. Horizontal black bars denote that the robot in *send* state, and blank spaces correspond to the *relay* state. (**C**) Evolution of the messages transmitted by each robot at each control cycle of the simulation. The colors indicate the transmitted message S_i, according to the color codes in the rightmost legend and to Equation 4.12.

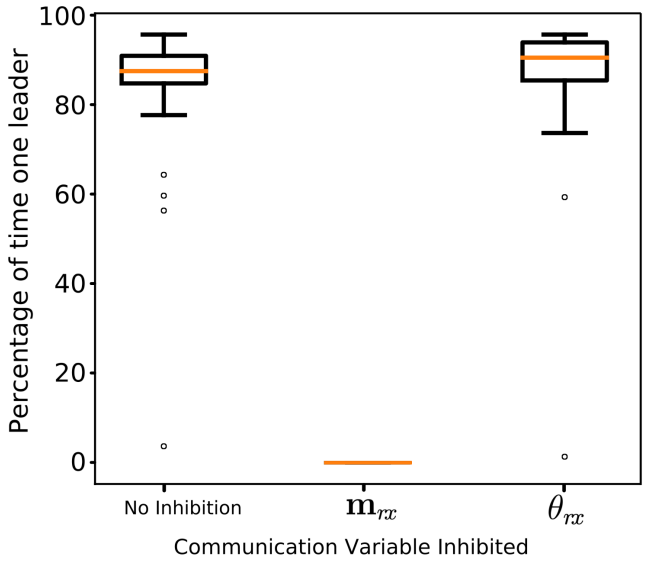


Figure 4.14: Box plot of the percentage of the total simulation time with a single leader in simulations when different variables are inhibited. Each box plot is constructed using data from 50 independent simulations.

(ii) with the received message, $\mathbf{m}_{rx}$, inhibited, and

(iii) with the relative orientation from where the message was received, θ_{rx}, inhibited.

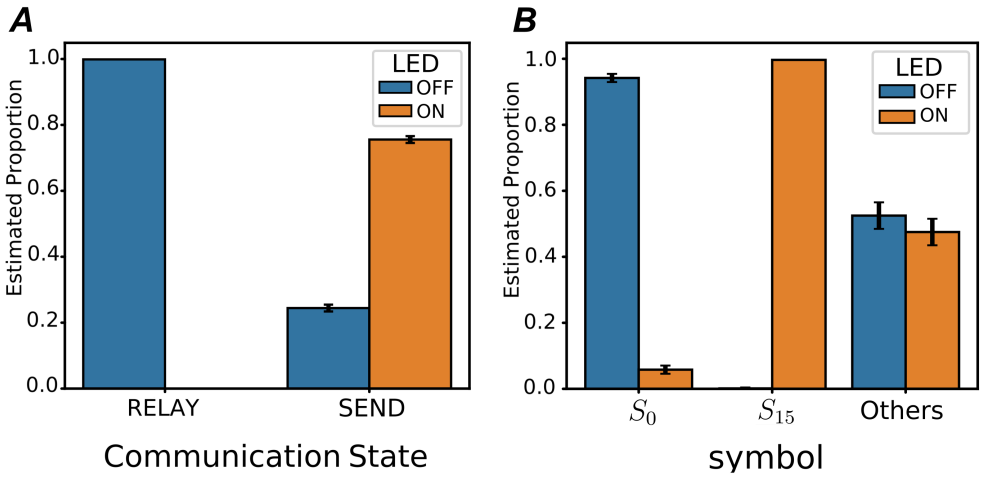


Figure 4.15: Semantics of the transmited message and the communication state in the evolved communication in the leader election. (**A**) Estimated proportion of control cycles that a robot had the LED turned on or off, conditioned to the value of the communication state. (**B**) Estimated proportion of control cycles that a robot had the LED turned on or off, conditioned to the transmitted message.

Inhibition implies that the robot controller (and, thus, the CTRNN), does not have access to the values of the corresponding variable during the whole duration of the simulation.

The results of Figure 4.14, revealed that the content of $\mathbf{m}_{rx}$ plays a crucial role in the emerged

communication, so that its inhibition results a drastic performance decrease. On the contrary, the results also suggest that variable θ_{rx} is not relevant in the emerged communication, because there is no statistically significant difference with the simulation under normal conditions. Moreover, the emerged communication is an *abstract communication* because only the abstract message is communicative.

Finally, Figure 4.15 focuses on the importance of each symbol in the evolved communication, and demonstrates the strong correlation between MODE and a_L. Figure 4.15A illustrates the estimated proportion of control cycles that a robot had its LED turned off or on, while the communication state was either *send* or *relay* (four combinations). It can be clearly observed that the robots that had their LED turned off are always in *relay* state. Besides, the robots that claim leadership (LED turned on) are in *send* state with an estimated probability of 0.752. Figure 4.15B shows the estimated proportion of times that robots transmit each of the available symbols. It was found that the other symbols apart from S_0 and S_{15} are rarely used and, therefore, we grouped them all in the same category. It can be noticed that the symbols transmitted are highly segregated, so that leaders mostly send the symbol S_{15} whereas followers transmit symbol S_0 more frequently.

4.3 Frontier identification

4.3.1 Description

The frontier identification is the problem of detecting the set of robots that are positioned in the perimeter of the swarm topology (the border or frontier). Formally, let $P \in \Re^2$ be a set of points, and $P_{\mathcal{B}}, P_{\mathcal{I}} \subset P$ two subsets of P. Moreover, $P_{\mathcal{B}}$ and $P_{\mathcal{I}}$ form a partition of P, so that $P = P_{\mathcal{B}} \cup P_{\mathcal{I}}$. Consequently, the frontier identification problem is defined as finding the subset of points $P_{\mathcal{B}}$ that creates a closed polygon whose area contains all the points in $P_{\mathcal{I}}$. The sets $P_{\mathcal{B}}$ and $P_{\mathcal{I}}$ are illustrated in Figure 4.16 colored in red and blue, respectively, for a randomly generated set of points P. The points $P_{\mathcal{B}}$ correspond to the positions of the robots in the border of the swarm, whereas the points in $P_{\mathcal{I}}$ are associated to the positions of the robots in the interior part of the group. Moreover, $\mathcal{B}$ and $\mathcal{I}$ are defined as the sets of robots belonging to the border and to the interior, respectively.

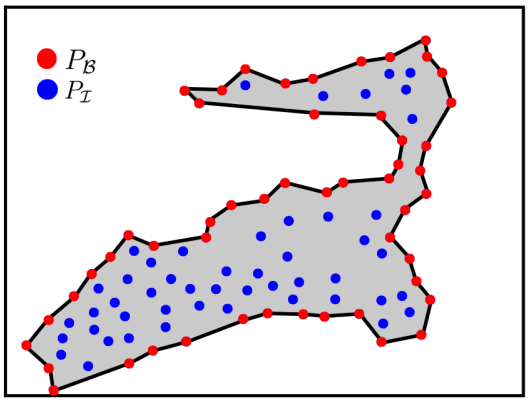


Figure 4.16: Representation of the sets of points $P_{\mathcal{B}}$ and $P_{\mathcal{I}}$ in the frontier identification primitive.

During the execution of the experiment, at each control cycle, each robot individually determines whether it belongs to $\mathcal{B}$ (the border) or to $\mathcal{I}$ (the interior). The task is optimally solved for a given t provided that the following conditions are met:

(a) all the robots in the border of the swarm correctly classify themselves as part of $\mathcal{B}$

(b) all the robots in the interior area correctly classify themselves as part of $\mathcal{I}$.

Therefore, for some t and for a given robot, the task can be seen as a distributed binary classification problem. The robots can classify themselves as border or interior members by respectively turning on or off their LED actuator.

Provided the set of points P, the perimeter points in $P_{\mathcal{B}}$ are computed using the alpha shape algorithm [242], which is a generalization of the convex hull. The alpha shape algorithm is controlled by the parameter α, which effect is depicted in Figure 4.17 for a set of 100 points.

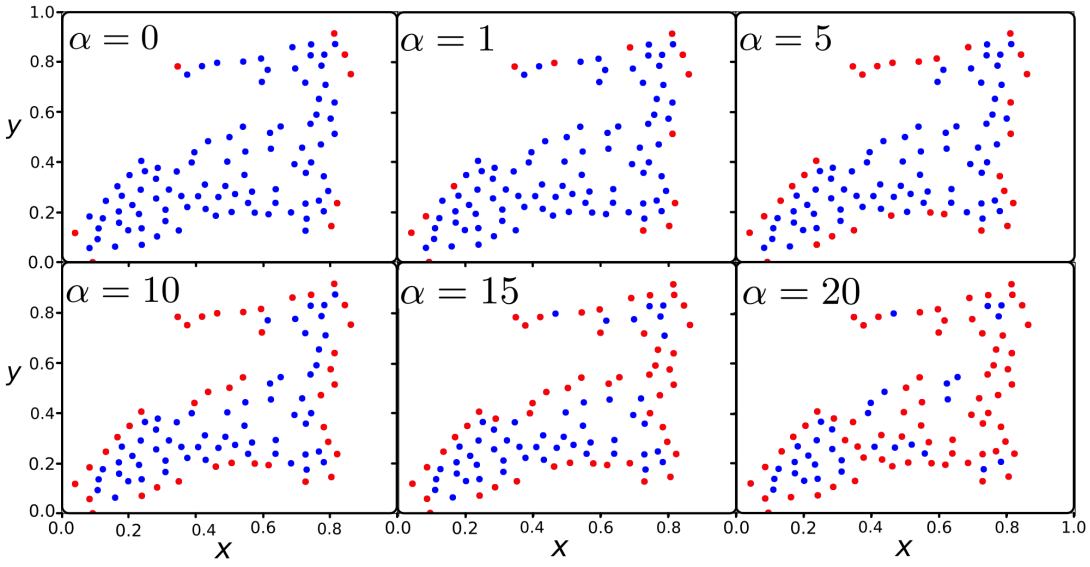


Figure 4.17: Alpha shape of a set of points P for multiple values of the control parameter α.

A value of $\alpha = 0$ generates an alpha shape that is equivalent to the convex hull, which is not sufficiently accurate for the requirements of the frontier identification task. Additionally, an increment of α leads to increasingly more intricate and well defined shapes of the swarm border. However, excessively large values can lead to an overfitted solution, as shown in the case of $\alpha = 20$ in Figure 4.17. After an exhaustive heuristic search with many different sets of points, it was found that an α of 15 is solid and suitable for the swarm robotics task of this section.

The robot swarm used to solve this task is static, and the agents cannot move or rotate. During the artificial evolution process, the positions and orientations of the robots in the swarm are initialized randomly, ensuring that there are not isolated robots in the swarm.

4.3.2 Robot controller

The robot controller used in this experiment is exactly the same CTRNN architecture with two hidden layers (ten neurons each) used in the leader election task (see Figure 4.7). The input stimuli, composed by the received message $\mathbf{m}_{rx}$, the associated value of θ_{rx}, and the current communication state MODE, remains the same as in Section 4.2. Similarly, the output actions correspond to the new message, $\mathbf{a}_{tx}$, the new communication state, and the action that controls the LED actuator, a_L. Although it does not affect the robot controller architecture, it is worth mentioning that the action a_L is no longer devoted to claim leadership, but to classify each robot as either border or interior member.

4.3.3 Fitness function

The fitness function contribution of each robot at each control cycle t is higher as the number of correct classifications, from both border and interior agents, increase. A correct decision implies that either a border member activates its LED (true positive) or that an interior robot turns off its LED (true negative). Equation 4.13 defines the overall swarm fitness function for a given time instant t,

$$f_B(t) = \frac{1}{B \cdot I} \left(\sum_{r \in \mathcal{B}} a_L(t, r) \right) \cdot \left(I - \sum_{r \in \mathcal{I}} a_L(t, r) \right), \tag{4.13}$$

where $\mathcal{B}$ and $\mathcal{I}$ are the sets of robots truly belonging to the border and interior areas (based on the alpha shape algorithm), and B, I correspond to the cardinality of such sets $\mathcal{B}$ and $\mathcal{I}$, respectively. The variable $a_L(t, r) \in \{0, 1\}$ is the action that turns either on or off the LED of a given robot r at some time instant t. The function has two terms that are combined as a product. The first term corresponds to the number of robots in $\mathcal{B}$ that successfully identified themselves as border members, divided by the total number of robots that are actually in the border. Alternatively, the second part is the number of correct classifications of agents in $\mathcal{I}$, divided by the total number of robots that are actually in the interior area.

The total fitness function of an entire episode of duration $T\Delta t$ seconds is shown in Equation 4.14,

$$FF_B = \frac{1}{T} \sum_{k=0}^{T} f_B(k\Delta t), \tag{4.14}$$

as the temporal average of the instantaneous function f_B at each control cycle $t = k\Delta t$. In this experiment we fixed Δt as 50 ms and T as 500 cycles. Finally, each individual is evaluated in five independent trials, so that multiple fitness samples are used to construct the final expected fitness estimate. The final estimate is the sample mean of the fitness values obtained in each trial.

4.3.4 Results

Behavior and overall performance

Figure 4.18 shows the behavior of the robots in a single simulation with a swarm of 30 robots. Figure 4.18A, sketches the spatial graph of the swarm, showing the ground truth LED values

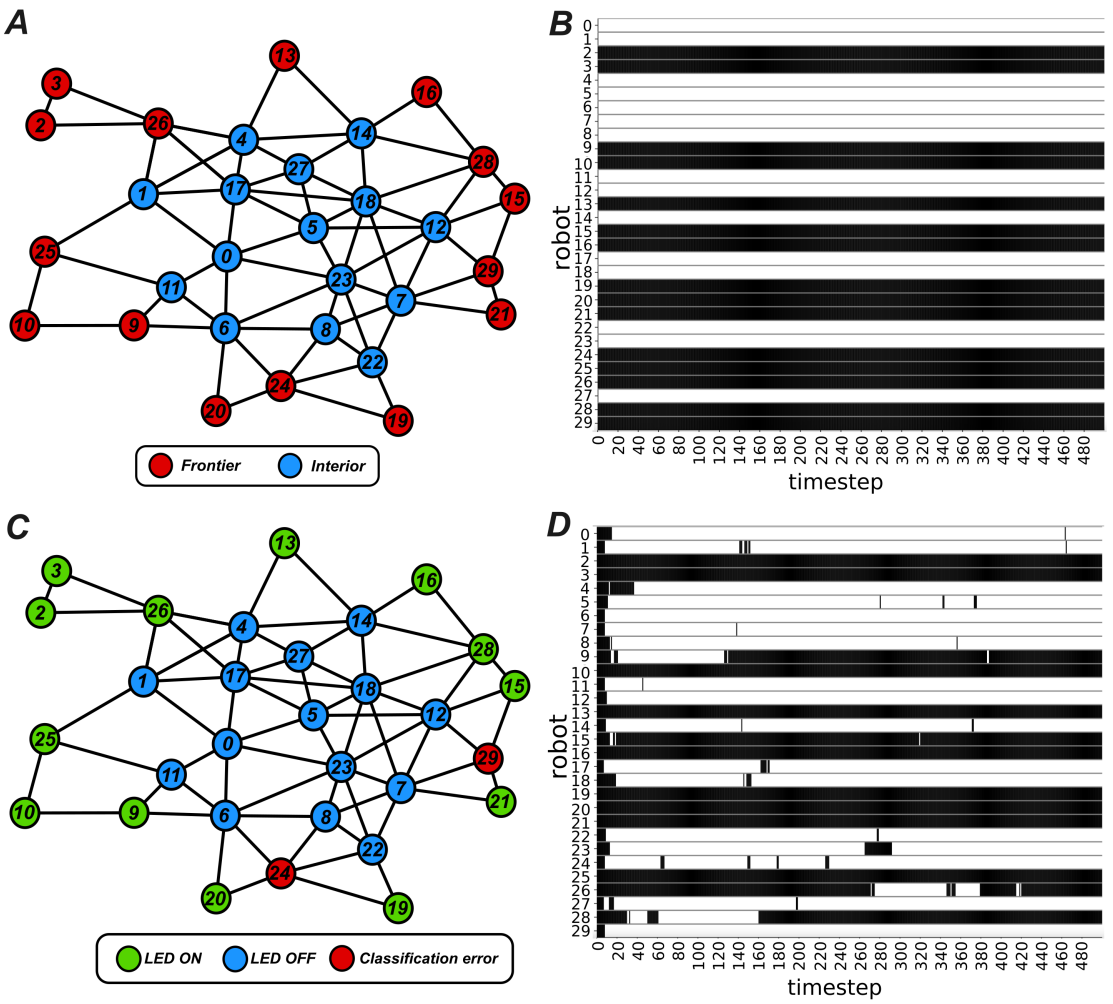


Figure 4.18: Results of a single simulation of the border identification primitive. (**A**) Spatial graph of the swarm showing the ground truth actions according to the alpha shape algorithm. (**B**) Ground truth actions depicted across simulation time (to be compared to the actual actions in (**D**)). (**C**) Spatial graph of the swarm showing the actual actions of the robots at the end of the simulation. Red nodes are erroneous classifications. (**D**) Evolution of the LED actions of the robots. Black bars correspond to LEDs turned on, while blank spaces are LEDs turned off.

that the robots should take according to the alpha shape algorithm (using $\alpha = 15$). Red circles indicate that the corresponding robot is in the perimeter of the group, whereas the blue circles are the robots in the interior area. Complementarily, Figure 4.18C highlights the actual LED actions of the robots at the last time step of the trial. It stands out the LEDs that are turned on in green and the LEDs that are deactivated in blue. Moreover, errors in the robot classifications, corresponding to robots 24 and 29, are highlighted in red. These two robots classified themselves as interior points while they belong to the border of the swarm according to the alpha shape algorithm (see Figure 4.18A). Nonetheless, observing the position of robots 24 and 29, one can notice that the decision is not as clear as in other nodes such as 10, 20, or 19. Figure 4.18B shows the target LED actions that the robots should

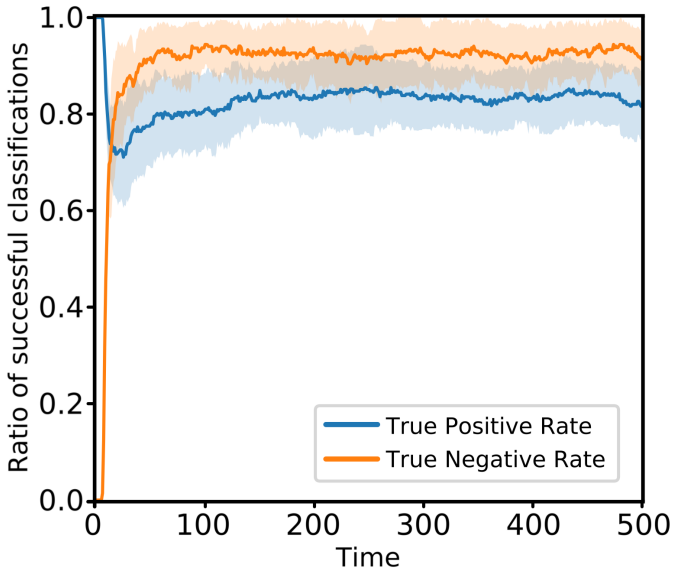


Figure 4.19: Evolution of the True Positive Rate (TPR) and the True Negative Rate (TNR) during the frontier identification primitive. In both cases, samples of fifty independent simulations are used to compute the evolution of the median values (darker curves) and the first and third quartiles (boundaries of the light contours).

perform during the entire simulation, according to the alpha shape algorithm. In contrast, Figure 4.18D illustrates the time evolution of the actual LED actions of each robot. The results show good classification accuracy, although, sporadically, there are some robots with wrong actions. Specifically, the classification accuracy oscillates between 86.6%, for instants near time step 80, and 93.3%, for the last control cycles of the simulation. It can be observed that there are some agents whose LED actions are remarkably stable (e.g. robots 2, 3 or 10) while the decisions of other robots are much more undefined (e.g. nodes 9, 26 or 28). A common observable feature is that agents with a robust decision generally have only few neighbors (one or two). On the contrary, undecided robots mainly belong to a dense part of the graph, with many neighbors (three, four or five). Thereafter, the errors principally occur when the agents have many neighboring robots, leading to the naive classification of being interior node (which is an incorrect assumption in some cases).

The problem is treated as a multi-agent binary classification problem and, therefore, Figure 4.19 plots the temporal evolution of the True Positive Rate (TPR) and the True Negative Rate (TNR), considering the robots of 50 independent simulations.

The TPR and the TNR are computed according to Equations 4.15a and 4.15b,

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \qquad (4.15a) \qquad\qquad \mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{FP} + \mathrm{TN}}, \qquad (4.15b)$$

where TP, FN, FP, and TN are the true positives, false negatives, false positives, and true negatives, respectively. Figure 4.19 reveals that errors are more likely to occur as false negatives

(predicted interior but it is actually frontier). Notice that the initial values of the TPR and the TNR are one and zero, respectively, because all the robots initially identify themselves as robots in the border (see Figure 4.18D). Subsequently, the robot swarm progressively converges to the stable values of TPR and TNR of about 0.85 and 0.92.

Scalability

Figure 4.20 shows the scalability of the robot swarm in the frontier identification task. It depicts the temporal evolution of the accuracy for different swarm sizes, each composed by 50 independent simulations. The accuracy is computed as in Equation 4.16,

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{R}, \tag{4.16}$$

being R the number of robots in the swarm.

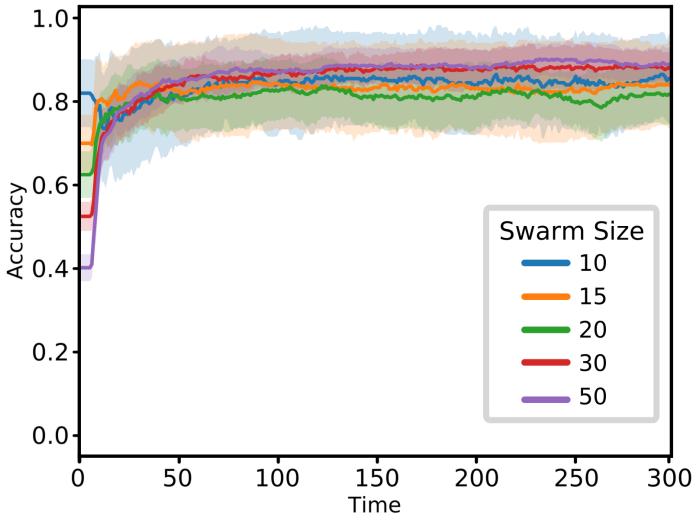


Figure 4.20: Scalability assessment in the frontier identification task. It shows the temporal evolution of the accuracy for different swarm sizes. The darker curves represent dependent median value of the accuracy, and the shadow areas indicate the first and third quartiles of the accuracy. Fifty independent simulations are used for each swarm size.

Figure 4.20 shows that, upon convergence, there is not a statistically significant difference between the results for each swarm size, suggesting that system is scalable up to 50 robots.

Robustness

In order to assess the robustness of the system, the topology of the robot swarm is changed every 200 control cycles. This perturbation causes an abrupt change in the swarm distribution, so that the robots are forced to reconsider whether they are in the border of the swarm.

Figure 4.21 shows both ground truth LED actions according alpha shape (A), and the actual LED actions of the robots (B) during a single simulation. Even though there are some errors, the robots are able to solidly react to the topology changes.

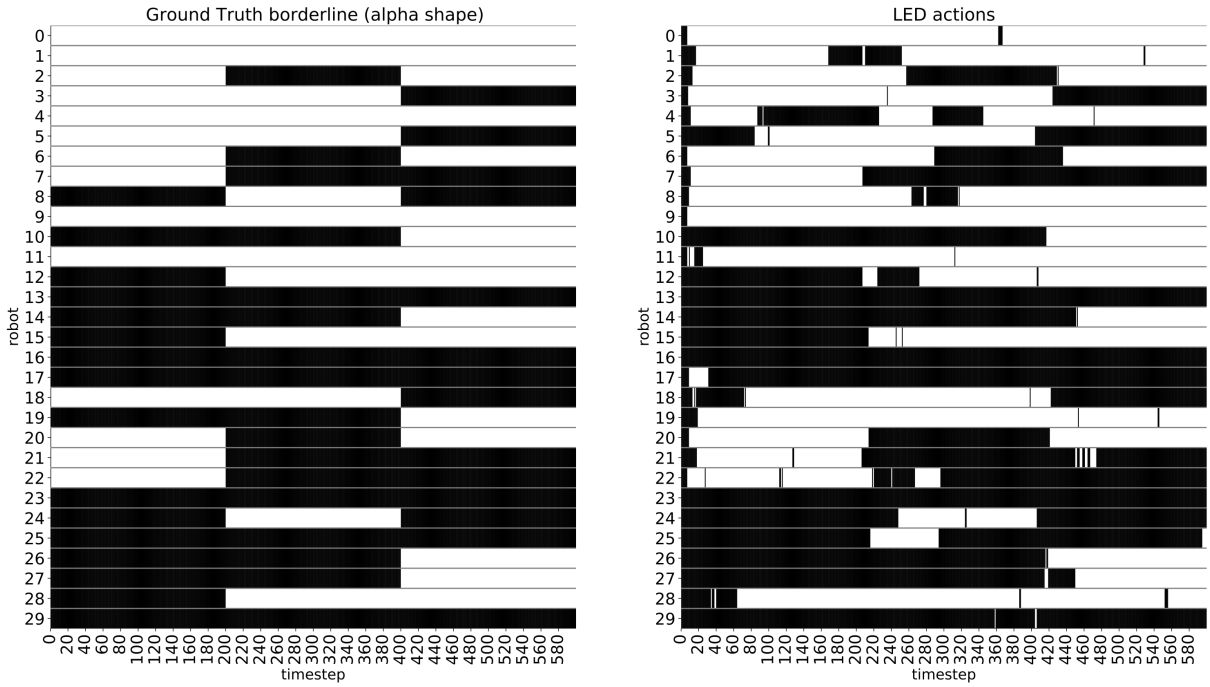


Figure 4.21: Robustness assessment in the frontier identification task. The topology of the robot swarm is drastically changed every 200 control cycles. (**A**) Ground truth LED actions according to the alpha shape algorithm. (**B**) Evolution of the actual LED actions of the robots. Black bars indicate that the LED is turned on and blank spaces represent LEDs turned off.

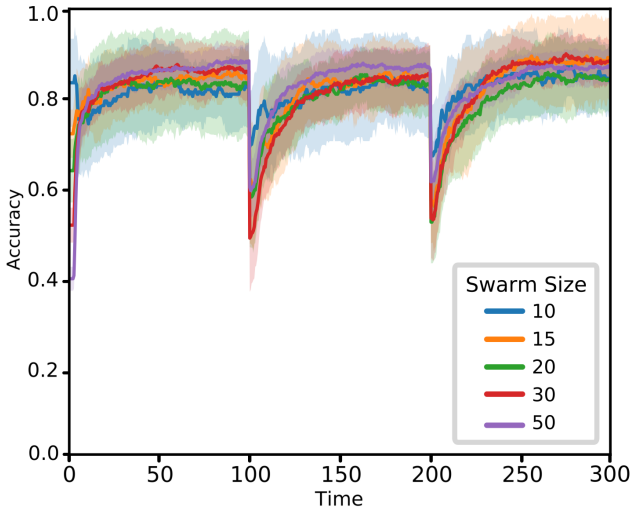


Figure 4.22: Robustness assessment of the frontier identification task. It shows the temporal evolution of the accuracy for different swarm sizes. Every 200 control cycles the topology of the robot swarm is changed. The darker curves represent the median value of the accuracy, and the shadow areas indicate the first and third quartiles of the accuracy. Fifty independent simulations are used for each swarm size.

Additionally, Figure 4.22 illustrates the temporal evolution of the accuracy metric under the periodic changes of the swarm topology. It can be observed that the robots react and overcome successfully two abrupt swarm topology changes. The figure, moreover, shows the accuracy results of multiple swarm sizes, demonstrating both scalability and robustness at the same time.

Communication

With the aim of studying the communication that emerges as a result of evolution, Figure 4.23 analyzes the importance of the communication variables in solving the task. Specifically, it shows a comparison of the evolution of the accuracy metric among a scenario with no inhibition (normal conditions) and when different communication variables, $\mathbf{m}_{rx}$ and θ_{rx}, are inhibited. Each inhibition is performed independently, and implies that the robot controller no longer has access to the corresponding input.

The comparison of the figure reveals that θ_{rx} is a highly relevant context variable, whose deletion causes system breakdown (drastic accuracy drop). On the contrary, the figure does not reflect a significant accuracy decrease when the message $\mathbf{m}_{rx}$ is inhibited. Therefore, this finding indicates that the message is not a highly relevant communication variable to solve this task. This leads to a solution to the task with purely *situated communication*, because only the context underlying the message (θ_{rx} specifically) is communicative. The communication state was not included in this analysis because it was found that the CTRNN of the robots always fix its value to *send* state.

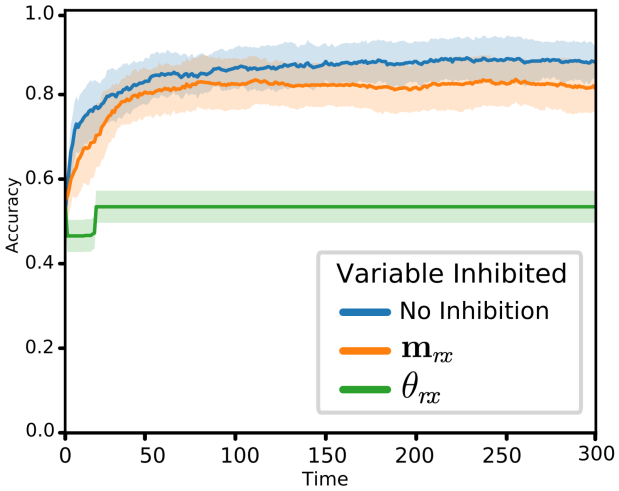


Figure 4.23: Evolution of the accuracy metric when different variables are inhibited in the frontier identification task.

4.4 Heading orientation consensus

4.4.1 Description

The heading orientation consensus is a SR task which aim is that all the robots in the swarm must point to the same direction. Consequently, an optimal solution to the problem implies that the heading orientation ($\theta_r(t)$) of every robot r would converge to the same value. In this experiment, the robots cannot alter their positions $\mathbf{x}_r$. Nonetheless, to achieve the sought solution, the robots can alter their heading orientations by means of either clockwise or counterclockwise rotation movements. Both magnitude and sense of the rotation are set by the robot controller through the action a_ω, that takes values between -1 and 1. An important remark is that robots cannot acquire absolute sensing references, such a light source or a compass, to solve the task. Instead, the agents can only infer the current relative orientation of their neighbors using the communication system in Section 4.1.2. Figure 4.24 illustrates a

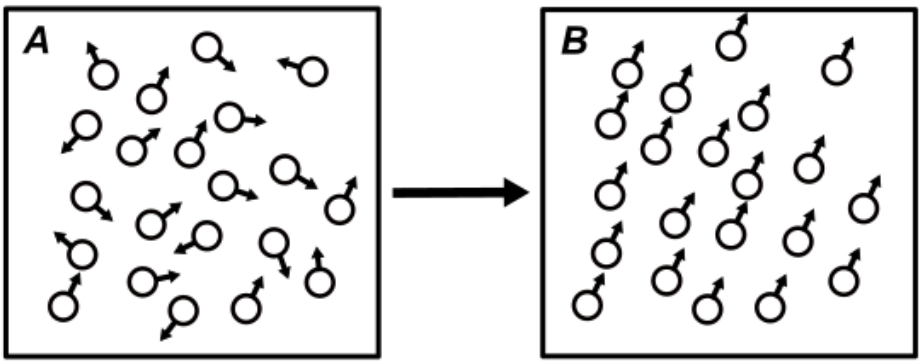


Figure 4.24: Illustrative example of the heading orientation consensus task. (**A**) Initial conditions, and (**B**) desired behavior.

hypothetical example of the heading orientation consensus task. It shows in Figure 4.24A some initial conditions where the heading orientations of the robots are highly different, and in Figure 4.24B the desired behavior in which all robots have approximately the same value of θ_r.

4.4.2 Robot controller

The robot controller that defines the behavior of the agents in this experiment is also based on a CTRNN model, that directly maps sensor readings to actuator actions. Specifically, Figure 4.25 shows the diagram of the CTRNN architecture. As in previous tasks, there are two hidden layers with ten hidden neurons each. The input layer, with seven nodes in green, is placed at the leftmost side of the diagram. It is fed with the information received and processed by the communication system (see Section 4.1.2), which are: (i) the received two-dimensional message, $\mathbf{m}_{rx}$, (ii) the current state of the MODE state, and (iii) the relative orientations θ_{tx} and θ_{rx} from where the message was transmitted and received (see Figure 4.4). The output layer, with four neurons in red, is responsible for generating the following actions:

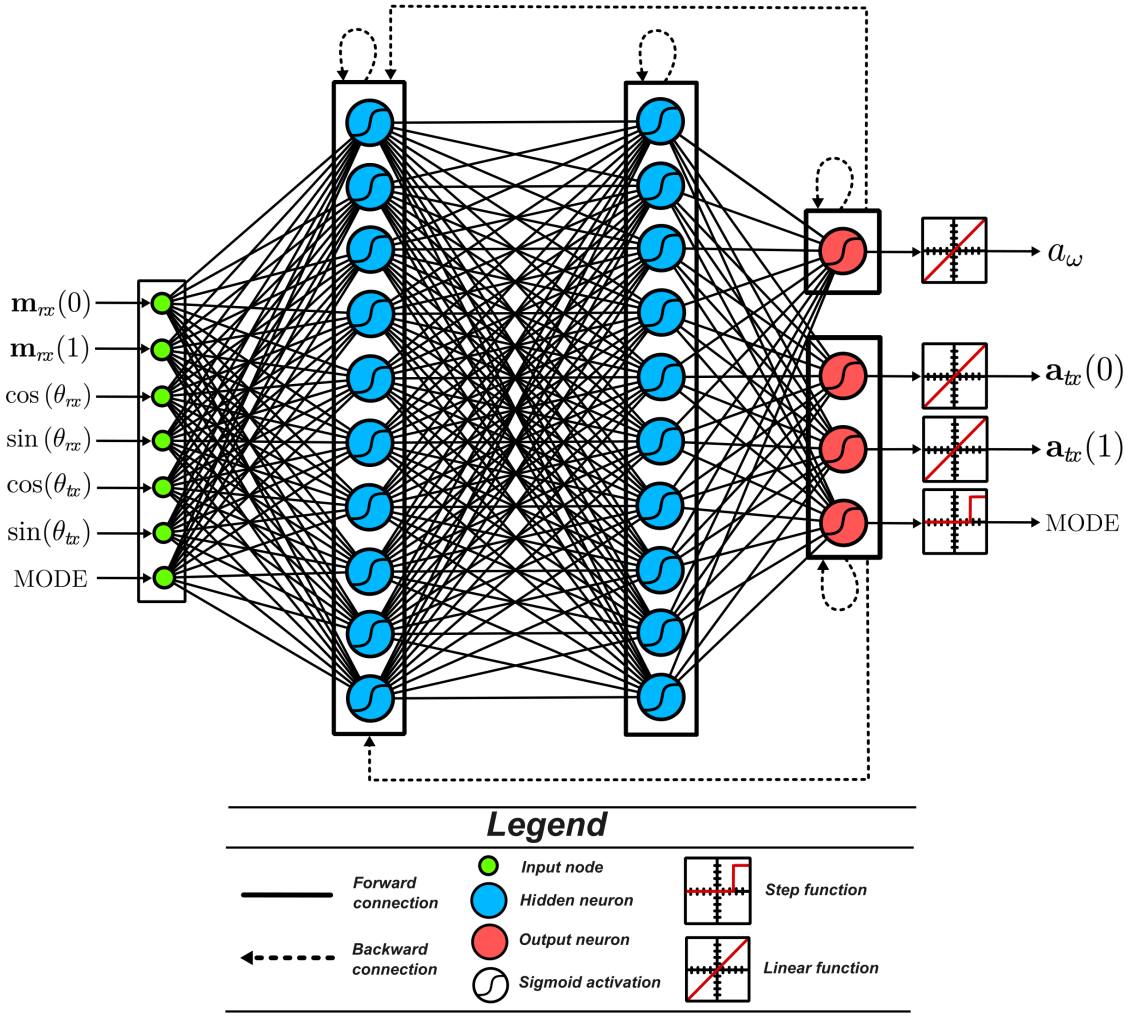


Figure 4.25: Artificial neural network architecture used in the heading orientation consensus primitive.

- The next value of the communication state MODE. The neuron output is injected to a step function to constrain its value to either zero or one.
- The two-dimensional message $\mathbf{a}_{tx}$ that will be broadcasted to the neighborhood, provided that the condition MODE $= 1$ is met.
- The rotation speed and sense of the robot, a_w. A hyperbolic tangent function is applied to the neuron activation to transform its values to $[-1, 1]$.

4.4.3 Fitness function

For solving the heading orientation consensus task, a fitness function composed by two terms is proposed. These two terms account for different aspects of the task and are aggregated as

a product. Equation 4.17 defines the fitness function at a given time instant t,

$$f_C(t) = \frac{1}{R} \cdot \sum_{r=1}^{R} \left(1 - \frac{\min\{2\pi - |\theta_r(t) - \overline{\theta}(t)|, |\theta_r(t) - \overline{\theta}(t)|\}}{\pi}\right) \cdot \left(1 - |a_\omega(t, r)|\right), \quad (4.17)$$

where R is the number of robots, θ_r is the heading orientation of some robot r, and $a_\omega(t, r) \in [-1, 1]$ is the action that controls the rotation speed of the robot r at time t. The first term of f_C considers the deviation or misalignment of the heading orientations of each robot, relative to the mean orientation of the whole swarm, $\overline{\theta}$ (defined in Equation 4.18).

$$\overline{\theta}(t) = \arg\left(\sum_{r=1}^{R} e^{j\theta_r(t)}\right) \quad (4.18)$$

The first term in Equation 4.17 grows linearly as the heading orientation of the robot r approaches the value of $\overline{\theta}$. Similarly, it decreases when such deviation with respect to $\overline{\theta}$ increases. The optimal scenario of the first part of f_C, in which the orientation consensus is reached, is achieved when $\theta_r = \overline{\theta}, \forall r$. In contrast, the second term in Equation 4.17 rewards the robots for reducing their rotation speed, a_ω. The reason behind including such second term of f_C is to avoid that the robots rotate with the same phase and at a constant angular speed once the heading orientation matching is fulfilled.

The total fitness function of an entire episode of duration $T\Delta t$ seconds is shown in Equation 4.19,

$$FF_C = \frac{1}{T} \sum_{k=0}^{T} f_C(k\Delta t), \quad (4.19)$$

as the temporal average of the instantaneous function f_C at each control cycle $t = k\Delta t$. In this experiment we fixed Δt as 50 ms and T as 1000 control cycles. Finally, each individual is evaluated in five independent trials, so that multiple fitness samples are used to construct the final expected fitness estimate. The final estimate is the sample mean of the fitness values obtained in each trial.

4.4.4 Results

Behavior and overall performance

Figure 4.26 shows six frames of at different control cycles of a simulation, using the best performing individual. The blue circles represent the positions of the robots, and the red arrows indicate their heading orientations. It can be observed how the robot swarm achieves the orientation consensus in about 100 simulation cycles. Moreover, for time steps greater than 100, the heading orientation consensus is successfully preserved, although there are some small variations in the converged value.

Figure 4.27 represents the time evolution of the heading orientations of the robots in a simulation with 20 agents. The first 100 time instants comprise a transient period before the robot swarm converges to a common heading orientation. After the transient period, the robots successfully achieve the consensus up to a certain degree. Nonetheless, it can be

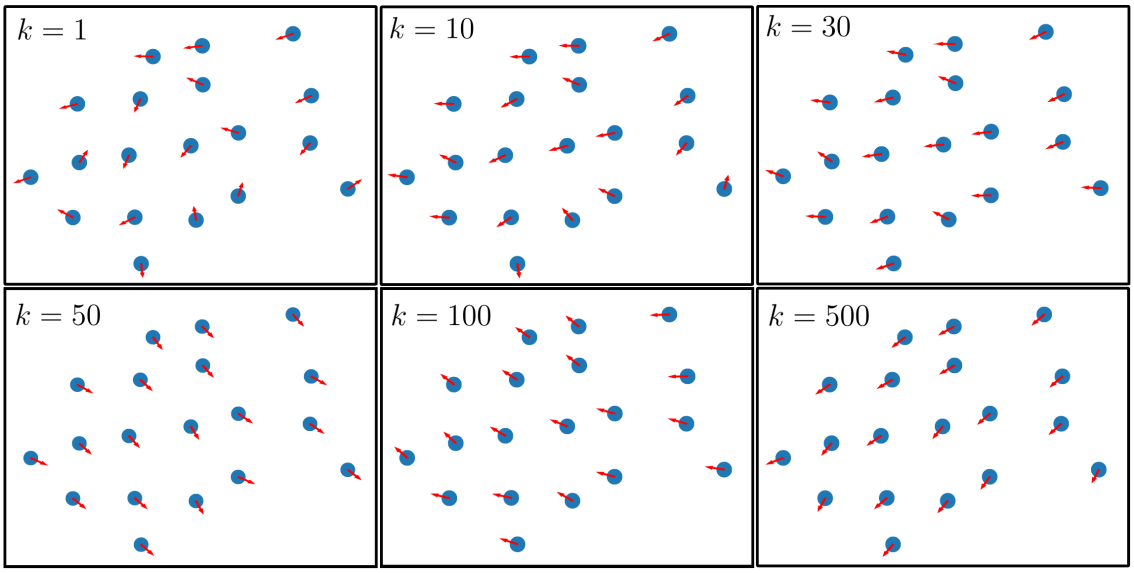


Figure 4.26: Snapshots of different control cycles during a simulation of the orientation consensus experiment. Each circle represents a robot and the arrows indicate the heading orientations. The variable k indicates the time step of the corresponding frame.

noticed that there is a very slight rotation to maintain the orientation consensus along time. In the figure, this residual rotation can be appreciated in the slope of the bean of curves, which is only about 0.01 radians per control cycle.

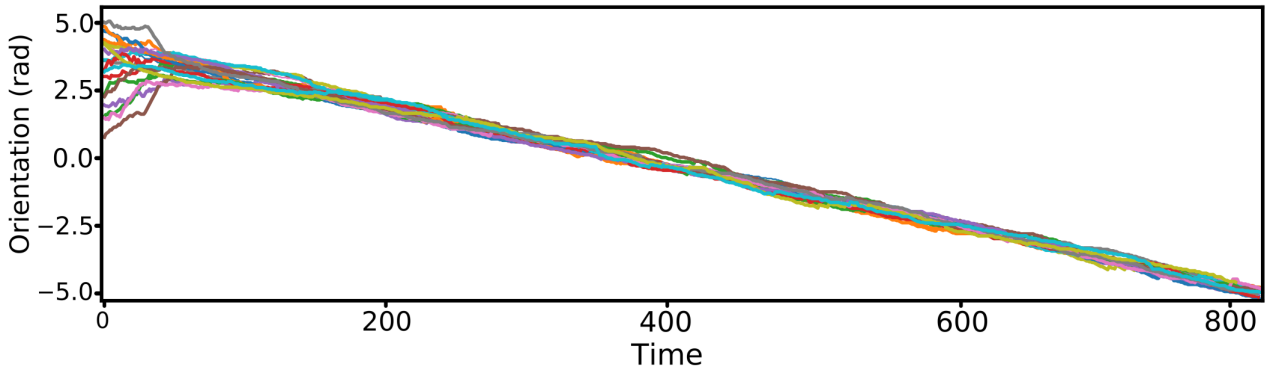


Figure 4.27: Evolution of the heading orientation of the robots during a simulation with 20 robots. Each curve corresponds to the orientation of one robot. The orientation range $[0, 2\pi)$ is extended to the real set for visualization purposes.

Scalability

To assess the scalability of the system in this task, the misalignment metric in Equation 4.20 is used,

$$M_\theta(t) = \frac{1}{R} \sum_{r \in \mathcal{R}} \min \left\{ |\theta_r(t) - \overline{\theta}(t)|, 2\pi - |\theta_r(t) - \overline{\theta}(t)| \right\}. \tag{4.20}$$

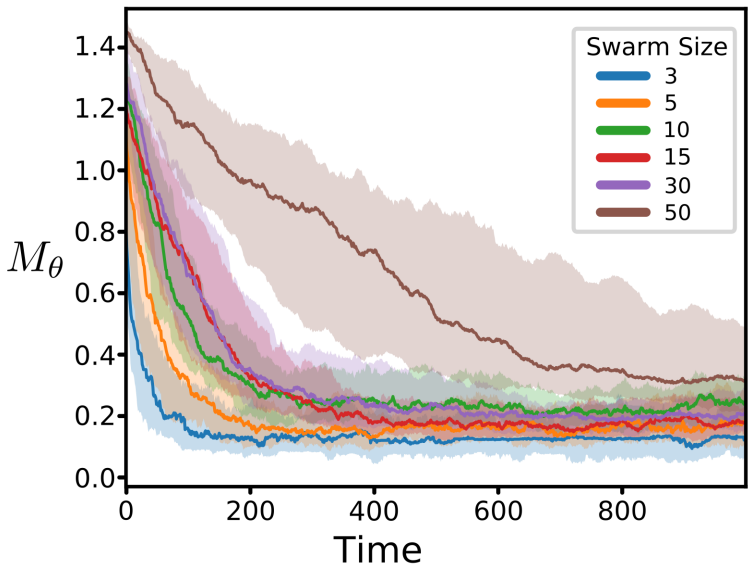


Figure 4.28: Scalability assessment in the orientation consensus primitive. Evolution of the misalignment metric (M_θ) for different swarm sizes. The darker curves represent the median value of M_θ, and the boundaries of the shadow areas indicate the first and third quartiles. Fifty independent simulations are used for each swarm size.

It computes the mean deviation of the heading orientation of each robot in relation to the mean orientation of the swarm. The optimal value of M_θ is zero, and it is met when the orientation consensus is reached. The mean orientation $\overline{\theta}$ was previously defined in Equation 4.18.

Figure 4.28 shows the temporal evolution of the M_θ as simulation time elapses. The results are illustrated for multiple swarm sizes, and each curve shows the median value (darker curves) and the first and third quartiles (upper and lower shadow contours) of the misalignment metric using 50 independent simulations. The growth in the number of robots results in a degradation of both the time to convergence and the steady state value of the M_θ. Nonetheless, even for large swarm sizes, the robot swarm is capable of reaching the orientation consensus. Therefore, these results demonstrate good scalability of the evolved system.

Robustness

The unexpected perturbation introduced during the evaluation simulation, with the aim of testing the robustness, is the following. At control cycle 200, a subset of the robots becomes *uncontrollable*, implying that their wheels are no longer controlled by the robots' CTRNN. Specifically, *uncontrollable* agents rotate at constant angular velocity until a precise orientation is reached. All the *uncontrollable* robots tend to the same orientation value, which is randomly chosen. It is, furthermore, imposed that at time step 600, the goal orientation of the *uncontrollable* robots is switched to a different one (also randomly sampled).

Figure 4.29 illustrates the evolution of the orientation of all the robots during a single simulation that implements the previously described mechanic. In the depicted trial there are four *uncontrollable* robots in a swarm of ten robots. The corresponding robots become *uncontrollable* at control cycle 200, and switch their goal orientation at control cycle 600. The

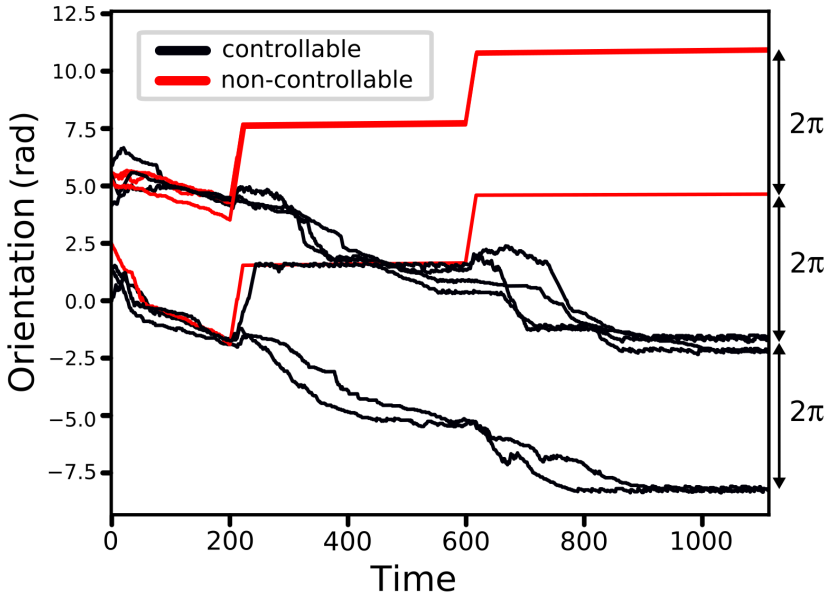


Figure 4.29: Results of the robustness analysis in the orientation consensus primitive. It shows the evolution of the orientations provided that four of the robots are *uncontrollable* since control cycle 200. The orientations are represented in radians in the range $(-\infty, \infty)$ in order to simplify the figure.

orientations are represented in radians in the range $(-\infty, \infty)$ in order to simplify the figure, so that any pair of values separated by 2π radians correspond to the same orientation. The figure reveals that the robots successfully react to the drastic orientation switch of the leaders, progressively converging to the target orientation of the *uncontrollable* agents.

Communication

In order to analyze the evolved communication, Figure 4.30A illustrates the temporal evolution of M_θ in four different scenarios with different communication variables inhibited. Specifically, it compares simulations with the inhibition of the received message ($\mathbf{m}_{rx}$) in orange, the reception and transmission relative orientations (θ_{rx} and θ_{tx}) in green and red, and without inhibition (normal conditions). The variable MODE is not included in the analysis because it was found that it always remains in the *send* state. The results in the figure show that the variables θ_{rx} and θ_{tx} are essential for correctly solving the task. In contrast, simulations with the inhibition of the message $\mathbf{m}_{rx}$ lead to a successful minimization of M_θ, equivalent to the one under normal conditions. Thus, the results suggest that the message itself is not relevant for reaching the orientation consensus.

However, complementary results that highly contrast with the previous statement are shown in Figures 4.30B and C. In Figure 4.30C, the evolution across time of the heading orientations of the robots is compared when, (i) no variable inhibition is produced, and (ii) when the received message ($\mathbf{m}_{rx}$) is inhibited. Although the restriction of the message to the agents is not relevant for reaching the consensus value, it is clearly used for the reduction of the rotation speed of the robots when consensus is reached. This property is not reflected

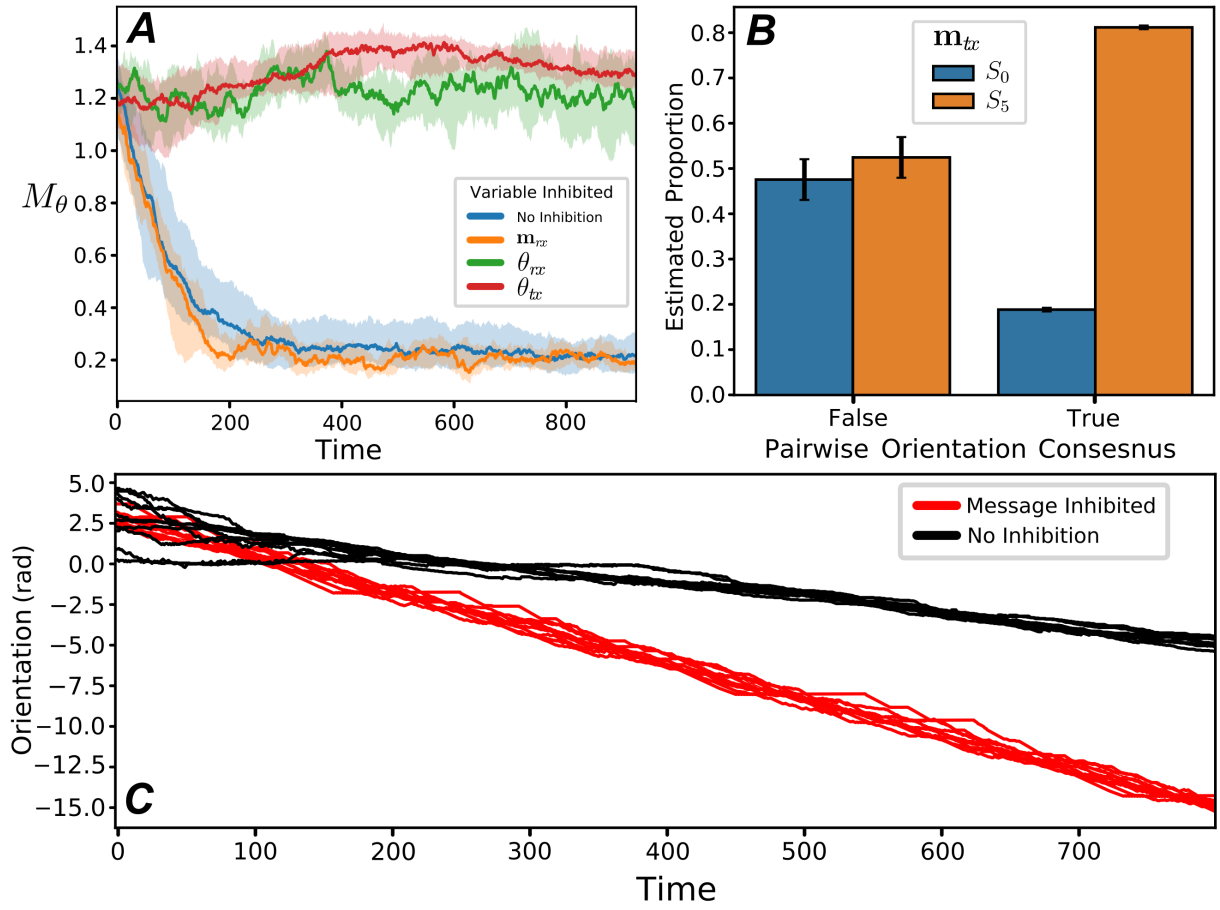


Figure 4.30: Analysis of the evolved communication in the orientation consensus primitive. (**A**) Evolution of M_θ when different communication variables are inhibited. The darker curves represent the corresponding median value of M_θ, and the contours of the lighter shadows are the first and third quartiles. Each curve is built using data from fifty independent simulations. (**B**) Proportion estimates and 95 % confidence intervals of the times each symbol is transmitted conditioned to the status of pairwise consensus. (**C**) Evolution of the heading orientations of two separate robot swarms in two independent simulations. The black curves are the orientations of the robots in a simulation under normal conditions, while the red curves correspond to the swarm of a simulation with the received message inhibited.

in the misalignment metric and, thus, Figure 4.30A incorrectly categorizes $\mathbf{m}_{rx}$ as a non-communicative signal. Additionally, Figure 4.30B deepens into the semantics of the symbols. The figure shows the estimate values of the proportion of times that robots send each symbol when the orientation consensus between the sender and listener robots is achieved. The estimates are computed using the results from 50 independent trials. Only the messages $S_0 = (0, 0)$ and $S_5 = (0.33, 0.33)$ are included because the remaining 14 symbols are never used in the evolved communication. The symbol 0.33 is mainly used once the heading orientation consensus is achieved between two communicating robots. In contrast, in the complementary scenario in which the robots are not aligned yet there is not a statistically significant difference in the proportion of times that each symbol is generated. This observation verifies the previous

hypothesis about the importance of the message once the consensus is reached.

4.5 Summary and discussion

In this Chapter, three different primitive SR behaviors (leader election, frontier identification, and orientation consensus) were separately evolved based on the same robot controller and communication system. These primitive behaviors are the basis to create increasingly more complex and advanced robot controllers, through composition and incremental evolution.

A minimal and highly constrained communication system was presented and used in all of the primitive behaviors. Such communication system defines the rules and mechanics of the communication, but the semantics emerged as a result of the artificial evolution. Some sort of communication emerged at the end of the artificial evolution in all the primitive tasks. However, the evolved communication semantics greatly diverged depending on the resulting behavior, even though all the evolutionary processes landed from an equivalent starting point. Specifically, the following communication semantics were evolved in each of the tackled primitives:

(1) **Leader election**: an *abstract communication*, in which only the message is communicative, emerged in the leader election.

(2) **Frontier identification**: a *purely situated communication*, in which the message was not relevant and all the communicative value lied on the associated environmental context, was evolved in the frontier identificiation primitive.

(3) **Heading orientation consensus**: a *situated communication*, as a hybrid scenario in which both the abstract message and the associated context are relevant in the communication, emerged in the frontier identification primitive.

In addition to the analysis of the emerged communication, all the evolved behaviors are subject to scalability and robustness assessments, which are two essential properties of SR systems. The results revealed that the optimized robot controllers achieve good scalability in all the tasks, up to swarm sizes of 50 agents. Additionally, the robot swarms are also robust against strong perturbations that highly alter the converged solution, imposing a complete reorganization.

The divergences in the evolved communication shed some light on one of the main problems of communication systems in SR, the lack of a standarized framework. Many SR studies, in which the emerged communication is the main focus, are specifically designed *ad hoc* for the task or problem studied [11]. Although different communications emerged from the same initial setup in this chapter, the robot controller of each task still required some *ad hoc* variations in the input and ouput spaces, and a complete redefinition of the fitness function. Moreover, the system was evolved separately and independently to obtain each of the goal behaviors. Therefore, the experiments of this chapter revealed that it is possible to obtain SR and communication systems with *weak transferability*, requiring minor design modification among the different tasks. Ideally, it would be desirable to obtain an evolutionary framework that can be applied to multiple distinguishable tasks without the need to re-evolve the robot

controllers and the communication (*strong transferability*).

Chapter 5

Evolution of transferable and self-organized communication modules

In the previous chapter, it was shown that diverse communication semantics can emerge when the same neuro-controller, that uses the same communication system, is evolved in different primitive tasks. However, the communication mechanics and semantics were specifically evolved ad-hoc for each SR task. That is, the SR system used in the experiments of the previous chapter showed *weak transferability*, because although the same system can be applied to multiple tasks with minor design modifications, a complete re-evolution of the system was required for each experiment. This Chapter goes a step beyond and tackles the *strong transferability* in SR, with the aim of using the exact same evolved communications in multiple tasks. As stated in [11], direct communication semantics and rules are generally designed and constructed for solving specific problems. This represents a significant challenge in the SR field, as there is a lack of standardized communication frameworks or methods that can be implemented across a wide variety of problems. This important concern is also discussed in [12], highlighting the relation between the lack of standardization and real-world applicability in the swarm robotics field. Additionally, a highly relevant line of research that emerges from Chapter 4 concerns how basic primitive behaviors can be combined and reused to produce more complex collective behaviors.

These issues are attacked in this chapter using a self-organized and decentralized communication module, whose logic is evolved only once and can be transferred to tackle multiple independent SR tasks and with varying swarm sizes. Its performance is demonstrated using the exact same evolved communication in three well-known SR problems, namely, (i) aggregation of the robots into groups of desired sizes, (ii) formation control, and (iii) a basic foraging task involving dynamic role allocation. It is shown that these distinct tasks can be successfully solved using the proposed system. The communication module relies on a virtual state space, namely *the communication space*, which is constructed so that each robot is viewed as a point or particle within this space (in addition to and independently of their coordinates in the physical world). The coordinates of the agents within the communication space are defined as the *communication state*. The communication space is partitioned into a set of *virtual regions*, and each region is linked to a physical behavior, such as seeking resources, phototaxis, or

recharging the battery. The individual behavior of a robots in the swarm is given by the region to which its current communication state belongs. Robots are able to dynamically change their communication state by virtually navigating in the communication space. Furthermore, at each control cycle, every robot communicates its communication state to other neighbors in range. The received states can be used by other robots to coordinate their virtual navigation in order to solve SR tasks. Thereafter, robot swarms can effectively coordinate their behavior in a self-organized manner using the proposed communication module.

There have been previous approaches of SR systems that make use of virtual structures in the literature. The authors of [113] proposed the use of a virtual structure for moving multi-robot systems in formation with a high-precision. In this virtual structure, robots form a rigid geometric relationship among them and a reference frame. Moreover, the formation control is bidirectional, meaning that in order to achieve the desired formation, the positions in the virtual structure and the physical positions of the robots are influenced in both directions. In [114], the authors considered a virtual rigid body scheme for achieving multi-robot coordination. Specifically, they constructed a virtual space with its basis and boundaries defined by real-world state variables. They also transformed physical and task constraints into regions in the virtual space. Moreover, they defined the goal scenario that the robots must achieve as another coordinate in the virtual construct. Thereafter, they simplified multi-robot tasks as the problem of finding a possible path that connects the initial robot coordinates with the goal coordinates without entering into the constraint regions. In [115, 116, 117], this virtual path planning procedure with constraints as virtual obstacles was extended, and referred to as motion planning on a Representation Space (RS), for solving diverse problems. Nonetheless, these works used a representation space as a tool for planning the motion of the robots, rather than as a virtual communication medium.

A study that shares more similarities with the approach of this chapter can be found in [243], and in [244]. In these works, the authors proposed a self-organized division of labor among the members of the swarm inspired by the division of labor of honeybees based on age. They denoted this framework as *partitioning social inhibition*, which is relies on a density-based distribution of the robots along a virtual segment, that analogously represents the honeybee age. They split this segment into multiple smaller segments, each representing a different labour or task. Thereafter, the robots can move along the segment to engage in the different roles based on the target density. While this approach shares similarities with the work presented in this chapter, there are several major differences: firstly, the communication module proposed here explores the self-organization concept at the agent level instead of at the group level. This means that the system is inherently specialized to allocate different roles to each individual in the swarm, allowing personalized roles such as the positions to be occupied in a swarm formation task (see Section 5.3). Moreover, the focus of this chapter is on the capability of using the communication module in multiple different SR tasks. Additionally, the authors of [243, 244] constrained the virtual space to a segment, which is convenient for low swarm sizes but it can decrease performance as swarm size and the number of roles grow. In contrast, here, both a one-dimensional and a two-dimensional spaces are proposed, and, furthermore, the one-dimensional space was improved by virtually connecting both endpoints as an unfolded circle. This modification drastically decreases the time required to travel between two tasks and enhances the convergence of the overall system. Lastly, in Section 5.4

it is experimentally demonstrated that the system presented in this chapter provides a much faster convergence time in the self-organization process, which is a critical feature in SR systems like the ones proposed here and in [244].

5.1 The Communication Module

Inspired by the fields of cooperative control [245] and graph neural networks [246], a transferable decentralized communication module for solving SR tasks is proposed. It is based on a virtual communication state space, or simply *communication space*, where each robot has its own *communication state*. The communication space is a purely abstract construct with no direct physical counterpart—it does not represent a spatial mapping, topology, or replica of the real world in any way. The communication states are the only pieces of information communicated between the robots.

The *communication module* is categorized as the combination the following components:

(1) the communication space, and the communication states of the robots within such virtual space,

(2) the virtual dynamics of the communication states of the robots, that defines their virtual navigation,

(3) the transmission of the communication state, and its perception by the neighbors in the surroundings,

(4) the bidirectional influence and data flow between the communication module and the robot controller.

5.1.1 Communication state space

Let $\mathcal{S}$ be the communication space. In addition, let $\mathcal{R}$ be the partition of $\mathcal{S}$ introduced in Equation 5.1.

$$\mathcal{R} = \{\mathcal{R}_1, \ldots, \mathcal{R}_M\}\,, \;\; | \;\; \mathcal{S} = \mathcal{R}_1 \cup \mathcal{R}_2 \cup \cdots \cup \mathcal{R}_M \tag{5.1}$$

Moreover, each region $\mathcal{R}_j$, with $j \in \{1, \ldots, M\}$, is represented by a centroid or virtual landmark $l_j \in \mathcal{R}_j$. Using these virtual landmarks, a convenient way to define the regions is shown in Equation 5.2,

$$\mathcal{R}_j = \{\mathbf{s} \in \mathcal{S} \mid d_{\mathcal{S}}(\mathbf{s}, l_j) < d_{\mathcal{S}}(\mathbf{s}, l_k),\ \forall k \neq j\}\,, \tag{5.2}$$

where $d_{\mathcal{S}}(\mathbf{s}, \mathbf{s}^{'})$ refers to some distance metric within $\mathcal{S}$. In the special case in which $d_{\mathcal{S}}(\mathbf{s}, l_j) = d_{\mathcal{S}}(\mathbf{s}, l_k)$, the region is determined randomly between the candidate regions.

The vector $\mathbf{s}_i \in \mathcal{S}$ is defined as the communication state of the i-th robot, considering $i \in \{1, \ldots, N\}$, where N is the number of robots (swarm size). Additionally, each robot has a virtual orientation, $\theta_i(t)$, that defines the direction of virtual movement in $\mathcal{S}$. An agent i is in region $\mathcal{R}_j$ at time instant t provided that $\mathbf{s}_i(t) \in \mathcal{R}_j$.

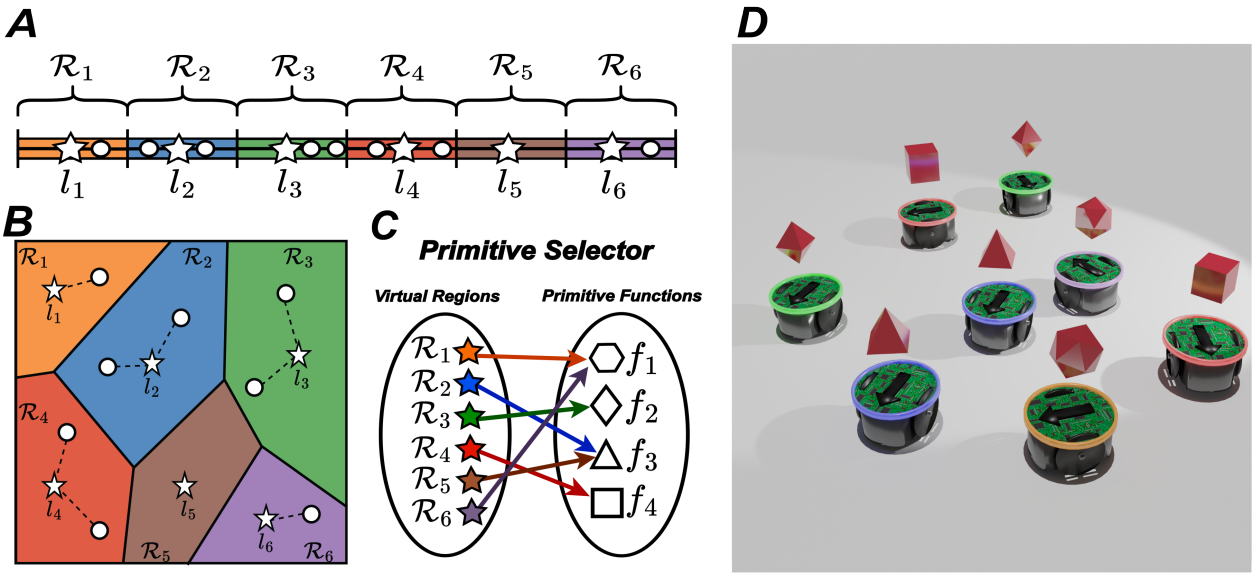


Figure 5.1: Hypothetical communication spaces and primitive selector. (**A**) Hypothetical one-dimensional communication space with six virtual regions ($\mathcal{R}_i$) and their respective landmarks l_i (denoted as stars). (**B**) Hypothetical two-dimensional communication space with six virtual regions and six virtual landmarks as in (**A**). In both (**A**) and (**B**), there are eight robots, whose communication states are represented as white circles. (**C**) Example of a primitive selector that maps virtual regions in $\mathcal{S}$ to primitive functions f_i to be executed by the robots. (**D**) Screenshot of the eight robots executing their corresponding primitive. The geometric shape displayed on top of each robot represents the primitive being executed, while the color of the robot's LED indicates its current virtual region (according to the colors of (**A** and **B**)).

Equation 5.3 defines the function $f_{\mathcal{R}}$ that determines the virtual region to which a given robot's state belongs.

$$f_{\mathcal{R}}(\mathbf{s}_i) = \mathcal{R}_k \iff \mathbf{s}_i \in \mathcal{R}_k. \tag{5.3}$$

Each virtual region corresponds to a physical behavior. Moreover, robots communicate their individual communication state to neighbors in range.

In this PhD Thesis, one-dimensional and two-dimensional communication spaces are considered. Nonetheless, all the definitions and building blocks related to the proposed communication module can be readily generalized to the n-dimensional scenario. Figure 5.1A shows a hypothetical one-dimensional communication space with six virtual regions $\{\mathcal{R}_i\}_{i=1}^{6}$ represented by their associated virtual landmarks $\{l_i\}_{i=1}^{6}$. Moreover, in this hypothetical example, there are eight robots whose communication states are represented by white circles. In this figure, there is one robot in $\mathcal{R}_1$, two robots in $\mathcal{R}_2$, two robots in $\mathcal{R}_3$, two robots in $\mathcal{R}_4$, a single robots in $\mathcal{R}_6$ and no robot in $\mathcal{R}_5$. Additionally, Figure 5.1B illustrates the analogous two-dimensional communication space with the same number of virtual regions and landmarks.

In order to make the communication module effective, some information must be shared between agents. At each control cycle, every robot i broadcasts its own communication state $\mathbf{s}_i$, which is, in turn, received by every other agent within its communication range.

5.1.2 Connection between virtual regions and robot behaviors

Similarly to the levels of competence in [90] and the behavior primitives in [92], a *primitive pool*, $\mathcal{F} = \{f_1, \ldots, f_K\}$, is defined. It is a set of primitive behaviors designed to solve specific and simple tasks. Some examples of primitives are "explore the arena", "approach a light source" (*phototaxis*), "move away from a light source", "follow another target robot", "stay in a nest ground area" or "forage from a resource area", among many other possible options. These primitives have access to the robot sensor readings and can modify the state of the robot actuators. In this PhD Thesis, only handcrafted or manually designed primitives are considered, but the set could also include evolved, learned, or hybrid primitives.

Each virtual region is associated with exactly one behavior primitive. Therefore, a mapping between the virtual regions in $\mathcal{R}$ and the primitives in $\mathcal{F}$ is defined. It implies that a robot whose communication state belongs to a virtual region executes the primitive for that region. This mapping between the sets $\mathcal{R}$ and $\mathcal{F}$ is refered as the *primitive selector*.

An example of a primitive selector is illustrated in Figure 5.1C, where six virtual regions of either Figure 5.1A or B are mapped onto four primitives $\{f_1, f_2, f_3, f_4\}$ (represented graphically as different geometric shapes). Figure 5.1D shows a small swarm of eight robots that are solving a certain task. The primitive that each robot of the figure is executing is graphically represented with a 3D geometry according to the robot's communication state in Figure 5.1A and B and the primitive selector in Figure 5.1C.

5.1.3 Virtual navigation in the communication space

The communication state of each robot is subject to some dynamics that modify its coordinates in $\mathcal{S}$. The communication state dynamics in $\mathcal{S}$ are denoted as *virtual navigation*. Furthermore, a *navigation policy* decides the virtual navigation rules and dynamics, driven by some goal or convergence criteria. In order to be effective, such navigation policy should take into consideration the virtual landmarks and the communication states of the neighbors. In this chapter, the navigation policy goal is established in Equation 5.4,

$$\forall \mathbf{s}_i, \mathbf{s}_j \in \{\mathbf{s}_1, \ldots, \mathbf{s}_N\},\ \mathbf{s}_i \neq \mathbf{s}_j \Rightarrow f_{\mathcal{R}}(\mathbf{s}_i) \neq f_{\mathcal{R}}(\mathbf{s}_j). \tag{5.4}$$

This target goal condition is met when the communication state of each robot belongs to a different virtual region of $\mathcal{R}$, that is not occupied by any other member of the swarm. Note, moreover, that the convergence of each agent to a distinct virtual region also implies that, upon convergence, each robot is controlled by a different primitive behavior (division of labor). To complement Equation 5.4, notice that the virtual regions form a partition of $\mathcal{S}$, so that any communication state always belongs to one and only one virtual region, regardless the navigation policy.

To fulfill the imposed policy, the virtual navigation is controlled by the *communication controller*, f_{comm}. The communication controller is a completely different unit that must not be confused with the traditional robot controller. The latter maps sensor readings to actuator actions and defines the robot behavior in the physical world, while the former defines the virtual navigation of the robots in the communication space. The communication controller

of the i-th robot is generically defined in Equation 5.5,

$$a_i,\ \theta_{tar,i} = f_{comm}(\mathbf{s}_i, \mathbf{s}_{cst}, l_{cst}, l_{tar}), \tag{5.5}$$

where the outputs are the normalized navigation speed $a_i \in [0, 1]$ and the target virtual orientation $\theta_{tar,i}$ of robot i in the communication space $\mathcal{S}$. Alternatively, the communication controller receives as inputs four vectors, namely, the closest communication state among the neighbors in $\mathcal{S}$, $\mathbf{s}_{cst}$, the closest landmark, l_{cst} (which is in turn the landmark associated to the virtual region $f_{\mathcal{R}}(s_i)$), a target landmark l_{tar}, and its own communication state $\mathbf{s}_i$. Before explaining the computation of the target landmark l_{tar}, let $\mathbf{p}_i \in \mathbb{N}_0^M$ ($\mathbb{N}_0 = \mathbb{N} \cup 0$) be a vector of priorities, where is 0 is the highest priority and the priority decreases as the number grows. The priorities in $\mathbf{p}_i$ are associated to each of the virtual landmarks as a one-to-one mapping, so that $\mathbf{p}_i(m)$ is the priority of landmark l_m for all $m \in \{1, \dots M\}$ according to robot i. The sub-index i of $\mathbf{p}_i$ is used to highlight that the priorities are subjective to each robot $i \in \{1, \dots, N\}$, so that each robot can have different landmark priorities.

The target landmark l_{tar} is selected according to three criteria:

(1) whether the virtual region $\mathcal{R}_{tar}$ of l_{tar} is already occupied by other neighbors,

(2) the priority of the landmark, and

(3) the distance from the robot's communication state to the landmark.

Each of the three criteria are also listed according to their importance in the selection, meaning that finding a virtual region that is unoccupied is more important than finding the highest priority and, in turn, selecting the highest priority is more relevant than distance.

The communication controller is responsible for generating a virtual navigation velocity and direction. The dynamics of the virtual orientation of a robot i are characterized by Equation 5.6,

$$\tau_\theta \frac{\partial \theta_i(t)}{\partial t} = \theta_{tar,i}(t) - \theta_i(t), \tag{5.6}$$

while the dynamics of the communication state are defined in Equation 5.7,

$$\tau_s \frac{\partial \mathbf{s}_i(t)}{\partial t} = a_i(t) \cdot \begin{pmatrix} \cos(\theta_i(t)) \\ \sin(\theta_i(t)) \end{pmatrix}. \tag{5.7}$$

The fixed time constants τ_θ and τ_s also influence the linear and angular speeds of the virtual navigation, and they are settled to $20 \cdot \partial t$ in both cases. Preliminary studies revealed that they are a good trade-off between smoothness and fast response in the virtual navigation.

Equations 5.6 and 5.7 represent a dynamical system specifically designed for two-dimensional communication spaces. Thus, in the case of one-dimensional communication spaces, the dynamics simplify to Equation 5.8, as there is no longer a need to include dynamics for the orientation.

$$\tau_s \frac{\partial \mathbf{s}_i(t)}{\partial t} = a_i(t) \cdot \theta_{tar,i}(t) \tag{5.8}$$

In Equation 5.8, the variable $\theta_{tar,i} \in \{-1, 1\}$ has two possible values, so that each one represents one of the two navigation directions in the single dimension of the communication space.

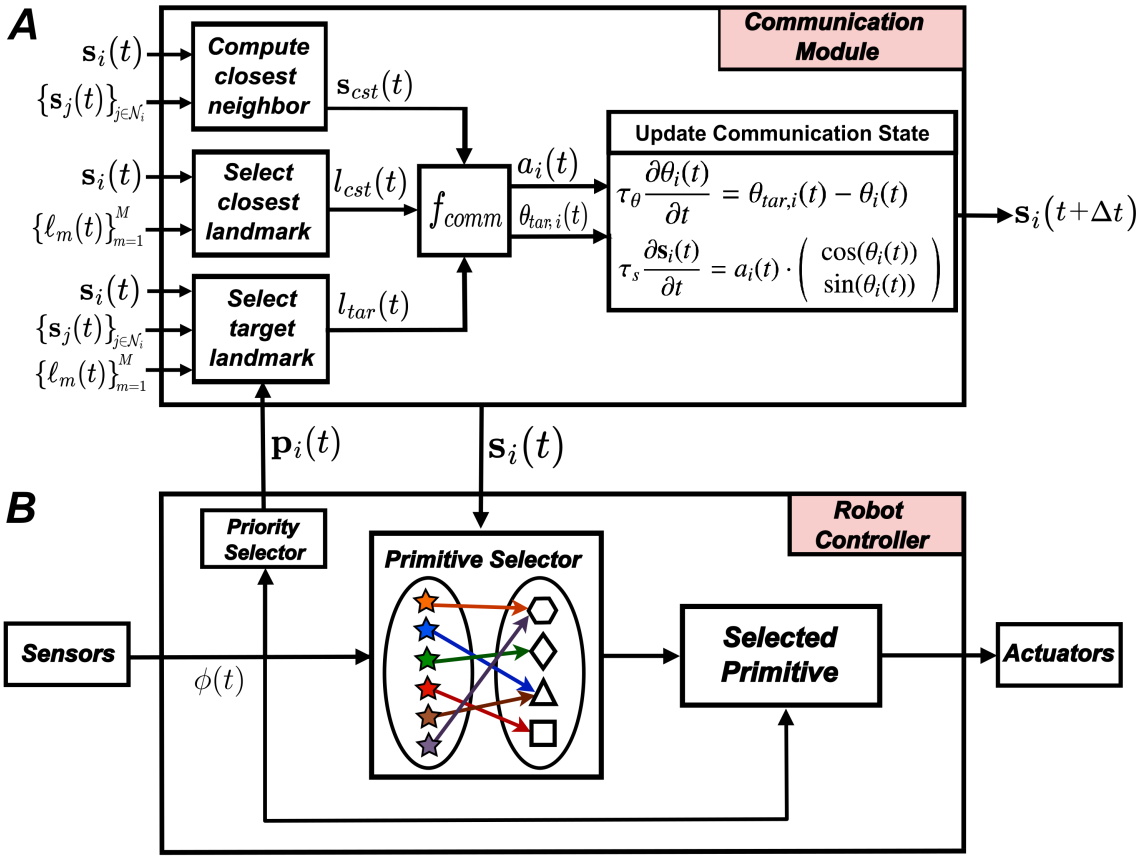


Figure 5.2: Diagram of the overall system. (**A**) Communication module. (**B**) Robot controller.

Figure 5.2A summarizes the stages of the virtual navigation process of a robot in the communication space. Firstly, $\mathbf{s}_{cst}$, l_{cst} and l_{tar} are computed using all the landmarks ($\{l_m\}_{m=1}^{M}$), and all the communication states of other robots in communication range ($\{\mathbf{s}_j\}_{j\in\mathcal{N}_i}$). Subsequently, these vectors are fed to the communication controller f_{comm}, that generates the desired navigation speed and virtual orientation. Finally, the communication state of the robot is updated based on Equations 5.6 and 5.7.

5.1.4 Robot controller

The robot controller, illustrated in Figure 5.2B, is responsible for the physical behavior of a robot. In each control cycle, the primitive selector is used to pick one of the primitives available in a *primitive pool*, based on the current region of the robot in the communication space. Moreover, the selected primitive is executed using the current sensory readings, denoted as $\phi(t)$. In addition, the robot controller is also composed by the *priority selector*. A virtual region's priority determines its relative importance for a given robot and is considered when the robot performs virtual navigation. The priority selector is defined as in Equation 5.9,

$$\mathbf{p}_i(t) = \text{PrioritySelector}\left(\phi(t)\right), \tag{5.9}$$

so that its outcome is the vector $\mathbf{p}_i$, resulting from the concatenation of the priorities of all the primitives of the i-th robot. The priorities can be set to fixed values, for instance, as in the aggregation and formation tasks (see Section 5.3.1 and 5.3.2). However, priorities can also be dynamically updated using conditional rules and the current sensor readings, $\phi(t)$. As an example, a change in the priority of a primitive can happen when a robot runs low on battery and needs to critically prioritize a "*recharge battery*" behavior. In this scenario, the priority selector would simply check whether some battery level threshold was crossed, and modify the "*recharge battery*" priority accordingly.

5.1.5 Interactions between the communication module and the robot controller

Figure 5.2 shows an overview of the communication module, the robot controller, and the data flow between them. The bidirectional interactions between these two components are the following:

- **Influence of the communication module on the robot controller**: the communication module modifies the behavior of the robots in the physical environment through the primitive selector and the communication state. Specifically, the communication state of a robot and the virtual region to which it belongs determine the physical behavior primitive executed by a robot. Similarly, transitioning from one virtual region to another also causes a change in the robot's physical behavior.
- **Influence of the robot controller on the communication module**: the robot controller can alter the virtual navigation that characterizes the communication module by means of the priority vector $\mathbf{p}_i(t)$ local to each robot i. Consequently, $\mathbf{p}_i(t)$ of a robot i determines the virtual regions that this robot should occupy with highest priority at time instant t. The robot controller can modify the values of $\mathbf{p}_i$ at each control cycle based on the sensor readings.

5.1.6 Evolution and deployment phases

The parameters of the ANN that define the communication controller (f_{comm}) are optimized with the ultimate aim of fulfilling Equation 5.4. The evolution of the communication controller is completely decoupled from the physical robot controller. Therefore, two complementary stages can be considered: (i) an *evolution phase* and (ii) a *deployment phase*.

- The *evolution phase* is devoted to the optimization of the communication controller, in isolation, where robots learn to navigate in the communication space. In this study, the communication controller is agnostic to the SR task taking place in the physical environment. Therefore, no primitive behaviors are assigned to the virtual regions at this stage.
- The *deployment phase* occurs after the communication controller is evolved. In this phase, the communication module is configured for a specific SR task, namely, assigning primitive behaviors to virtual regions and defining the primitive selector and the priority selector. Even though manually designed primitives and selectors are considered in this

Chapter, these components could also be evolved or learned according to the specific task requirements.

One of the main strengths of the proposed communication module is that it is task-agnostic and can be used in diverse tasks without the need to re-design, re-evolve, or re-train it. This property of the module is refered as *strong transferability* in this Thesis. To use the communication module in a new task, it is only necessary to configure the module as described above in the deployment phase.

5.1.7 Illustrative example

A simple foraging task can be conveniently used as an example to explain the way in which the communication module is used to solve SR tasks. Such foraging problem, which also serves as an introduction to one of the three experiments defined and solved in subsequent sections, is composed by a nest zone and two food areas. The aim of the robots is to find the food areas and, thereafter, transport as many food elements as possible to the nest by performing round trips between the corresponding resource area and the nest. Additionally, the goal in a multi-agent scenario is to perform a temporal task allocation so that each robot seeks and forages from a different food area.

This example with two robots is shown in Figure 5.3. Figure 5.3A depicts the communication space and the virtual navigation of each of the two robots (as white and black curves, respectively). There are five virtual regions ($\mathcal{R}_1, \ldots, \mathcal{R}_5$), each linked to a distinct primitive behavior ($f_1, \ldots, f_5$) according to the primitive selector shown in Figure 5.3B. The callouts show the five-dimensional priority vectors with their values set to either `OFF` or `ON` for each of the five regions. The style of the curve (dotted, dashed, or solid) depicts the part of the trajectories in which the priorities of the callouts are used by each robot. The value `OFF` means that the robot's virtual navigation will not be attracted towards it. For example, the vector (`OFF`, `OFF`, `OFF`, `OFF`, `ON`) indicates that only region $\mathcal{R}_5$ has priority, while the rest can be neglected. Consequently, the robot should navigate towards $\mathcal{R}_5$ with the aim of executing behavior f_5 ("Return to nest").

Focusing on the trajectory of robot 1 (in white), the virtual navigation is the sequence $\{\mathcal{R}_1, \mathcal{R}_3, \mathcal{R}_5, \mathcal{R}_3, \mathcal{R}_5, \ldots\}$, so that the primitives executed by this robot are $\{f_1, f_3, f_5, f_3, f_5 \ldots\}$. In terms of robot behaviors, this sequence is as follows: (i) the robot searches for a food area by exploring the environment, and once food area 1 is found, (ii) it continually transports resources from that food area to the nest. The virtual navigation of robot 2 (in black) is the series $\{\mathcal{R}_2, \mathcal{R}_4, \mathcal{R}_5, \mathcal{R}_4, \mathcal{R}_5, \ldots\}$ and the sequence of robot 2 primitives are $\{f_2, f_4, f_5, f_4, f_5, \ldots\}$. In this case, robot 2 starts executing the primitive "Find food area 2", because robot 1 is already looking for the other food source. Subsequently, robot 2 performs a round trip foraging from the food area 2 and the nest. The values of the priority vectors play a crucial role in this example. For instance, the round trips between $\mathcal{R}_3$ and $\mathcal{R}_5$ are produced because when a robot acquires food, the robot controller updates the priority vector to (`OFF`, `OFF`, `OFF`, `OFF`, `ON`), indicating that only region $\mathcal{R}_5$ is active, and the new target behavior of the robot is returning to the nest (primitive f_5).

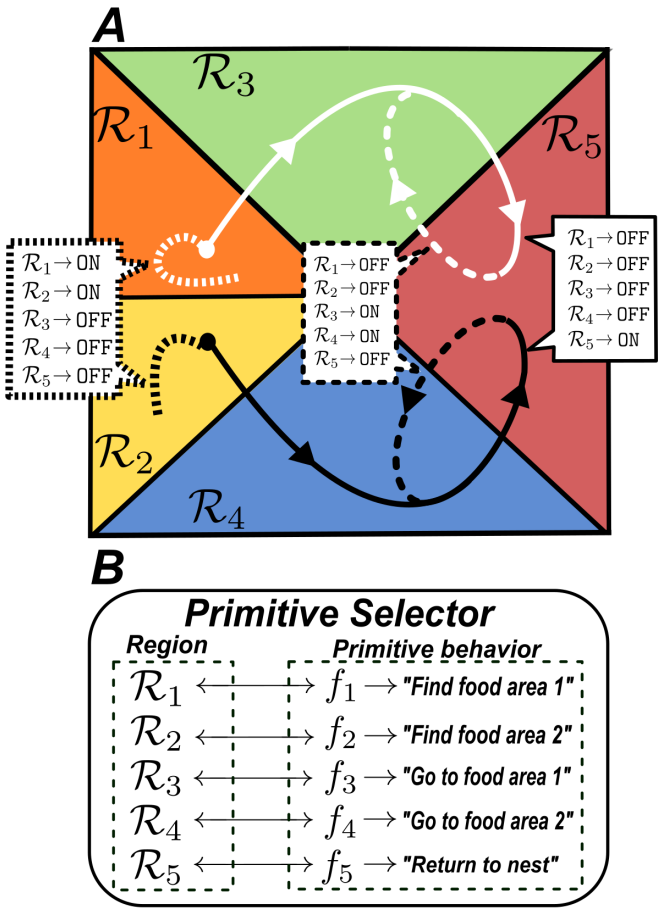


Figure 5.3: Example showing how the communication module and robot controller can solve a simple foraging. (**A**) Communication space showing example trajectories of two cooperating robots solving a simple foraging. (**B**) Primitive selector showing five primitive behaviors (f_i) and how they are associated to the regions of the communication space. It also provides a short description of each goal of each primitive. (**A**) also shows multiple callouts with the values of the priorities of the robots during specific parts of the trajectories (illustrated with the style of the curves, being either solid, dashed, or dotted). For the sake of clarity, in this example, either `ON` and `OFF` priority values are considered, so that `ON` means that the corresponding region has priority and the robot is attracted to it, and `OFF` implies that the corresponding region can be neglected. For example, the vector (`OFF`, `OFF`, `OFF`, `OFF`, `ON`) indicates that only region $\mathcal{R}_5$ has priority, and, thus, the robot's target behavior would be f_5 ("Return to nest").

5.2 Evolution of the communication controller

5.2.1 Description of the evolutionary process

In order to obtain a virtual navigation of robots in the communication space that satisfies the desired goals defined in Equation 5.4, an ANN is used as the communication controller f_{comm}. Moreover, the parameters and topology of f_{comm} are evolved using the NeuroEvolution of Augmenting Topologies (NEAT) algorithm [83]. The reader is redirected to Appendix A for a

detailed theoretical description of NEAT.

During each generation of NEAT, each individual is evaluated using a robotics simulation software (see Section 5.3). Each simulation is accomplished using a swarm of ten robots and a communication space composed of ten virtual landmarks. The evolution process is task-agnostic and is decoupled from any physical behavior. Therefore, the evaluations are done using static robots that do not tackle any specific SR task. Instead of the physical robot behavior, the focus of the simulations during this evolution stage lies on the dynamics of the communication states and on the convergence of the virtual navigation of the robots. At each control cycle, each robot transmits its communication state, $\mathbf{s}_i$, to its neighborhood. Thereafter, one iteration of the communication module (with its control flow and main stages depicted in Figure 5.2A) is computed, resulting in the next communication state of each robot. In this chapter, the last aim of the artificial evolution is that the communication state of each robot converges to a different virtual region that is not occupied by any other agent (thus satisfying Equation 5.4). The final result of each evaluation is a fitness score that evaluates the performance of the NEAT's individuals, using the fitness function defined in Section 5.2.4. Once evolved, the optimized communication module can be transferred to multiple tasks in the deployment stage without the need to re-evolve the communication controller.

The physical positions and virtual communication states of the robots are initialized randomly, ensuring the compactness of the robot swarm and avoiding isolated robots. The initialization of the communication states of the robots is uniform along the entire communication space. The coordinates of the virtual landmarks in the communication space is also randomly determined, guaranteeing a minimum distance between any two landmarks. This minimum distance is a hyper-parameter that depends on the total number of landmarks. For ten landmarks, it is fixed to 0.3.

5.2.2 Topology of the communication space

The communication framework is assessed in experiments with both 1D and 2D communication spaces. The one-dimensional communication space is defined as the line segment $\mathcal{S}_1 = [-L/2,\ L/2)$ with an additional property that allows a direct connection between the endpoints $-L/2$ and $L/2$ (virtually the same point in the communication space). This one-dimensional communication space can be also seen as a circle embed in a higher dimensional space.

Figure 5.4A shows the one-dimensional communication space as a circle in a 2D space, while Figure 5.4B illustrates the unfolding of such circle that leads to the mentioned line segment. Notice that both representations depict the exact same communication space. The distance metric that characterizes $\mathcal{S}_1$ is defined in Equation 5.10,

$$d_{\mathcal{S}_1}(\mathbf{s}, \mathbf{s}^{'}) = \min\left\{\left|\mathbf{s} - \mathbf{s}^{'}\right|, L - \left|\mathbf{s} - \mathbf{s}^{'}\right|\right\}, \tag{5.10}$$

where $\mathbf{s}, \mathbf{s}^{'} \in \mathcal{S}_1$.

Besides, the two-dimensional communication space is defined as an unfolded torus in $\mathcal{S}_2 = [-W/2,\ W/2) \times [-H/2,\ H/2)$. The upper and lower boundaries of $\mathcal{S}_2$ are connected and the left and right sides are also connected. Figure 5.4C shows $\mathcal{S}_2$ as a torus in a 3D space, while Figure 5.4D illustrates the unfolded torus. As in the one-dimensional case, the communication

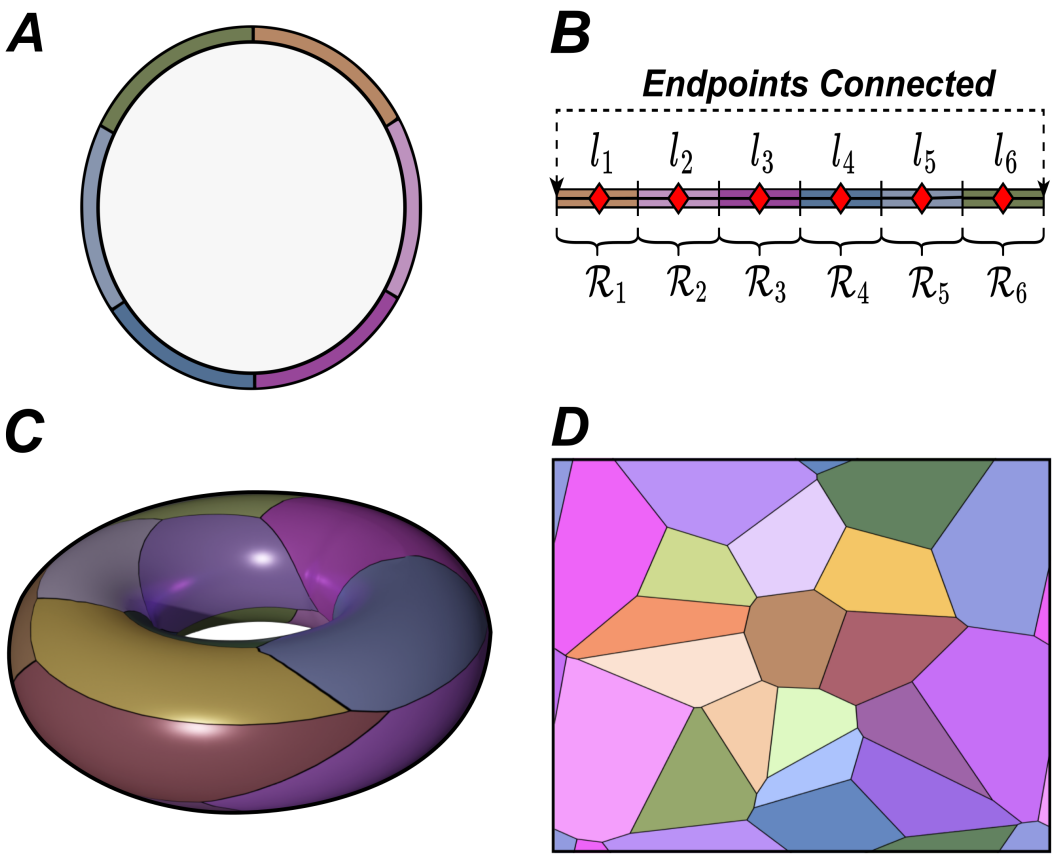


Figure 5.4: Communication space topologies used in the experiments. (**A**) Circle communication space embedded in a three-dimensional space. Remark: the thickness of the circle is only represented for visualization purposes, albeit the structure is a compact one-dimensional manifold. (**B**) Unfolded circle communication space. (**C**) Torus communication space embedded in a three-dimensional space. (**D**) Unfolded torus communication space.

space represented in both figures is equivalent. The distance metric of $\mathcal{S}_2$ is formulated in Equation 5.11,

$$d_{\mathcal{S}_2}(\mathbf{s}, \mathbf{s}') = \left\| \begin{pmatrix} \min\left\{ \left| \mathbf{s}_1 - \mathbf{s}_1' \right|, W - \left| \mathbf{s}_1 - \mathbf{s}_1' \right| \right\} \\ \min\left\{ \left| \mathbf{s}_2 - \mathbf{s}_2' \right|, H - \left| \mathbf{s}_2 - \mathbf{s}_2' \right| \right\} \end{pmatrix} \right\|_2 \tag{5.11}$$

where $\mathbf{s} = (\mathbf{s}_1, \mathbf{s}_2)^\top$ and $\mathbf{s}' = (\mathbf{s}_1', \mathbf{s}_2')^\top$ and $\mathbf{s}, \mathbf{s}' \in \mathcal{S}_2$.

5.2.3 Evolved communication controller

The ANN model used in f_{comm} is a CTRNN [247], which is an ANN with feedback connections that operates in continuous time. The use of a recurrent neural network is motivated by the dynamic nature of the virtual navigation in the communication space, which poses a scenario where a feed-forward model without any sort of memory may not be sufficient. A detailed description of the CTRNN model and the firing rate neurons that compose it is provided in

Appendix B. The precise choice of the CTRNN model instead of a simpler recurrent neural network that operates in discrete time is justified by the continuous dynamics of the mentioned virtual navigation.

The membrane time constants of the neurons are fixed to the value of $\tau_m = 1$ and the Euler time step used in the simulations is $\Delta t = 0.1$. A ratio of $\frac{\tau_m}{\Delta t} = 10$ is a good trade-off value that allows the neurons to react quickly to input changes. Smaller values of this ratio would be too close to the time discretization Δt used by Euler integration. On the contrary, large values of τ_m would result in an undesired slow virtual movement in the communication space. The activation function of the neurons is the sigmoid function for all the neurons in the CTRNN.

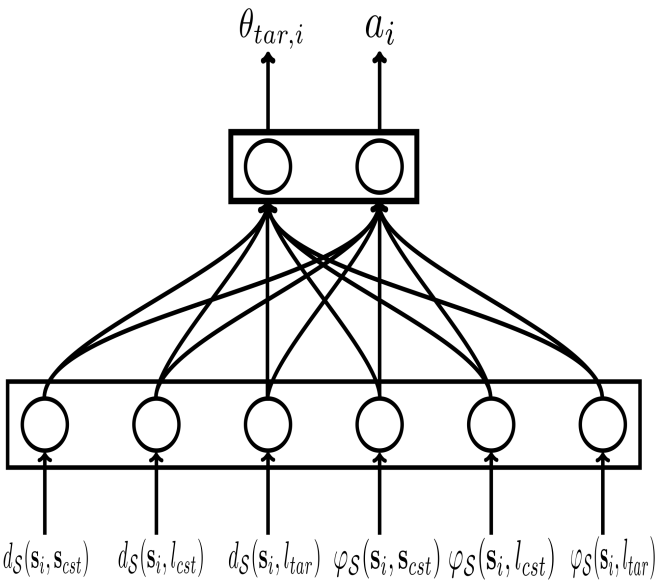


Figure 5.5: Architecture of the CTRNN model used to control the virtual navigation of the robots in the communication space.

Figure 5.5 shows the diagram of the CTRNN architecture used in the experiments. The bottom circles of the figure denote the input nodes of the neural network and the top nodes represent the output neurons. Recall from Section 5.1.3 that f_{comm} receives as input the closest neighbor's communication state, $\mathbf{s}_{cst}$, the closest landmark, l_{cst}, and the target landmark l_{tar}. The CTRNN execution is preceded by a pre-processing stage in which these vectors are converted to the distance and angle between the robot communication state and the corresponding vector. For instance, $d_{\mathcal{S}}(\mathbf{s}_i, \mathbf{s}_{cst})$ and $\varphi_{\mathcal{S}}(\mathbf{s}_i, \mathbf{s}_{cst})$ are the distance and angle between $\mathbf{s}_i$ and $\mathbf{s}_{cst}$ in $\mathcal{S}$. On the contrary, the outputs of the CTRNN are the virtual navigation velocity and target orientation for robot i, as defined in Section 5.1.3.

5.2.4 Evolutionary Setup

The fitness function *FF*, defined in this subsection, is used to guide the artificial evolution towards a solution that fulfills the objective in Equation 5.4. The individual fitness score of a robot increases while the distance to a virtual landmark is decreased, if and only if that robot

is the only individual in the surroundings of the landmark. This fitness function imposes that there should only be one robot inside each virtual region to reach the goal of Equation 5.4. The instantaneous fitness function of a single robot i, $FF_i(t)$, used by the NEAT algorithm to optimize the CTRNN model is shown in Equation 5.12,

$$FF_i(t) = \begin{cases} \exp\left\{-\alpha \cdot d_{\mathcal{S}}(\mathbf{s}_i(t),\, l_i^*(t))\right\}, & \text{if } \mathbf{s}_j(t) \notin \mathcal{R}_i^*(t), \forall j \neq i \\ 0, & \text{otherwise,} \end{cases} \tag{5.12}$$

where $\mathcal{R}_i^*(t)$ is the virtual region to which the communication state of agent i currently belongs (i.e. $\mathbf{s}_i \in \mathcal{R}_i^*$ at time step t). Additionally, l_i^* refers to the virtual landmark corresponding to region $\mathcal{R}_i^*$. The value of FF_i will exponentially increase as the distance to l_i^* is reduced, provided that agent i is the only robot whose communication state is inside virtual region $\mathcal{R}_i^*$ (i.e. $\mathbf{s}_j(t) \notin \mathcal{R}_i^*(t), \forall j \neq i$). The value of α modulates the decay rate of the fitness as the distance to l^* rises. In the experiments, it is established $\alpha = 8$ to avoid sparse fitness landscapes while not rewarding too much the robots for being on the borders of the virtual regions.

The final fitness score of a NEAT individual is computed as the average of $FF_i(t)$ across robots $i \in \{1, \ldots, N\}$ and simulation control cycles $k \in \{1, \ldots, T\}$ considering a sampling period Δt (see Equation 5.13).

$$FF = \frac{1}{NT} \sum_{k=0}^{T} \sum_{i=1}^{N} FF_i(k\Delta t) \tag{5.13}$$

Additionally the fitness function is evaluated 10 different times for each individual, with random and independent initialization.

The NEAT population is composed of 100 individuals, among which the best two are preserved each generation as elites. The number of elites is fixed to two individuals as a good trade-off between exploitation of the best-performing candidates and the exploration of new solutions. Increasing the number of elites further would incrementally result in a high selective pressure of the evolutionary algorithm (and thus increasing the possibilities of finding a local optima that does not satisfy the problem requirements). Each parameter of the CTRNN model (i.e. synapse weights and neuron biases) is mutated with a probability of $p_\theta = 0.1$, using a Gaussian mutation with a mutation power of 0.05. Moreover, there is a probability of 0.05 to add a new node, and a probability of 0.2 to generate a new connection to the CTRNN topology. The rest of the hyper-parameters are based on the proposed configuration by the authors of NEAT [83] and on previous studies of NEAT applied to SR [248, 249]. Specifically, the values $c_1 = 1$, $c_2 = 1$, and $c_3 = 0.4$ are used as the compatibility coefficients, and $\delta_t = 3$ for the compatibility threshold. These parameters are used during the speciation process to select the most compatible species for each genotype. The weights and biases are constrained to the range $[-10, 10]$ and are initialized using a Gaussian distribution with zero mean and a $\sigma = 1$.

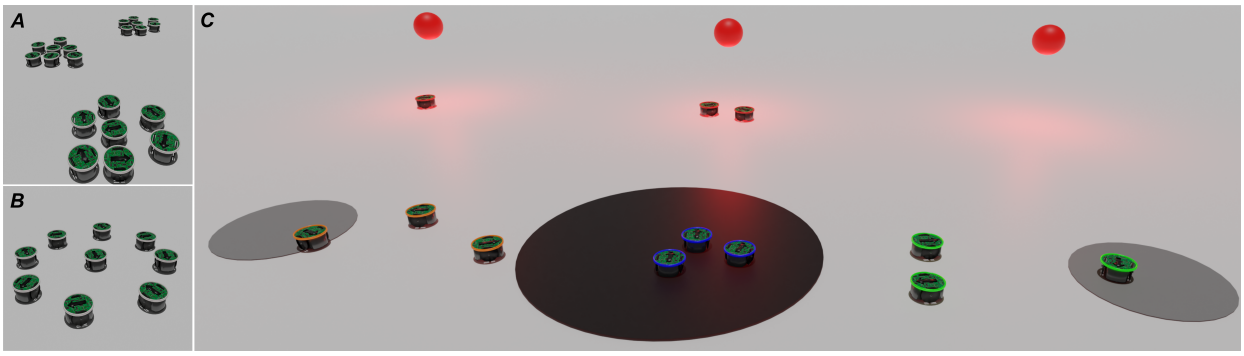


Figure 5.6: Illustration of the proposed SR experiments. (**A**) Aggregation in groups of desired sizes. (**B**) Formation of target swarm geometries. (**C**) Basic foraging with a nest area (black), two food areas (grey) and three red light sources where the robots can recharge their batteries. The aim is to perform a temporal role allocation so that there are always $N/4$ robots foraging from each of the resources ($N/2$ foragers in total), while the remainder of the swarm members are either charging their batteries or waiting inside the nest.

5.3 Experiments

To assess the communication framework, three widely known SR tasks are addressed: (i) an aggregation of the robots in groups of fixed sizes, (ii) a formation of target swarm geometries, and (iii) a basic foraging with robot battery dependence. The exact same communication module, previously evolved according to the details and hyper-parameters discussed in Section 5.2, is used. For all the experiments, different primitives are defined to suitably fulfill the requirements imposed by the corresponding task. Moreover, simulations are accomplished using the `pybullet` library [250] for real-time collision detection and multi-physics simulations. The robots are equipped with 8 IR-based proximity sensors distributed along their perimeter, a GPS in order to know their absolute position in the arena and communication transmitters and receivers that share their communication states along their neighborhood. In the case of the foraging experiment, the robots are also equipped with 8 light sensors positioned along its perimeter and a ground sensor that detects the color of the floor below the robot.

5.3.1 Aggregation in groups

The aggregation in groups is a highly important task that goes beyond simple aggregation, as it can be used to conveniently divide the labor of the swarm across teams in real-world geographically distributed tasks [251, 252], such as firefighting or search and rescue. The reader is redirected to Chapter 2.5.1 for a review of SR studies in the literature that have addressed the aggregation task using different algorithms and robot controllers.

A clustered aggregation task, in which the robots have to form a desired number of groups with a target number of members in each group, is considered. Moreover, the robots of each group have to aggregate as closely as possible with the other members of their group, while avoiding collisions. G is defined as the number of groups and $G_i, i \in \{1, \ldots, G\}$, as the number of target members of each group. Thereafter, there are G primitive behaviors with

the following meanings:

$$f_i \longrightarrow \begin{array}{l}\text{“Go to the center of mass coordinates}\\ \text{of the neighbors also belonging to group } i\text{”.}\end{array}$$

Furthermore, the number of virtual regions of the communication space equals the swarm size and, for each $i \in \{1, \ldots, G\}$, the primitive selector maps G_i different virtual regions to the primitive function f_i. Figure 5.6A illustrates an example of the aggregation in groups with 21 robots, $G = 3$ groups, three primitives $\{f_1, f_2, f_3\}$, and $G_1 = G_2 = G_3 = 7$. Consequently, in this example, $\{\mathcal{R}_1, \ldots, \mathcal{R}_7\}$ are all mapped onto f_1, $\{\mathcal{R}_8, \ldots, \mathcal{R}_{14}\}$ are mapped onto f_2 and $\{\mathcal{R}_{15}, \ldots, \mathcal{R}_{21}\}$ are associated to f_3.

All the target groups of this task have the same importance and, therefore, all the virtual region priorities are fixed to the same value of 1. Therefore, the priority selector used in this experiment is simply:

$$\mathbf{p}_i(t) = (1, \ldots, 1),\ \forall t,\ \forall i \in \{1, \ldots, N\} \tag{5.14}$$

At the beginning of the task execution, the target number of groups and robots per group are established by the researcher. Moreover, robots' positions are initialized randomly within a square area of either 2×2 meters for swarms with less than 20 robots, and 3×3 meters for swarm sizes greater than 20. The orientations of the robots are also randomly initialized between 0 and 2π.

5.3.2 Formation control

The second experiment is a formation of desired swarm geometries (see e.g. [253, 254]). Formation control is one of the fundamental behaviors in SR, and it is the pillar of many real-world applications, such as collective motion, coordinated patrolling, area coverage, or surveillance, among others. The reader is redirected to Chapter 2.5.2 for a review of SR studies in the literature that have addressed the formation control using different algorithms and robot controllers.

The proposed formation is relative to the center of mass of the robots, implying that it can be established in any area of the arena. The only requirement is that the distances between robots are approximately preserved according to the geometry. As an example, Figure 5.6B illustrates a target formation with 9 robots.

Let the target formation of N points be formulated as in Equation 5.15,

$$\{\mathbf{x}_1^*(t), \ldots, \mathbf{x}_N^*(t)\} = \{\mathbf{c}(t) + \Delta_1, \ldots, \mathbf{c}(t) + \Delta_N\}, \tag{5.15}$$

where $\mathbf{c} \in \mathbb{R}^2$ is the center of mass of the swarm and, $\forall i \in \{1, \ldots, N\}$, $\Delta_i \in \mathbb{R}^2_+$ is the offset vector that defines the coordinates of the point $\mathbf{x}_i^*$ relative to $\mathbf{c}$ in the formation.

In this experiment, the primitive selector is an injective mapping between $\mathcal{R} = \{\mathcal{R}_1, \ldots, \mathcal{R}_N\}$ and $\mathcal{F} = \{f_1, \ldots, f_N\}$, so that each primitive behavior has the following meaning,

$$f_i \longrightarrow \text{“Go to coordinates } \mathbf{x}_i^*(t) = \mathbf{c}(t) + \Delta_i\text{”.}$$

All the spots of the target formation have the same importance and, therefore, all the virtual region priorities are fixed to the same value of 1 (see Equation 5.14). At the beginning of the task execution, the target formation geometry is manually set. Moreover, robots' physical positions and orientations are randomly initialized as in the aggregation task (Section 5.3.1).

5.3.3 Foraging

The third experiment is a basic multi-place foraging with a nest area and two resource areas located at opposites sides of the nest (see Figure 5.6C). Foraging [28] is a highly important SR task that can be applied to multiple real-world problems, such as garbage recolection and warehouse logistics and automation, among many others. The reader is redirected to Chapter 2.5.2 for a review of SR studies in the literature that have addressed the foraging task using different algorithms and robot controllers.

The robots are equipped with a battery that is discharged when the robot is outside the nest and can be recharged in the proximity of a light source. The aim is to maintain a 25% of the robots foraging form *food 1* and other 25% that forage from *food 2*. The remaining 50% of the robots either wait inside the nest or recharge their batteries. Thus, there are four roles engaged by the robots, namely, *stay in nest*, *forage food 1*, *forage food 2* and *recharge battery*. Notice that it is a temporal or dynamic role allocation because when forager robot runs out of battery, another robot staying at the nest has to update its role to cover the absence of the first agent. Figure 5.6C depicts the foraging experiment with $N = 12$ robots accomplishing the task with the optimal distribution of roles. The figure also illustrates the locations of the nest (black ground area), the food areas (grey ground areas) and the red light sources where the robots can recharge their battery level.

The battery level of each robot i is denoted as $b_i(t) \in [0, 1]$ and is subject to the dynamics in Equations 5.16 and 5.17 for its charge and discharge, respectively,

$$\frac{\partial b(t)}{\partial t} = \gamma\,(1 - b(t)^2), \tag{5.16}$$

$$\frac{\partial b(t)}{\partial t} = \Delta b, \tag{5.17}$$

where γ and Δb are the charge and discharge coefficients and $b(0) = 1$ is the initial condition.

In this experiment, there are $N + 1$ virtual regions and four primitive behaviors $\{f_1, f_2, f_3, f_4\}$ with the following meanings:

$f_1 \longrightarrow$ "Stay in the nest area",
$f_2 \longrightarrow$ "Forage from food area 1",
$f_3 \longrightarrow$ "Forage from food area 2",
$f_4 \longrightarrow$ "Recharge battery (*phototaxis*)".

Additionally, the primitive selector performs the following mapping between virtual regions

and primitives:

$$\begin{array}{lcll} \mathcal{R}_{nest} & = & \{\mathcal{R}_1, \ldots, \mathcal{R}_{\frac{N}{2}}\} & \longrightarrow f_1 \\ \mathcal{R}_{food1} & = & \{\mathcal{R}_{\frac{N}{2}+1}, \ldots, \mathcal{R}_{\frac{3N}{4}}\} & \longrightarrow f_2 \\ \mathcal{R}_{food2} & = & \{\mathcal{R}_{\frac{3N}{4}+1}, \ldots, \mathcal{R}_N\} & \longrightarrow f_3 \\ \mathcal{R}_{bat} & = & \mathcal{R}_{N+1} & \longrightarrow f_4 \end{array}$$

By default, the priorities of each virtual region are 2 for the regions in $\mathcal{R}_{nest}$, 1 in the case of $\mathcal{R}_{food1}$ and $\mathcal{R}_{food1}$, and 3 for $\mathcal{R}_{bat}$. Consequently, foraging from one of the food areas has highest priority, staying in the nest has medium priority, and recharging the battery is a low priority primitive. Nonetheless, when the condition $b_i(t) < 0.4$ is met, the priority of $\mathcal{R}_{bat}$ is updated to 0, turning battery recovery into the primitive with highest priority. When the battery level is again above 0.9, the priorities for the corresponding robot return to their default values. Specifically, the following priority selector is used in the foraging task:

$$\mathbf{p}_i(t) = \begin{cases} (\underbrace{1, \ldots, 1}_{N/2}, \underbrace{2, \ldots, 2}_{N/2}, 3), & \text{if } \textit{battery recharged} \\ \\ (\underbrace{1, \ldots, 1}_{N/2}, \underbrace{2, \ldots, 2}_{N/2}, 0), & \text{if } \textit{battery depleted} \end{cases}$$

The environment is composed of a nest area (circular black ground area), two food areas (circular gray ground areas) of equal radius, and three red light sources that can be sensed from every point in the environment, see Figure 5.6. The nest area is always located at the center of the arena and has a radius twice as large as the food areas. Specifically, the nest radius is fixed to 1 meter for swarm sizes smaller than 40 and 2 meters for swarms with more than 40 robots. The food areas are located symmetrically at either sides of the nest, being the *nest-to-food* distance: 3 meters for swarms up to 40 robots, and 5 meters for swarm sizes greater than 40. The robot positions are initialized randomly within a square centered at the nest's origin, of either 2×2 and 3×3 for swarm sizes in the ranges $[1, 40)$ and $[40, 60]$, respectively. The battery is always fully charged at the beginning of every simulation, and γ and Δb are always fixed to 0.01 and 0.001.

5.4 Results

5.4.1 Communication module in isolation

First, the results of the evolved communication module are presented in isolation (*evolution stage* described in Section 5.1.6). Thus, it describes the results of the virtual navigation of the robots in $\mathcal{S}_1$ and $\mathcal{S}_2$ without a specific real-world task, assessing whether the navigation goal in Equation 5.4 is fulfilled or not. Focusing on the one-dimensional communication space $\mathcal{S}_1$ first, Figure 5.7A shows the results of the virtual navigation with ten virtual landmarks (red stars) and robots (white circles), and in three different and independent executions (horizontal lines). In this case, the virtual navigation is perfectly accomplished according to the convergence criterion in Equation 5.4. In addition, Figure 5.7B depicts the results in $\mathcal{S}_2$ using 20 virtual regions and 20 robots.

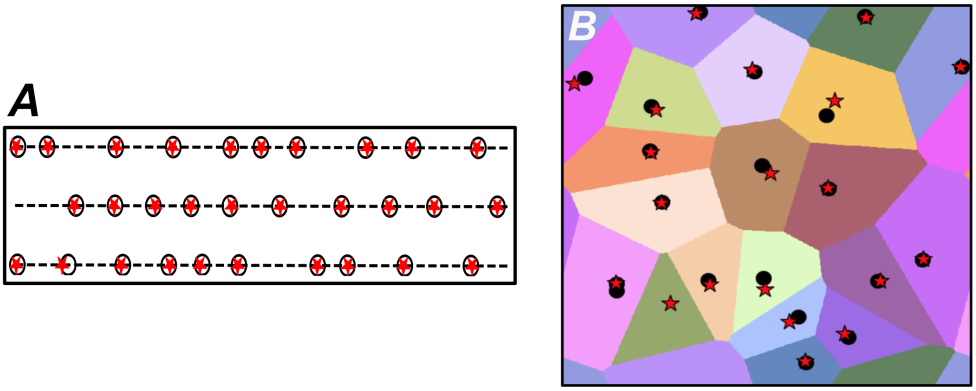


Figure 5.7: Results of communication module in isolation, considering a single trial. (**A**) Results of three independent simulations of the virtual navigation in the one-dimensional communication space $\mathcal{S}_1$ with 10 virtual landmarks and robots. (**B**) Results of the virtual navigation in the two-dimensional communication space $\mathcal{S}_2$ with 20 virtual landmarks and robots. Red stars indicate the coordinates of the virtual landmarks and circles represent the communication states of the robots.

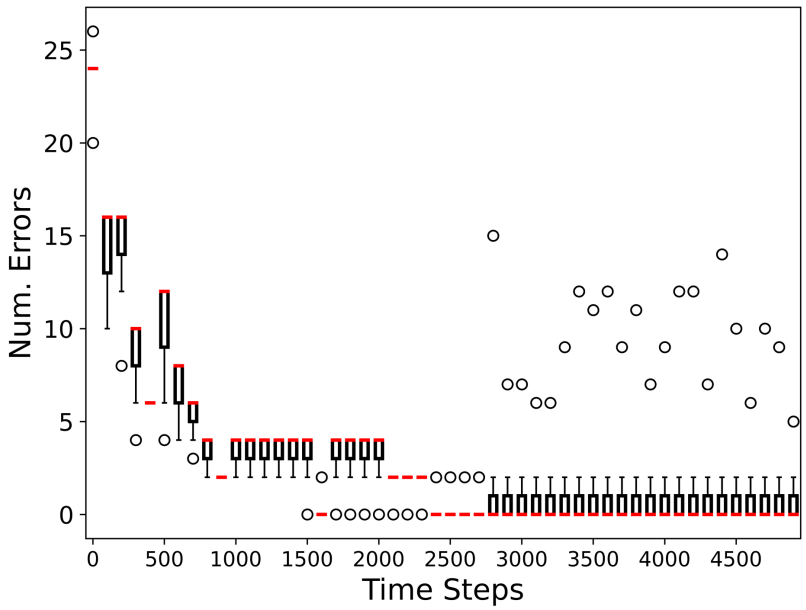


Figure 5.8: Time evolution of errors of the communication module in isolation, considering multiple trials. Boxplot showing the evolution of the number of errors in the virtual navigation as simulation time marches. An error occurs when a robot goes to a virtual region that is already occupied by others.

Figure 5.8 shows the evolution of the number errors as the simulation time steps increase, in a swarm of 30 robots, and using $\mathcal{S}_2$ as the communication space. Every 100 time steps, the distribution of errors is summarized as a box plot using 50 independent trials. The convergence time to reach a median error (red segments) of 5 is about 700 time steps and to obtain a median error of 0 is around 2400 iterations. Moreover, the steady state solution

presents an stable median value of 0 and low dispersion.

5.4.2 Aggregation in groups

To assess the performance of the swarm in the aggregation in groups task, Figure 5.9 shows the positions of the robots in the last frame of four simulations with 20 robots and with diverse number of groups.

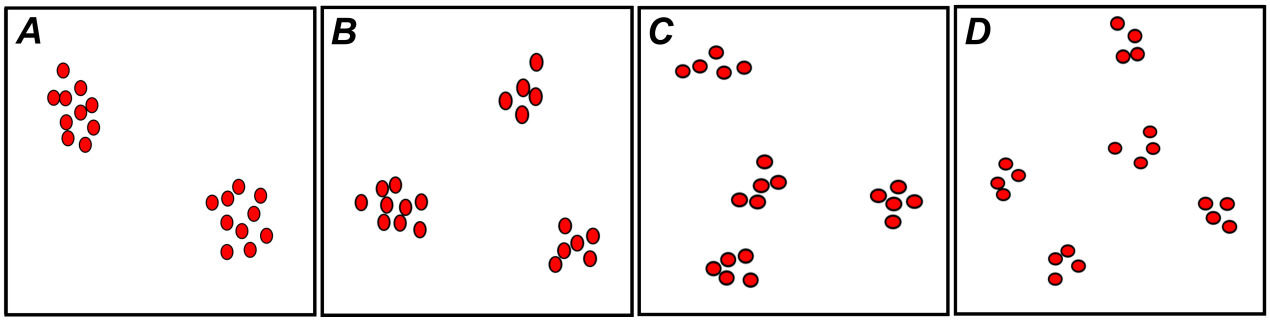


Figure 5.9: Results of the aggregation in groups with swarms of 20 robots. (**A**) Aggregation in two groups of ten robots in each group. (**B**) Aggregation in three groups with irregular sizes of 5, 6 and 9 robots. (**C**) Aggregation in four groups of five robots. (**D**) Aggregation in five groups of four robots.

In all of the depicted simulations, the aggregation in groups is correctly solved. It should be highlighted that the robot swarm is capable of allocating and distributing its members to the different groups to fit the target group sizes. Robots aggregate as near as possible to the center of mass of the corresponding cluster. Moreover, agents also tend to escape from robots in other groups to avoid cluster fusion. The swarms of all the simulations of Figure 5.9 were initialized randomly as a unique group. It is the communication through the virtual space the mechanism that is responsible for converging to the final solution.

5.4.3 Formation control

The swarm formation task is also solved with promising results. Figure 5.10A-E illustrates the convergence process to four different formation geometries, showing the trajectories and the initial and final formations. In all the representations, the colors of the circles indicate the time instant of the corresponding formation according to the time line in the color bar of Figure 5.10G.

Additionally, the short black lines denote the heading orientation of the robot. All the formations are reached with high accuracy, starting from randomly selected robot positions. Moreover, Figure 5.10F goes an step beyond by presenting a simulation in which the swarm formation is switched in real time thrice. The desired formation geometry is fixed by explicitly updating the Δ_i vectors of the primitive functions (see Section 5.3.2). In real-world use cases, this primitive update can be tele-operated or programmed. In addition to the formation switching, Figure 5.10F also shows the coordinated movement of the swarm, while preserving the converged formation. Recall from Section 5.3.2, that the aim of the primitive functions is

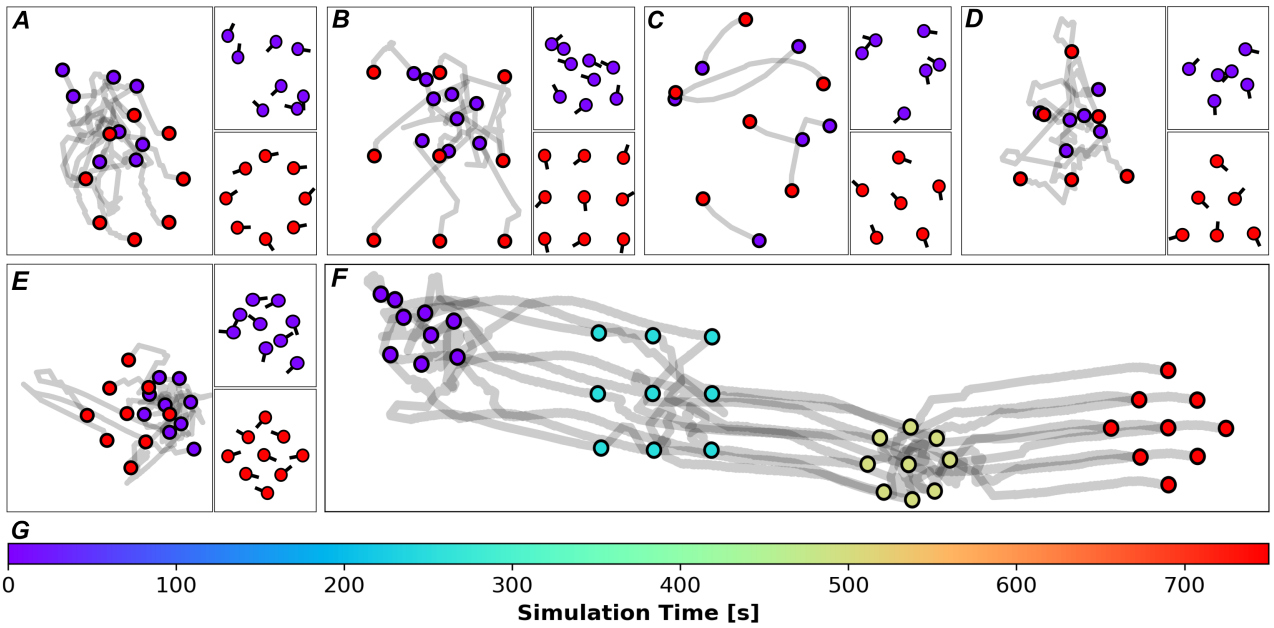


Figure 5.10: Results of the swarm formation task. (**A** to **E**) Results for five different formation geometries, showing the trajectories of the robots (left), a zoom on the initial positions of the robots (upper right) and a zoom of the final formation results (bottom right). (**F**) Swarm switches among three different formations in the same simulation. Moreover, it also shows a coordinated navigation preserving the desired formation. (**G**) Color bar of the simulation time line. In all the figures, the colors of the circles denote the time when the robot was in that position, according to the color bar.

to drive the robots towards the formation points $\{\mathbf{c}(t) + \Delta_1, \ldots, \mathbf{c}(t) + \Delta_N\}$. Thus, provided that the swarm has successfully converged to the desired formation, the entire swarm can be controlled by simply adding some offset or bias to the estimated center of mass. In the case of the coordinated navigation of Figure 5.10F, the primitive functions are modified as follows:

$$f_i \longrightarrow \text{“Go to coordinates } \mathbf{x}_i(t) = \mathbf{c}(t) + \begin{pmatrix} \epsilon \\ 0 \end{pmatrix} + \Delta_i\text{”}.$$

5.4.4 Foraging

The results of the foraging experiment are presented in this section with a swarm of 40 robots using four different graphics with complementary information.

Firstly, Figure 5.11 illustrates the distribution of roles using a ternary plot. The three axes are the proportion of robots foraging from *food 1*, *food 2* and the proportion of agents either recharging battery or in the nest. The size of the points indicates the number of samples corresponding to the given proportions. Even though there is some dispersion, the larger points are gathered around the optimal solution:

$$(\textit{Food 1}, \textit{Food 2}, \textit{Nest+Battery}) = (0.25, 0.25, 0.5).$$

On the right side, Figure 5.11 also shows a zoom of the points inside the highlighted rectangle

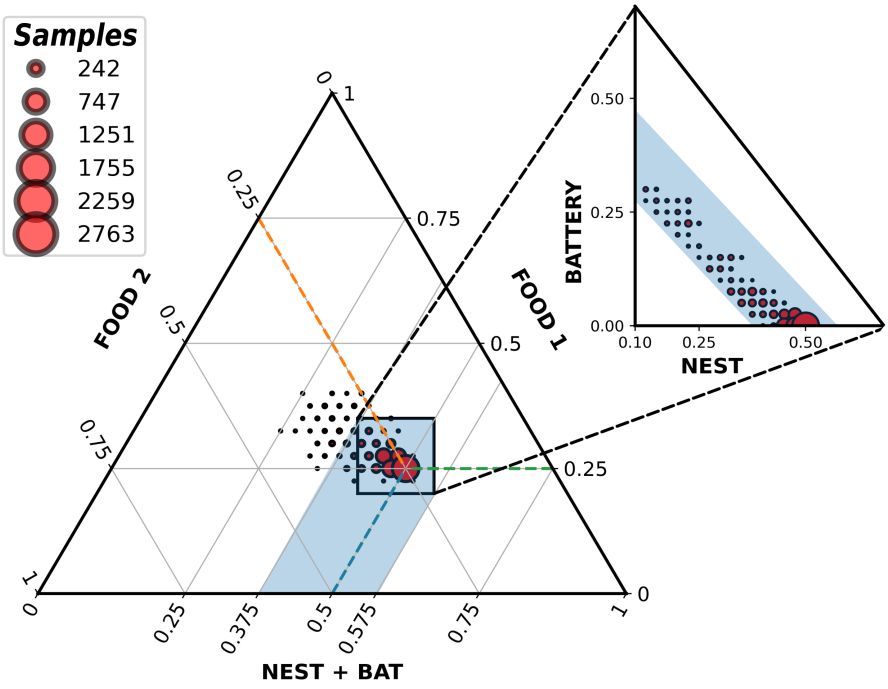


Figure 5.11: Ternary plot showing the frequency of the proportions of robots engaging in the different foraging roles. The three axes, left, right and bottom, correspond to the proportions of robots foraging from food area 1 (right), food area 2 (left), and the proportion of the agents either recharging their batteries or staying in the nest. The size of the circles indicates the number of times that the given proportion of robots engaging in each role is repeated across simulations. For instance, the most repeated division of labor (highlighted in the figure with dashed lines) corresponds to the optimal solution of 25% of robots foraging from food 1 and another 25% foraging from food 2.

of the ternary plot, depicting the proportion of robots waiting inside the nest versus charging the battery. In the most common scenario, 50% of the swarm is waiting in the nest, while there are no robots running out of battery. As the number of robots recharging the battery increases, the individuals in the nest is diminished in order to keep the other 50% of foragers constant.

Figure 5.12 shows a three-dimensional representation of the foraging experiment. The horizontal plane at the bottom represents the flat arena where robots can move. Additionally, the surfaces illustrate the density of robots that have visited the (x, y) coordinates of the plane underneath at any time step of the simulation. There are four density surfaces with colors, blue, green, orange and red, corresponding to the primitives f_1, f_2, f_3 and f_4, respectively (see Section 5.3.3). Notice that the density surfaces perfectly adjust to the desired behavior of the robots (see also Figure 5.6C for the location of the different parts of the arena). For example, the blue surface corresponds to the positions of the robots that are staying in the nest, while the green surface illustrates the round trips of the robots foraging from one of the food sources.

The results presented in Figure 5.11 show that the robots are able to successfully self-organize

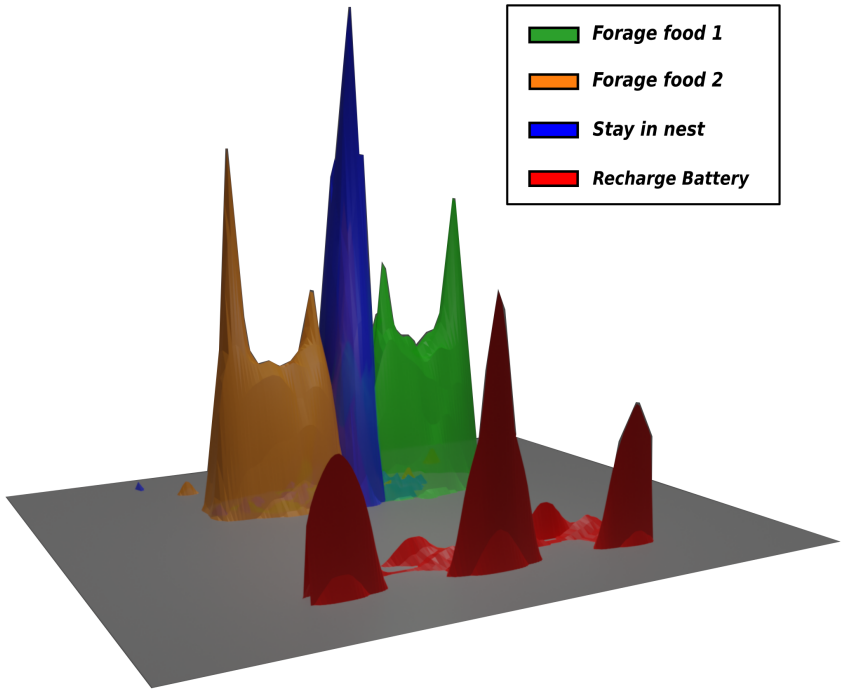


Figure 5.12: Surface illustrating the arena zones (XY plane) visited more frequently by the robots depending on their role (see legend).

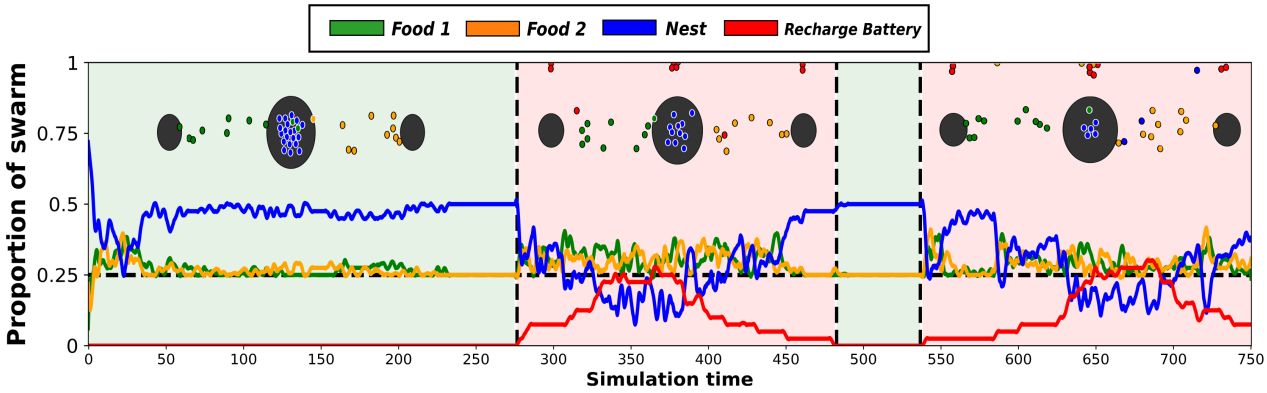


Figure 5.13: Evolution of the proportion of robots engaging in each role as simulation time elapses. The two types of time windows, green and red, represent the absence or presence of at least one robot charging its battery.

and allocate the different roles of the experiment using the communication module. However, there was no temporal information in order to assess the timings of the role allocation. Therefore, Figure 5.13 depicts several time series showing the proportion of robots executing each role as the time elapses. There are two different types of time windows that alternate during the simulation, corresponding to the scenarios with all the robots with recharged

battery (green) and at least one robot running out of battery (red). Notice that in the green time windows the green and orange curves, representing the proportion of robots foraging from each food source, converge to the optimal value of 0.25. When the number or robots without battery increases, the proportion of robots foraging fluctuate more notably. Nonetheless, the robots in the nest (blue curve) are able to absorb the loss of foragers, with the aim of preserving the blue and orange curves with the constant value of 0.25.

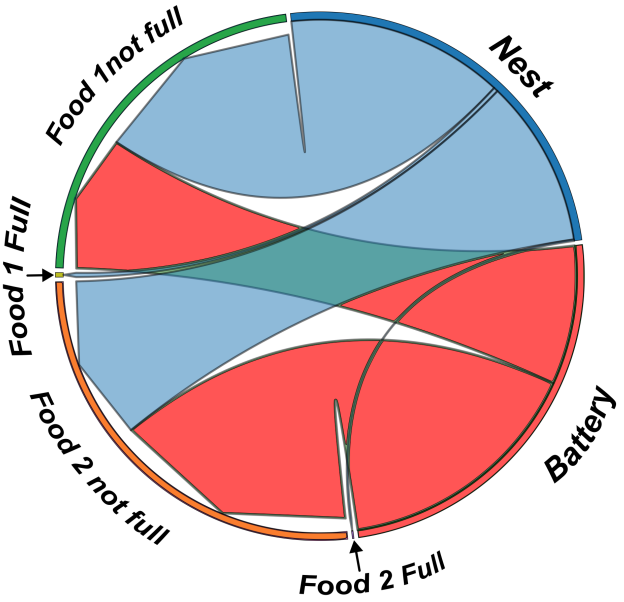


Figure 5.14: Circular plot showing the transitions among the roles engaged by the robots in the foraging task. Specifically, it shows the transitions between the roles *Nest* and *Charge battery* and foraging each of the food sources. The width of the transition arrows indicate the number of times that a robot switched between the corresponding roles. Additionally, it is also included the distinction between transitioning to the role "*Forage from area 1 or 2*" when there are enough robots already engaging in that role ("*Food full*") or when there are still foragers needed ("*Food not full*"). The target scenario is the one where robots switch to a foraging role that is not full.

Finally, Figure 5.14 complements the previous figures by showing the role transitions during the experiment simulations. Essentially, the graphic shows the percentage of robots that transition from waiting in the nest (blue) or from recharging their batteries (green) to foraging from each food source. In each case, a differentiation is made between the transitions to a foraging role that does not require any other worker ("Food 1 or 2 full"), and that requires new foragers ("Food 1 or 2 not full") to reach the target proportion of 0.25. The percentage of the former transitions, that are undesired and in most cases spurious, are negligible compared to the latter ones when the previous role is "stay in nest" and zero when the previous role is "recharge battery".

5.4.5 Scalability assessment

One of the most important aspects of SR systems is their scalability property or their capacity to maintain the performance as the swarm size increases. This subsection concludes the experimental results by assessing that the robot swarm fulfills this property in the three experiments.

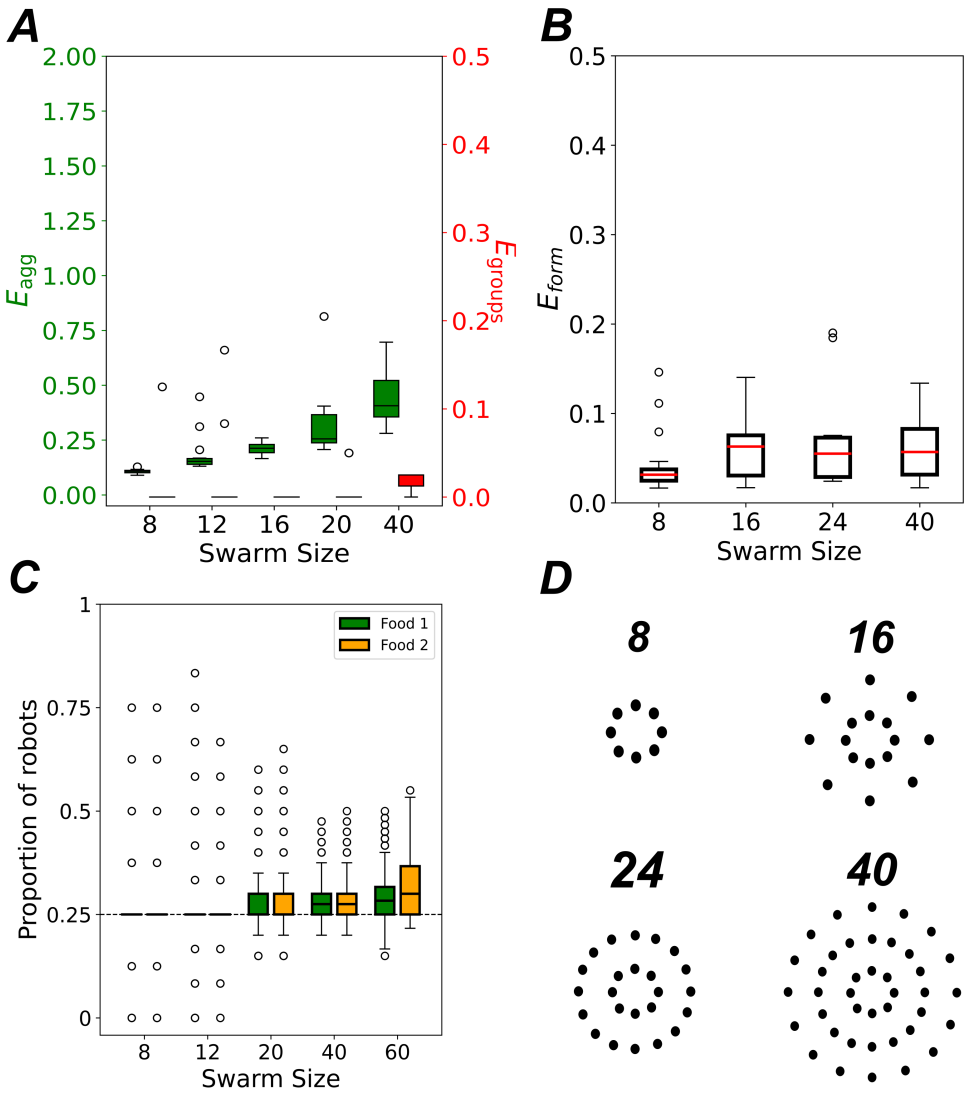


Figure 5.15: Scalability results for the three experiments. (**A**) Boxplot showing the scalability in the aggregation in groups, as the evolution of the errors metrics E_{agg} and E_{groups} as the swarm size increases. (**B**) Scalability results in the formation task, each box shows the distribution of the errors E_{form} for a given swarm size. (**C**) Scalability results in the foraging task. For each swarm size, the green and orange boxes summarize the distributions of the proportion of robots foraging from each food source. All the boxes in (**A** to **C**) are constructed using samples of 50 independent simulations. (**D**) Target formations for each swarm size in (**B**).

In the aggregation in groups, the error metrics in Equation 5.18 are considered,

$$\left.\begin{aligned} E_{\text{agg}} &= \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{x}_i(T)-\mathbf{c}_i(T)\right\|_2 \\ E_{\text{groups}} &= \frac{1}{G}\sum_{k=1}^{G}\max\left(\frac{G_k(T)-G_k^*}{G_k^*},\,0\right) \end{aligned}\right\} \tag{5.18}$$

so that they separately evaluate the aggregation and the formation of groups of correct sizes, respectively. E_{agg} is the average distance between the robot physical positions $\mathbf{x}_i$ and the center of mass of their group $\mathbf{c}_i$, at the last time step of the simulation T. Besides, E_{groups} measures the average number of robots that wrongly aggregated in a group that already has the target size G_k^*.

Figure 5.15A summarizes the results of both error metrics for swarm sizes up to 40 robots. In all the cases, the robots aggregate in four groups of $N/4$ robots each. In the case of E_{groups}, the number of errors for all swarm sizes tend to gather near zero, demonstrating the correct performance of the communication module. Even though a slight degradation of E_{agg} is visible, the errors are still low for the highest swarm sizes. This increase in E_{agg} is elevated because the avoidance of obstacles is prioritized over the aggregation, which is more noticeable in the large swarms.

For the second experiment, the formation of geometries, the error metric is defined in Equation 5.19,

$$E_{\text{form}} = \frac{1}{N}\sum_{i=1}^{N}\min_{j\in\{1,\ldots,N\}}\left\{\left\|\mathbf{x}_i^*(T)-\mathbf{x}_j(T)\right\|_2\right\}, \tag{5.19}$$

which essentially averages the distances between the target points in the formation $\mathbf{x}_i^*$ and the positions of the closest robot, $\mathbf{x}_j$, to the formation points, at the last simulation time instant T. Figure 5.15B collects different box plots summarizing the error E_{form} for diverse swarm sizes. Each swarm size corresponds to the target formation depicted in Figure 5.15D. An increment in the swarm size causes a increase in the median value and dispersion of the error samples. However, such error growth is significantly low, demonstrating the scalability of the system.

Lastly, Figure 5.15C illustrates the scalability in the foraging experiment. In this task, the results are directly evaluated based on the proportion of robots foraging from each food source, which optimal values are 0.25. The summary of statistics shown by the box plots reveals that the proportions of foragers from both food areas is highly close to the target proportion of 0.25.

5.5 Discussion

The results presented in Section 5.4 showed that the proposed communication module can be used to solve three notably different SR tasks in an efficient and scalable way. Moreover, exactly the same evolved communication controller, that leads to some desired virtual navigation in the communication space, was transferred and employed in all the tasks. The only part of the

entire SR system that was reconfigured in each task was the robot controller set of primitives, primitive selector and, in the case of the foraging task, the virtual region priorities. In all the tasks, promising results were obtained both in terms of performance and scalability, which goes an step further towards standarized and task agnostic communication systems in SR.

In Figure 5.8, the distribution of errors and convergence time was assessed in a division of labor or role allocation using a swarm of 30 robots, and using the proposed communication module. It was demonstrated that the system can achieve a median error of 0 in about 2700 time steps and a median error around 5 in 700 iterations. These results outperform the convergence time shown in [244], which is about 10,000 time steps. It was verified that the framework is scalable in the three proposed SR tasks. Specifically, it was shown that the aggregation in groups, the formation control, and the foraging tasks can be solved by robot swarms up to 40, 40, and 60 robots, respectively. In addition to the flexibility and scalability analyses, robustness is mostly evaluated in this work in the formation experiment, in which the formation architecture is changed at runtime, using three different formations along the same simulation (see Figure 5.10F). It was demonstrated that the swarm is capable of reacting to this unexpected change and adapt to the new formation topology. This experiment is a first approach to assess the robustness of the system in real-world missions (e.g. search and rescue) that demand formation changes due to unexpected perturbations (such as changes in the terrain or in the mission goal). However, as it is out of scope in this PhD Thesis, a deeper analysis of the robustness of the system in real robots and in a wide-range of unexpected scenarios is left as future work. The same previously evolved communication module was successfully applied to all proposed tasks.

A single CTRNN-based communication controller was evolved only once, and successfully used across all three tasks. This flexibility was possible because of the *task-agnostic* evolution and the high degree of decoupling between the communication module and the robot controller. Nonetheless, a task-specific configuration of the robot controller (particularly the set of primitives, the primitive selector, and the priority selector) was required for each experiment. The evolution of a communication controller that generates a virtual navigation that, upon convergence, satisfies Equation 5.4 provides a highly suitable solution for SR tasks that require a precise division of labor and dynamic role allocation. In the aggregation task, the division of labor is produced through the allocation of robots to groups desired sizes. In addition, the roles that are distributed through the communicative process in the formation task are the positions in target formation geometries. That is, the robots negotiate using the communication module on the spots of the formation that each one should occupy. Finally, foraging, inherently linked to division of labor, also demonstrates the effectiveness of the evolved communication module. Specifically, the robots dynamically allocate the required foraging roles (foraging from different food sources or staying in the nest) among the swarm members. The evolved communication enables an optimal convergence to the specific number of robots engaging in each role. Moreover, it allows role switching to cover the absence of other robots whose battery was unexpectedly depleted.

The main limitations of the overall proposed system are twofold. First, the performance and robustness of the communication module would be potentially limited in constrained and low-range communications. In such constrained scenarios, the control and communication

are produced locally, and if the swarm is too large or the distances between robots grow, it would result in the generation of clusters of robots. For example, a formation control of 30 robots can result in three different smaller swarms of 10 robots each, that would behave as separate swarms from each other group. Clearly, once the robot swarm is split into clusters, it is impossible to deliberately aggregate and re-unite the whole swarm unless some long-range communication is used. Constrained communications are specifically critical in tasks similar to the proposed foraging, in which robots need to know the communication states of other robots that may be too far away (e.g. one robot in the nest and another robot in a food area). In this case, this limitation is not considered in depth in this PhD Thesis and it should be revisited in future works. Besides, scenarios with noisy communications with significant signal interference can be an issue specifically when some robot's communication state being transmitted is near to a boundary between two virtual regions (e.g. the robot is inside virtual region $\mathcal{R}_A$ but close to $\mathcal{R}_B$). Such noisy transmission could potentially cause that other hearer robots wrongly receive $\mathcal{R}_B$ as the virtual region of the speaker agent. However, this limitation is very specific and, when produced, can be rapidly solved as the communication state of the speaker robot converges to the inner area of the corresponding virtual region. The second limitation to be highlighted is related to the robot primitives of the proposed experiments. Essentially, it concerns the use of high-level sensors that provide the robots with information related to their global positions in the environment. A solution to this problem could be addressed by means of using multi-agent localization and social odometry [150, 57] to estimate the positions, at the cost of losing some accuracy due to the estimations.

5.6 Summary and conclusions

This chapter proposes a self-organized and distributed communication module that is evolved only once and that can be transferred and used in many independent SR tasks. Therefore, this represents a first approach to SR communication with *strong transferability*, an important challenge in the evolution of communication in SR. The pillar of the proposed system is the use of a virtual state space, denoted as the *communication space* where the robots, represented as particles in this virtual space, can freely alter their communication states. The communication space is composed of a fixed number of virtual regions whose limits are defined by centroid points called *landmarks*. Each virtual region is mapped into a primitive function that defines some basic behavior in the physical space (e.g. explore the surroundings or approach a light source). Consequently, robots perform a collective virtual navigation in the communication space in order to reach certain regions and execute the associated primitives. It should be noted that communication among robots is produced because, by acknowledging the communication states of its neighbors, a robot can know their current primitives and, thus, their roles and plans in the physical environment. This is the pillar for solving, for instance, distributed role allocation problems. The flexibility and transferability of the communication module are demonstrated in three different SR tasks: (i) aggregation in groups, (ii) formation of swarm geometries, and (iii) foraging. Moreover, it was shown that the communication module, which was evolved in isolation and only once, could be used to successfully address all the proposed applications in a scalable way.

There are several extensions of this study that should be addressed in the future. Firstly, the

communication module should be revisited to tackle tasks with very constrained and low-range communications. Additionally, the SR tasks proposed in this Chapter can be improved by estimating the relative positions of the robots using techniques, such as multi-agent localization or SLAM. Finally, the number of virtual landmarks, their coordinates, and the semantics and grounding associated to them (i.e. the primitives) are defined by the researcher beforehand. It can be viewed as some innately acquired semantics and lexicon. Therefore, a highly promising extension of this work would consider the emergence and self-organization of such semantics and lexicon through some cultural evolution and learning processes. Specifically, these techniques can be potentially used to learn from zero the communication space topology, the virtual landmarks, and the primitive associated to each virtual region. The communication space would be initialized with a single virtual landmark with no associated behavior primitive. Thereafter, new virtual regions would incrementally emerge, with an associated set of primitive behaviors being automatically designed for the target task.

Chapter 6

Emergence of perceptually grounded compositional lexicons

6.1 Introduction

Communication, as a fundamental aspect of any SR system, has been traditionally considered from the ER perspective. Nonetheless, cultural evolution, that is one of the main properties of complex human-like languages [15, 16], has not beed sufficiently explored in SR [11]. Cultural evolution allows the preservation and growth of communication across generations via cultural transmission and learning. Moreover, it has been hypothesized that the emergence of language is possible when biological evolution, individual learning, and, very importantly, cultural transmission processes, that operate at different time scales, are intertwined in a closed loop [184, 161, 185]. In previous chapters, different communication systems of diverse complexities, semantics, and mechanics were obtained using evolutionary computation. However, all of them fall under the category of signaling communication, because most of the essential properties of complex language-like communications are missing (see Chapter 3.2). In this chapter, cultural evolution, language games, semiotics, and lifetime learning are considered as the basic processes to reach communications with higher levels of complexity. Under this frame, the self-organization and sharing of a common lexicon to refer to relevant objects, events, or concepts in the environment, which is of essence to achieve language-like communications, is addressed. The emerged common lexicon exhibits most of the required properties of human-like languages, namely, arbitrariness, compositionality, and duality patterning.

The proposed experiment is the following: the robots are embodied and situated in an environment with multiple geometric objects of different types scattered around. The robots navigate and explore the environment, eventually discovering the referents on which symbols will be grounded based on their own experience. The perceptual grounding process can only be accomplished through individual exploration, and through local communication with other robots in the swarm. In the latter case, neighbors engage in language games to negotiate on the social component of the semiotic symbols (the forms). This greatly diverges with most similar works reviewed in Chapter 3, because language games are generally played by pairs of

robots in the population regardless of their distance and, in some cases, without exploration of the environment and discovery of the referents. Even though there have been previous studies with simulated and real mobile robots that attack the SGP as a whole, the number of agents is typically restricted to two mobile robots [14, 221, 205]. Studies with more robots and larger lexicons are generally either based on the Talking Heads Experiment [196, 236, 195] (robots are static and there are only two simultaneous agents embodied in the same environment) or on abstract environments with a lack of agent embodiment [201, 203]. In the field of SR, studies are limited either to using language games as an optimization tool to solve certain SR tasks (e.g. aggregation) [240, 241, 53], or to the convergence of lexicons of few words [213]. Lexicon formation and sharing has not been sufficiently addressed in SR from the perspective of semiotics, making a distinction between private meanings and shared forms. For a detailed review on the emergence of language and the acquisition and self-organisation of a shared and perceptually grounded lexicon the reader is redirected to Section 3.5.

6.2 The experiment

6.2.1 Description of the experiment

A simulated swarm of two-wheeled mobile robots is situated and embodied in a 10 m $\times$ 10 m square arena (Figure 6.1). In addition to the simulated robot swarm, multiple geometric objects of diverse colors and shapes are scattered along the arena. The color of the objects can be either red, blue, or green (of different tones and saturations), whereas the possible shapes are cubes, spheres, and pyramids. In addition to color and shape, the objects are also characterized by a quality score, in the range $[0, 1]$. The quality score can be associated with the level of imperfection of the given shape, e.g. due to roughness, distorted shape, and asymmetry. A quality score near one indicates that the corresponding geometric object is nearly perfect.

The aim of the experiment is to reach the emergence of a common compositional lexicon along the robot swarm to name the different geometric objects. Semiotics and language games (see Chapter 3 for more details) are used to achieve this task, so that robots self-organize a set of solid and perceptually grounded semiotic symbols. The grounding and sharing of symbols co-operates with the main robot control, which is a basic obstacle avoidance module that allows the robots to explore the arena without colliding with objects or other robots. Agents can only ground their meanings on the nearby objects perceived by their sensors. Furthermore, robots can only negotiate the forms (words) of their semiotic symbols with other neighbors in the short range. Consequently, a proper exploration and embodiment in the environment is of essence.

Robot simulations are done using the `pybullet` library [250] for real-time collision detection and multi-physics simulations. The simulated robots mimic the epuck robot [107], which is a two-wheeled mobile robot whose kinematics are based on a differential drive system. The main robot controller is a simple obstacle avoidance routine that navigates straight forward until an obstacle is detected. Upon obstacle detection, the robot rotates either clockwise or counter-clockwise depending on the direction of the obstacle. Obstacles can be either walls,

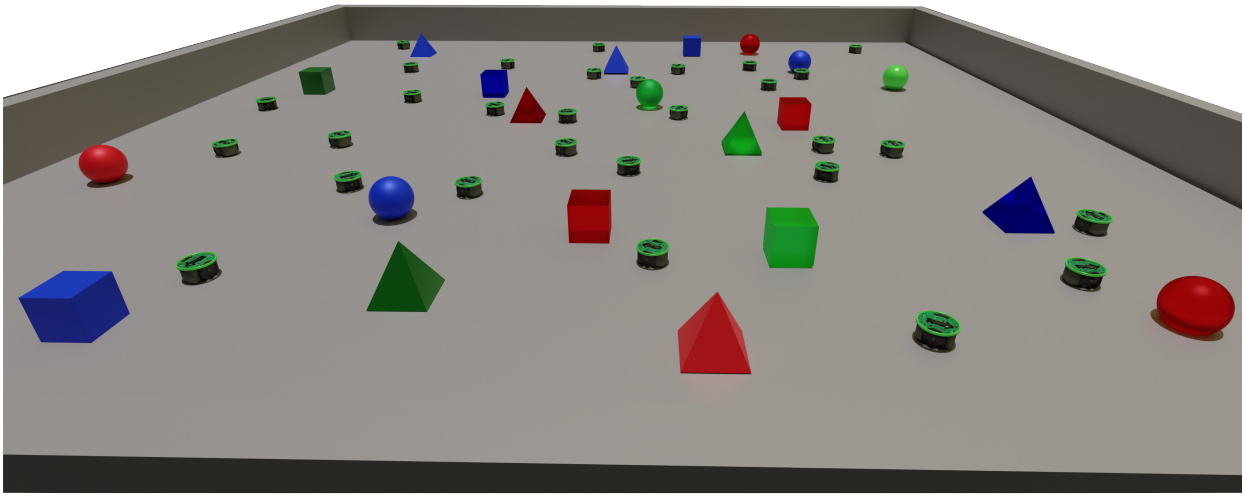


Figure 6.1: Representation of the arena where the experiment takes place. It is composed by a square arena, a swarm of simulated e-puck robots, and multiple geometric objects of different colors, shapes, and qualities (quality represented as the glossiness of the objects in the rendered image).

objects, or other robots, and they can be detected using eight IR proximity sensors positioned along the robot's perimeter. As it is not the focus of this chapter, the sensors that perceive color, shape, and quality score of the objects in the arena are abstracted by means of the front camera. These sensors simplify the corresponding sensing and processing phases by providing noisy estimates of the readings. The initialization of the robots and geometric objects in each simulation is random and uniform along the entire arena (constraining the sampling space to non-overlapping positions). Orientations are also initialized randomly in the the range $[0, 2\pi)$.

6.2.2 Sensing of the environment

Each robot perceives the local environment by means of a set of sensor channels. Each sensor captures different properties of external objects, i.e. color, shape, and quality. These properties are perceived by means of a frontal camera and a subsequent image processing stage to extract the color, shape and quality features (Figure 6.2). The overall feature vector, resulting from the aggregation of all the sensor readings, is represented as the tuple in Equation 6.1,

$$\mathbf{x} = (r,\ g,\ b,\ p_{\square},\ p_{\bigcirc},\ p_{\triangle},\ \mathcal{Q}), \tag{6.1}$$

where r, g, and b reflect the amount of red, green and blue present in the target object, and $p_{\square}$, $p_{\bigcirc}$, and $p_{\triangle}$ are probability estimates of such object being either a cube, a sphere, or a pyramid, respectively. The component $\mathcal{Q}$ is the quality score associated with the targeted object. All the components of $\mathbf{x}$ are constrained to the range $[0, 1]$. Color, shape and quality are projected into the robot's sensory apparatus with a high degree of variability. For instance, the exact same red sphere is perceived in an slightly different way (both in color and shape) by multiple robots due to their divergences in their sensors, and their positioning relative to the object, among other factors. Similarly, the same agent measures differently two objects of the same color and shape (equivalent in theory).

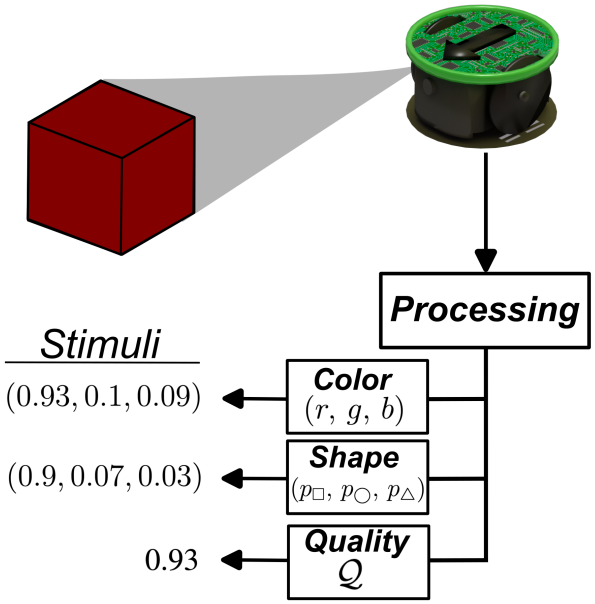


Figure 6.2: Sensing of the geometric objects scattered in the arena. A robot can sense external objects located in its close surroundings via a front camera. A computer vision pre-processing module extracts several features of such object to produce the final stimuli vectors. The extracted features are the RGB color, the shape encoded as a three-dimensional probability vector, and a quality scalar.

6.2.3 Iconization and discrimination

A discrimination game is used as the method to implement the discrimination and iconization of each robot. Specifically, the sensory space of each robot is incrementally partitioned into a set of regions, based on the agent's interaction with the environment. Each of these regions are associated to a single point, which is usually called *prototype* or *pattern*, used as a representative for the whole region. These prototypes are, in turn, directly associated to the iconic representations of the categorization process. Under this premise, the categorization and discrimination of continuous stimuli reduces to finding the region of the partitioned space where it lies, and selecting the representative prototype as the resulting icon. Moreover, a process of adaptation is applied to modify the location of the prototypes based on newly acquired information about the external world. This adaptive improvement eventually leads to the acquisition of the sought invariant and distinctive properties of the underlying categories, provided that the agents exploration of the real world is sufficient.

A holistic input space is defined as $\mathcal{X}$, so that the perceived sensory stimuli, $\mathbf{x}$, belongs to this space. Similar to the conceptual spaces used in [195, 197, 198], the holistic space $\mathcal{X}$ is decomposed into partial spaces $\mathcal{X}_s$, each one corresponding to one sensory unit (e.g. $\mathcal{X}_{color}$ of dimension three corresponds to the color sensor). Thus, the holistic input space is formulated as the union of all the partial input spaces as in Equation 6.2,

$$\mathcal{X} = \bigcup_{\forall s} \mathcal{X}_s. \tag{6.2}$$

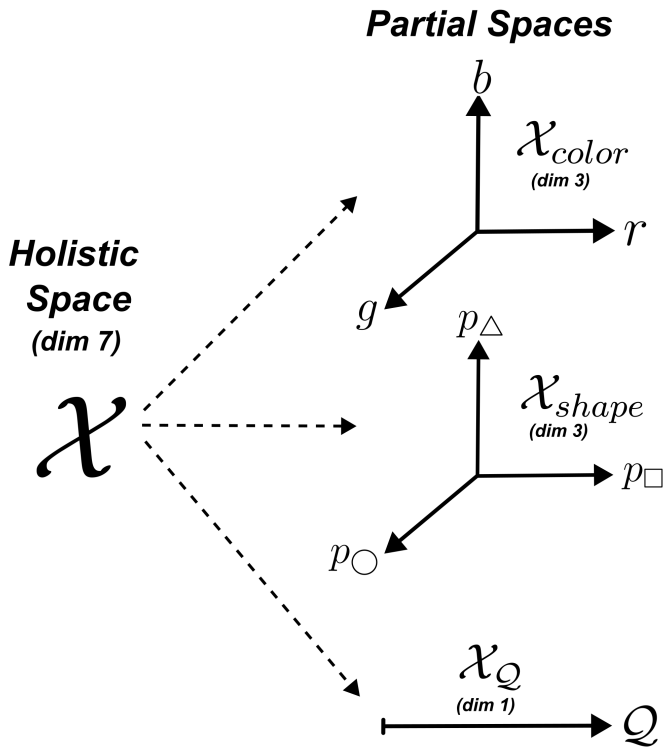


Figure 6.3: Decomposition of the holistic input space into partial input spaces. The holistic input space, $\mathcal{X}$, results from the union of multiple partial input spaces, each one corresponding to a separate sensor channel. Thus, the figure depicts the decomposition of $\mathcal{X}$ into three independent partial spaces ($\mathcal{X}_{color}$, $\mathcal{X}_{shape}$, and $\mathcal{X}_\mathcal{Q}$) corresponding to the color, shape, and quality sensor units.

Likewise, $\mathbf{x}_s \in \mathcal{X}_s$ is defined as the feature vector read from sensor s (e.g. the color sensor), and $\mathbf{x} \in \mathcal{X}$ as the holistic feature vector resulting from the concatenation of all the partial vectors $\mathbf{x}_s$. Figure 6.3 depicts the decomposition of $\mathcal{X}$, of dimension seven, into three partial spaces, $\mathcal{X}_{color}$, $\mathcal{X}_{shape}$, and $\mathcal{X}_\mathcal{Q}$.

Let $\xi_k,\ k \in \{1, \ldots, K\}$ be the *prototypes* or *patterns* that result from the iconization and discrimination processes. Upon convergence, these prototypes desirably represent invariant and unique properties of distal objects. The prototypes ξ_k can exist both in the holistic input space $\mathcal{X}$ or in the partial subspaces $\mathcal{X}_s$. Each partial input vector $\mathbf{x}_s$, perceived by each sensor, is compared to each of the stored prototypes of the corresponding partial subspace $\mathcal{X}_s$. This comparison can be observed in Equation 6.3,

$$\xi_s^{\min} = \underset{\xi_k \in \mathcal{X}_s}{\operatorname{argmin}} \left\{ \|\mathbf{x}_s - \xi_k\|_2 \right\}, \tag{6.3}$$

where the closest prototype $\xi_s^{\min} \in \mathcal{X}_s$ to the perceived stimuli $\mathbf{x}_s$ is calculated. Additionally, the distance between $\xi_s^{\min}$ and $\mathbf{x}_s$ is defined in Equation 6.4,

$$d_s^{\min} = \left\| \mathbf{x}_s - \xi_s^{\min} \right\|_2. \tag{6.4}$$

If the computed distance to the closest prototype is greater than a fixed threshold θ_ξ (which implies that $d_s^{\min} > \theta_\xi$), then the perceived stimuli is not sufficiently similar to any of the

stored prototypes, and a new prototype ξ_s^{new} is created with the current value of $\mathbf{x}_s(t)$. Alternatively, if $d_s^{\min} \leq \theta_\xi$, then $\mathbf{x}_s$ is associated to the closest stored prototype because of its similarity. Nonetheless, the selected prototype is subject to an adaptation process to update its coordinates, based on the new perceived information about the world (Equation 6.5).

$$\xi_s^{\text{new}} = \begin{cases} \xi_s^{\text{old}} + \lambda_\xi \cdot (\mathbf{x}_s(t) - \xi_s^{\text{old}}), & \text{if } d_s^{\min} \leq \theta_\xi \\ \\ \mathbf{x}_s(t), & \text{if } d_s^{\min} > \theta_\xi \end{cases} \tag{6.5}$$

The first case in Equation 6.5 shows the adaption of some stored prototype ξ_i^{old} based on the newly acquired information about the world. This update essentially moves the location of the *prototype* towards the new stimuli $\mathbf{x}_s(t)$, and based on some fixed constant λ_ξ that defines the length of the steps.

As an illustrative example, Figure 6.4 considers the partial input space corresponding to the color sensor, $\mathcal{X}_{color}$. In such figure, three *prototypes*, $\xi_{\texttt{red}}$, $\xi_{\texttt{green}}$, and $\xi_{\texttt{blue}}$, corresponding to the referent colors red, blue, and green are depicted. Surrounding each *prototype*, the associated discrimination region is also plotted. Such region implies, for example, that any new color stimuli perceived by the robot that lies on the red volume is categorized as $\xi_{\texttt{red}}$. The volume of the discriminating regions is determined by the threshold θ_ξ. Algorithm 2 summarizes the complete discrimination process acomplished by each robot for each sensor

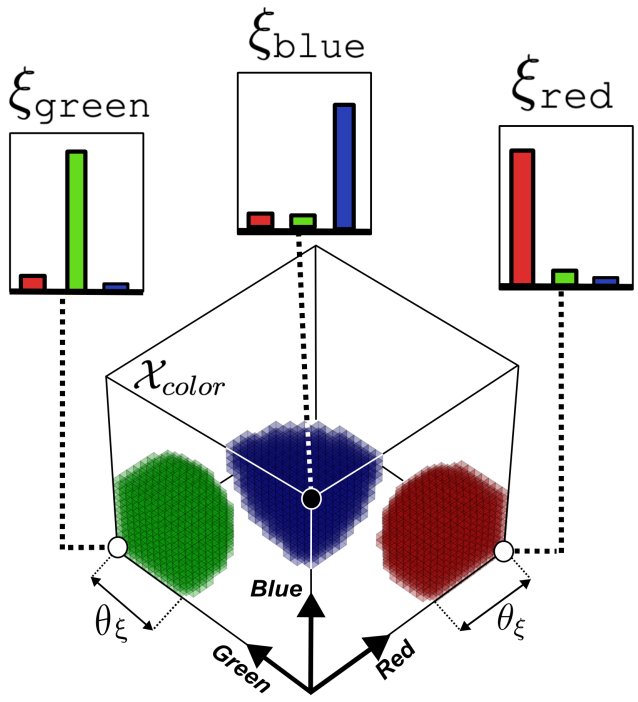


Figure 6.4: Discrimination process in the color partial space $\mathcal{X}_{color}$. The figure shows three *prototypes*, $\xi_{\texttt{red}}$, $\xi_{\texttt{green}}$, and $\xi_{\texttt{blue}}$ that were previously acquired through interaction with the environment and discrimnation games. It also shows the discrimination volumes for each prototype, with boundaries defined by a distance of θ_ξ with the respect to the corresponding prototype.

Algorithm 2 Discrimination process for a given sensor s

procedure $\mathcal{D}(\mathbf{x}_s)$
  $\Xi_s \leftarrow \{\xi_1, \ldots, \xi_K\}$ ▷ List of previously acquired *prototypes* from sensor s
  $d_{cst} \leftarrow \infty$
  $\xi_{cst} \leftarrow \emptyset$
  for $\xi_k \in \Xi_s$ **do** ▷ Compute the most similar *prototype* to $\mathbf{x}_s$
    $d_k \leftarrow \|\mathbf{x}_s - \xi_k\|_2$
    if $d_k < d_{cst}$ **then**
      $d_{cst} = d_k$
      $\xi_{cst} = \xi_k$
    end if
  end for
  if $d_{cst} > \theta_\xi$ **then** ▷ $\mathbf{x}_s$ is not similar enough to any stored *prototype*
    $\xi_{new} \leftarrow \mathbf{x}_s$ ▷ Create new *prototype*
    Add ξ_{new} to Ξ_s
  else ▷ $\mathbf{x}_s$ is associated to some stored *prototype*
    $\xi_{new} \leftarrow \xi_{cst} + \lambda_\xi \cdot (\mathbf{x}_s - \xi_{cst})$ ▷ Update stored prototype with $\mathbf{x}_s$
    Replace old prototype $\xi_{cst} \in \Xi_s$ with ξ_{new}
  end if
end procedure

channel s.

6.2.4 The semiotic symbols

Semiotics and semiotic symbols are considered as one of the pillars to achieve the emergence of a shared and perceptually grounded lexicon in the robot swarm. Recall from Chapter 3, a semiotic symbol is a triad composed by a referent, a meaning, and a form. However, in this chapter, the notion of *semiotic triad* is complemented by *semiotic tetrad*, which augments the constituent parts of a semiotic symbol to four. Specifically, a semiotic symbol is now defined as the combination of a referent, a form, a meaning, and a prototype. The referents are the geometric objects scattered in the arena, and forms are arbitrary three syllable words agreed by the swarm during the semiosis process. Formally, the set of objects (referents) is denoted as $\mathcal{O} = \{o_1, \ldots, o_N\}$, and the set of words (forms) or lexicon as $\mathcal{L} = \{\omega_1, \ldots, \omega_N\}$. The third piece of a semiotic symbol is the meaning, which is defined as an n-dimensional vector $\mathbf{m}$ that belongs to some vector space $\mathcal{M}$. Finally, the prototypes $\Xi = \{\xi_1, \ldots, \xi_N\}$ are iconic representations of the referents (as defined in Chapter 6.2.3), prior to their arbitrary transformation to meanings (through the identification process).

In this PhD Thesis two types of semiotic symbols are distinguished: *elementary*, and *composed* semiotic symbols. *Elementary* semiotic symbols are directly grounded on the most basic external referents, i.e. red, green, blue, cube, sphere, pyramid, high quality, and low quality. Conversely, *composed* semiotic symbols are built on top of the elementary ones using compositional rules (see Chapters 6.2.5 and 6.2.6 for more details). *Composed* symbols are grounded

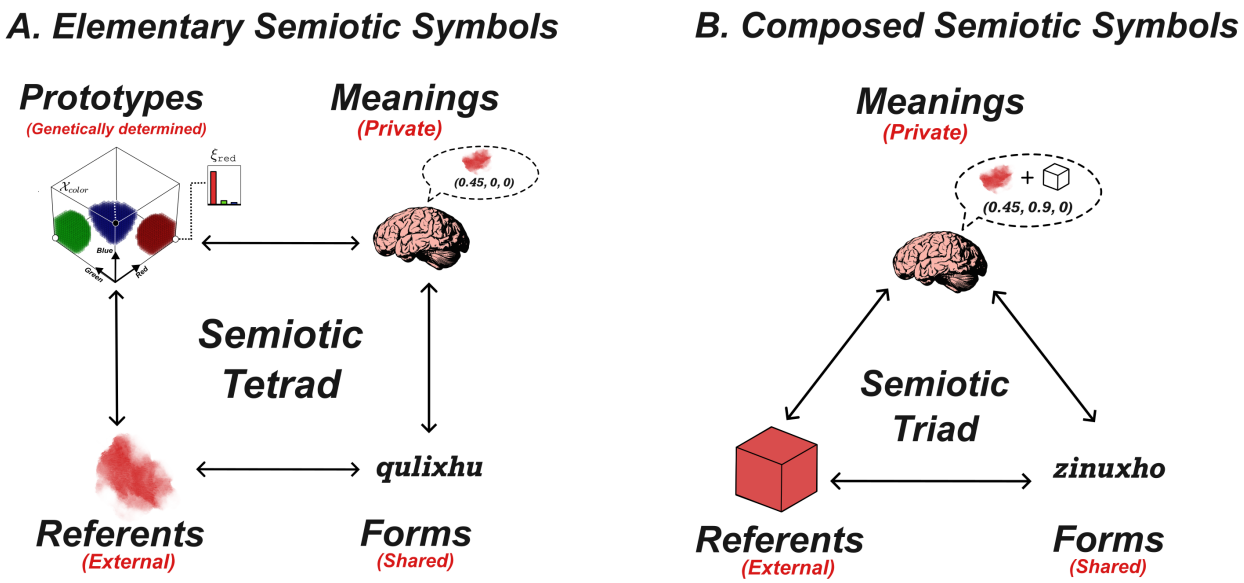


Figure 6.5: Comparison between elementary and composed semiotic symbols. (**A**) Semiotic tetrad of elementary semiotic symbols, showing the referent, the prototype, the meaning, and the form. It depicts an example grounded on the color red. (**B**) Semiotic triad of composed semiotic symbols, showing the referent, the meaning, and the form. It depicts an example grounded on a red cube.

on more complex external referents, such as red squares, green pyramids, or high quality blue spheres. The composed symbols, furthermore, inherit the grounding of the most basic ones that are used for its creation (e.g. the symbol corresponding to red squares, indirectly inherits the physical grounding of the previously developed semiotic symbols of red, and square). Additionally, the concept of the *semiotic tetrad* presented above is considered only for elementary semiotic symbols, whose *direct grounding* compulsorily requires the discrimination and iconization of referents as prototypes. Alternatively, a semiotic triad (referent, meaning, and form) is used to represent composed semiotic symbols (see Figure 6.5). The prototype component is no longer required in composed semiotic symbols, that inherit their grounding from the lower-level symbols.

In this PhD Thesis, the reader can also find the terms *elementary* or *composed* words, *elementary* or *composed* meanings, and *elementary* or *composed* referents to respectively denote these constituent parts of *elementary* or *composed* semiotic symbols. However, only elementary prototypes are considered because, in this work, compositionality only occurs in the meaning space.

A clear distinction is made between prototypes and meanings, so that the former ones are preliminary iconic representations that originate the latter ones through the process of identification. Prototypes are similar and resemble the referents on which they are grounded. In contrast, meanings are arbitrary, in the sense that there is no correlation with the external referents being represented. That is, the identification is a bijection that transforms each of the stored prototypes onto a different meaning through an arbitrary mapping (see e.g. [13, 14]).

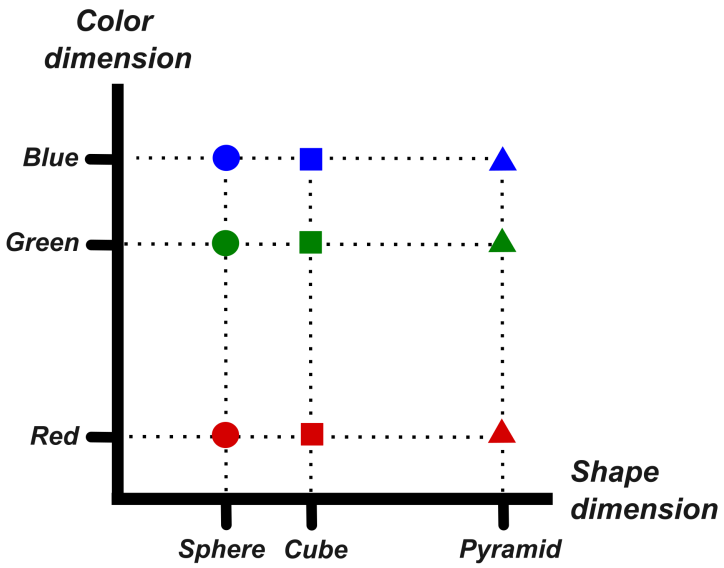


Figure 6.6: Example of a meaning space $\mathcal{M}$ with color and shape dimensions.

Each prototype has an associated meaning, and, when a new prototype is learnt, a new meaning is also created by sampling a random point in the meaning space $\mathcal{M}$. Notice that the *prototype-meaning* mapping is private to each robot and, therefore, it is not negotiated and shared using language games. An important property of the meaning space is that its dimension is fixed to the number of sensor channels (e.g. if the sensory apparatus is composed by color, shape, and quality channels then the dimension of $\mathcal{M}$ is three). All these dimensions or axes are orthogonal and form a basis of $\mathcal{M}$. This structural constrain is similar to the quality dimensions of the conceptual spaces in [197, 198], and they are included with the aim of enabling some features of compositionality of semiotic symbols. Additionally, prototypes are *genetically determined*, so that kin-related individuals would generally converge to highly comparable sets of protypes, because of their similar sensory apparatus. Conversely, even though there is still a genetic component, meanings are private and learnt by each agent through their experience and interaction with the environment.

A new meaning created by an agent is randomly generated, but can only be embedded inside the corresponding dimension of $\mathcal{M}$. This correspondence is dictated by the sensor that was used to perceive the referent on which the new meaning is grounded. That is, elementary meanings grounded on colors are constrained to the color axis, while elementary meanings grounded on shapes will always lie along shape dimension of $\mathcal{M}$. Likewise, a composed meaning grounded on both color and shape (e.g. red cubes) will always belong to the `color-shape` plane (see Figure 6.6).

Figure 6.7 provides a general diagram of the complete *semiosis* process (i.e. creation of semiotic symbols). Only the semiosis of elementary semiotic symbols is represented using as an example four different geometric objects (blue cube, red sphere, blue sphere, and red pyramid). Moreover, for the sake of clarity, all the objects are assumed to be of high quality. In the figure, one can observe four different sets (i.e. referents, prototypes, meanings, and lexicon) that are cyclically mapped through a series of biyective mappings. The union of all

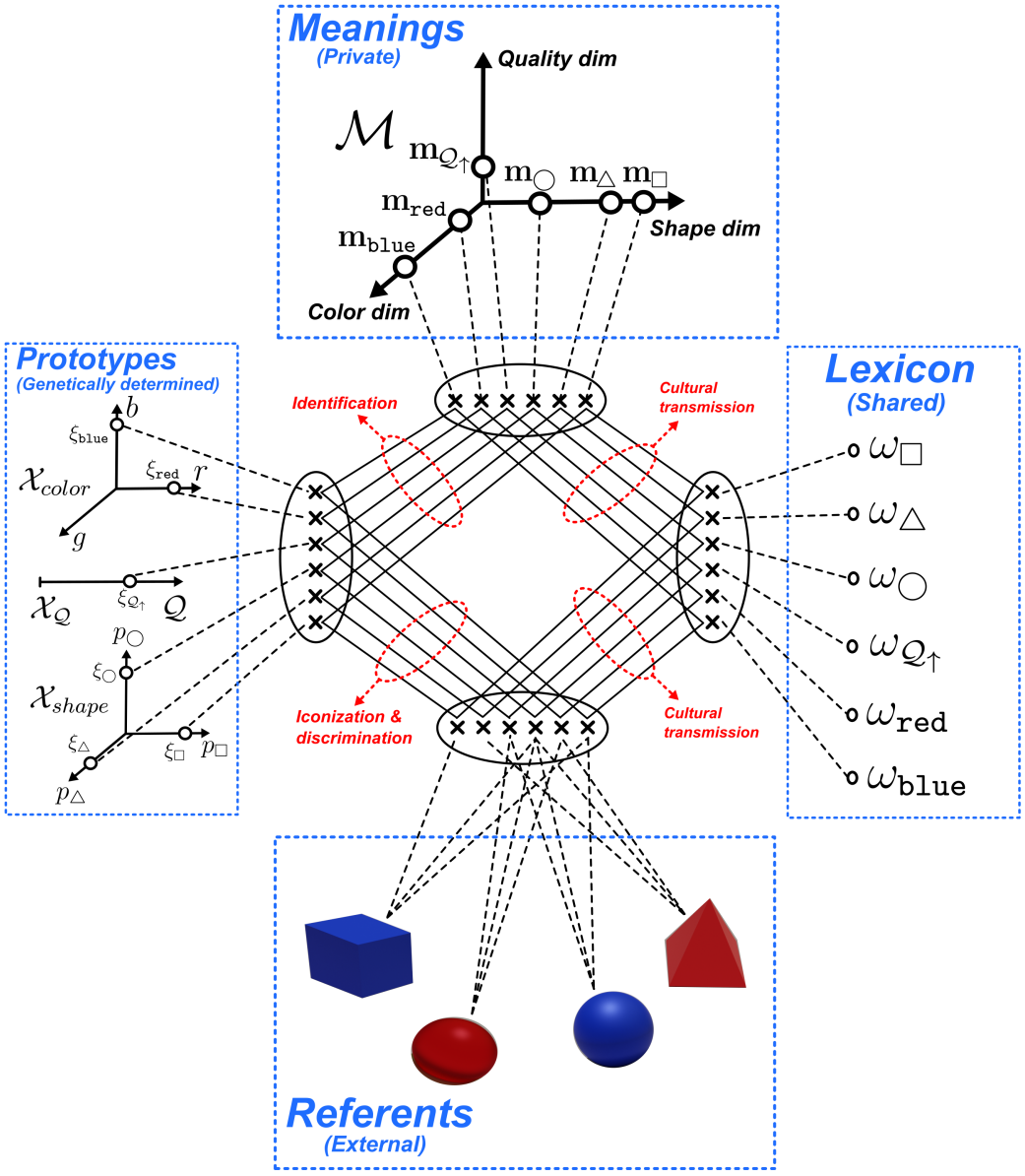


Figure 6.7: General diagram of the complete *semiosis* process of *elementary semiotic symbols*. It shows the different sets that constitute the tetrad of a elementary semiotic symbols (referents, meanings, forms, and prototypes), in blue, and the mapping between them, in red. The diagram shows an example with four different geometric objects: blue cube, red sphere, blue sphere and red pyramid. Furthermore, for the sake of clarity, all the objects are assumed to be of high quality.

these sets and mappings is what gives rise to semiotic symbols. The type of mapping from each set is highlighted in red, showing that, for instance, referents are mapped onto prototypes through the discrimination and iconization procedures (Section 6.2.3), whereas the mapping between prototypes and meanings is what Harnad defined as identification [13]. Additionally, the meaning-word and word-referent mappings strongly rely on a social component based on

cultural transmission (see Section 6.2.7).

6.2.5 Compositionality of the lexicon

Two complementary mechanics that promote the emergence of a compositional lexicon, namely, the meaning space and the Meaning Association Network (MAN), are implemented. The MAN is formally introduced as a directed graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$ that associates and connects meanings to create new semiotic symbols. Each node in the set $\mathcal{N}$ corresponds to a meaning, and edges in $\mathcal{E}$ are associations between nodes to give rise to new meanings. Input nodes correspond to elementary meanings that are directly grounded on objects in the real world, rather than resulting from composition of other meanings. In contrast, deeper layers in the MAN contain more abstract meanings produced from the cascading of multiple compositions. In this PhD Thesis, the constrain stating that meanings can only result from the combination of two other meanings is imposed, so that the in-degree of a node can only be zero (elementary meanings) or two (composed meanings). Besides, the creation of self-connections (composition a meaning with itself) and feed-back connections is avoided. Therefore, the pair of in-nodes or parent nodes, if any, define the meanings that are composed to create any semiotic symbol.

Figure 6.8 depicts an example of a fully developed MAN with six elementary meanings (in green), and six composed meanings (in orange and red). Among the composed meanings, four result from the combination of two previously acquired elementary meanings (level 1 composition), and two are created through the combination of one composed and one

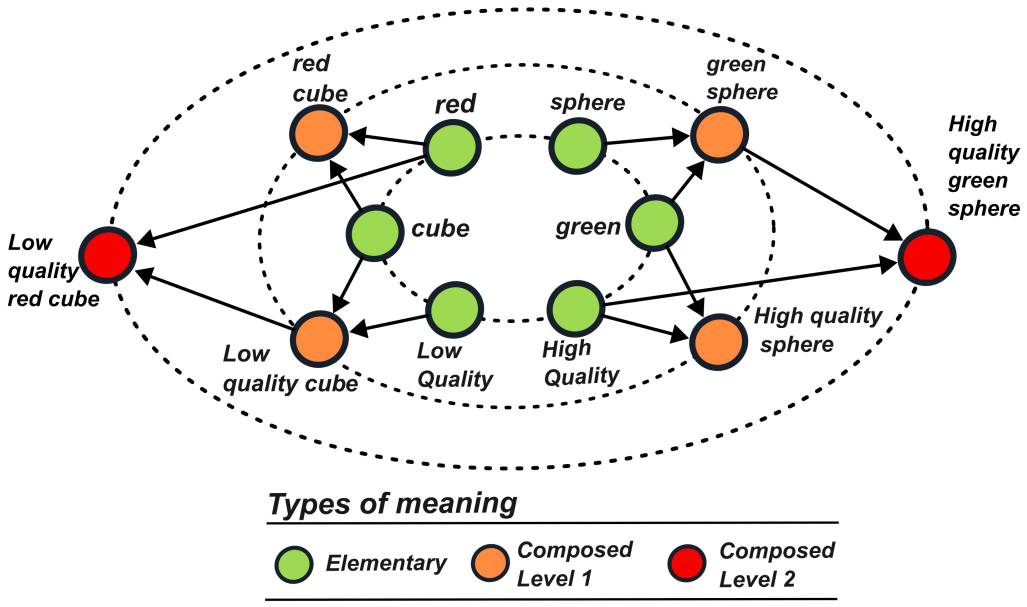


Figure 6.8: Meaning Association Network (MAN) in a hypothetical scenario. The depicted MAN is composed by six elementary symbols (in green) that are grounded on the colors red and green, the shapes cubes and spheres, and high and low qualities. These elementary symbols are combined to create six composed symbols, either resulting from a level 1 composition (in orange), or from a level 2 composition (in red).

elementary meanings (level 2 composition). The text attached to each node denotes the external referent on which the corresponding meaning is grounded. Additionally, the dashed ellipses represent the level of complexity and abstraction of the meanings lying on its perimeters.

The MAN is embedded in the meaning space $\mathcal{M}$, so that every node in $\mathcal{G}$ has an associated coordinate in $\mathcal{M}$. With this embedding in mind, the notation of the MAN can be extended to a spatial directed graph $\mathcal{G}(\mathcal{N}, \mathcal{E}, \mathcal{M})$. As it was previously mentioned, elementary meanings are positioned in $\mathcal{M}$ as random points in the corresponding axis or dimension. Composed meanings, rather than being explicitly random, are computed as the vector sum of the meanings of the in-nodes in $\mathcal{G}$. When some new meaning a is created from the composition of other meanings b and c, its meaning vector $\mathbf{m}_a$ is calculated as the sum of the meanings of its parents $\mathbf{m}_b + \mathbf{m}_c$. More formally, the condition in Equation 6.6 must be always fulfilled,

$$\mathbf{m}_a = \mathbf{m}_b + \mathbf{m}_c \iff (b, a), (c, a) \in \mathcal{E}, \tag{6.6}$$

where $a, b, c \in \mathcal{N}$ are nodes in $\mathcal{G}$, and $\mathbf{m}_a$, $\mathbf{m}_b$ and $\mathbf{m}_c$ are the meaning vectors associated to the nodes a, b and c, respectively. This property can be clearly observed in the example of Figure 6.9. The figure illustrates an example in which three elementary meanings, $\mathbf{m}_a$, $\mathbf{m}_b$ and $\mathbf{m}_c$, grounded on some arbitrary referents a, b, and c, are composed to give rise to complex meanings. Specifically, the figure shows how the new meanings are computed as the sum of the states of the two previous nodes in the MAN. It can be interpreted as the forward propagation of the directed graph $\mathcal{G}$, so that $\mathbf{m}_a$, $\mathbf{m}_b$ and $\mathbf{m}_c$ are the inputs of the system.

The main reason for embedding $\mathcal{G}$ in the meaning space $\mathcal{M}$ is that the agents can harness the topology of $\mathcal{M}$ to rapidly query certain properties of meanings without the need to back-propagate the MAN. For instance, imagine a certain complex meaning $\mathbf{m}'$ that results from the cascading of multiple compositions, so that $\mathbf{m}' = \mathbf{m}_{\texttt{red}} + (\mathbf{m}_{\square} + \mathbf{m}_{\mathcal{Q}_{high}})$. Then, an agent can directly extract the color component of the composed meaning by projecting

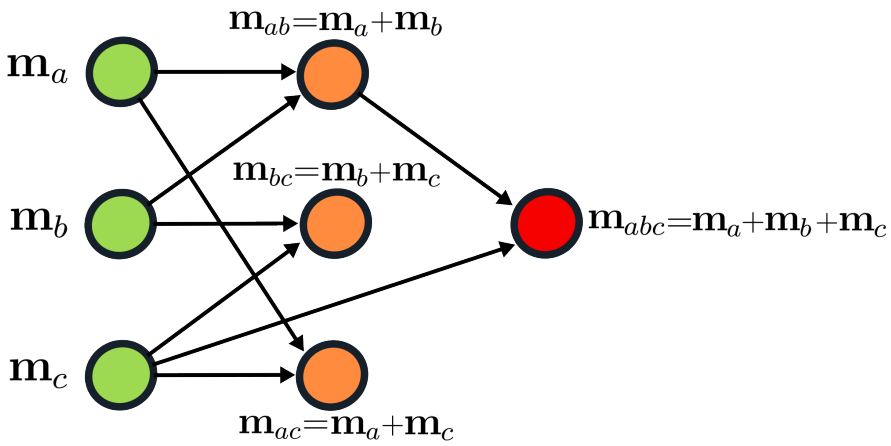


Figure 6.9: Example of the creation of composed meanings. The figure shows a MAN with three aribitrary elementary meanings ($\mathbf{m}_a$, $\mathbf{m}_b$ and $\mathbf{m}_c$), and all the possible complex meanings that can be derived from them. It also illustrates the process of calculating the meaning vectors as a forward propagation of $\mathcal{G}$.

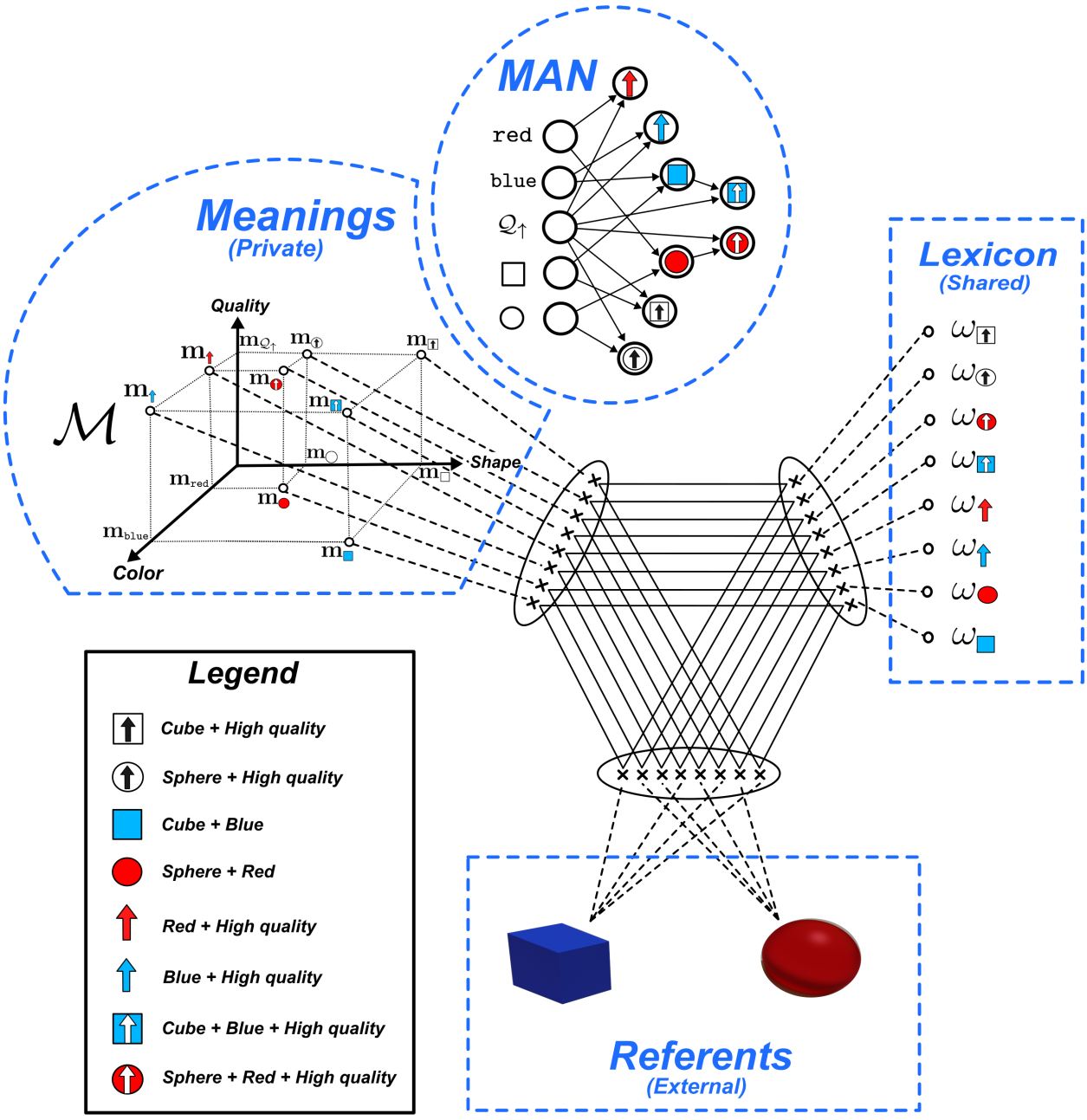


Figure 6.10: General diagram of the complete *semiosis* process of *composed semiotic symbols*. It shows the different sets that constitute the triad of a composed semiotic symbols (referents, meanings, and forms), in blue. The diagram shows an example with two different geometric objects: high quality blue cube, and high quality red sphere. Furthermore, the MAN is shown alongside the meanings, indicating how compositionality is defined.

$\mathbf{m}'$ onto the color dimension, whose outcome would be the meaning red $\mathbf{m}_{\texttt{red}}$. Similarly, by means of the operation $\mathbf{m}' - \mathbf{m}_{\square}$, a robot can verify whether the meaning $\mathbf{m}_{\square}$ grounded on cubic shapes is part of the holistic meaning $\mathbf{m}'$.

Figure 6.10 illustrates the general diagram of the complete *semiosis* process of composed

semiotic symbols. It shows the three interacting sets, namely, referents, meanings and forms, that gives rise to the *semiotic triad* of composed symbols, as opposed to the *semiotic tetrad* of the elementary ones depicted in Figure 6.7. For the sake of clarity, an example with two different geometric objects (high quality blue cube and high quality red sphere) is considered. The MAN graph is shown alongside the meaning space, so that the combination of both elements provides a complete representation of meaning compositionality.

6.2.6 Learning of compositionality

Robots acquire new meanings and expand their MAN via interaction with the environment. During the early stages of the robots' lifetime, only *elementary* meanings are created throughout direct grounding. After the acquisition of a solid basis of *elementary* meanings, agents start the learning of more complex *composed* meanings. This learning process is based on Hebbian learning [255], which is a biologically plausible learning and adaptive process of synaptic plasticity that operates under the premise of connecting more strongly those neurons that are fired together. In our scenario, instead of wiring neurons through synapses, the learning is produced through the association of unconnected nodes (meanings) in $\mathcal{G}$. New associations are created based on the co-occurrence of previously acquired meanings. As an example, lets imagine that an agent has previously learnt the elementary meanings $\mathbf{m}_{\texttt{red}}$ and $\mathbf{m}_{\square}$ grounded on the red color and the cubic shape. The agent further explores the environment and encounters red cubes, that evoke the activation of both $\mathbf{m}_{\texttt{red}}$ and $\mathbf{m}_{\square}$ at the same time, that are not yet associated. The Hebbian learning algorithm will use such co-ocurrence to create a new composed meaning $\mathbf{m}_{\texttt{red}+\square} = \mathbf{m}_{\texttt{red}} + \mathbf{m}_{\square}$ in $\mathcal{M}$ and a new association in $\mathcal{G}$.

Each meaning $\mathbf{m}_i$ has an associated trace β_i, that acts as a short-term memory indicating how much time has elapsed since the referent on which the meaning is grounded was last seen. Specifically, traces are subject to the dynamics in Equation 6.7,

$$\beta_i(t+\partial t) = \begin{cases} \beta_i(t) - \lambda_\beta \partial t \beta_i(t), & \text{if not } \mathbf{m}_i \textit{ activated} \\ 1, & \text{if } \mathbf{m}_i \textit{ activated} \end{cases} \tag{6.7}$$

where λ_β is some fixed forgetting rate. A meaning $\mathbf{m}_i$ is *activated* only if one of the following conditions is met:

(1) it is an *elementary* meaning, and the robot is currently observing the external referent on which it is grounded (e.g. the meaning $\mathbf{m}_{\texttt{red}}$ is *activated* when the agent observes red objects).

(2) it is a *composed* meaning, and both of its constituent parent meanings are *activated* (e.g. the meaning $\mathbf{m}_{\texttt{red}+\square}$ is *activated* when the agent observes a cube that is red).

In addition to β_i, some learning weights α_{ij} are defined between every pair of meanings $\mathbf{m}_i$ and $\mathbf{m}_j$. These weights represent the level of association between two meanings, and they are used to determine whether there should exist an association between them or not. Equation 6.8

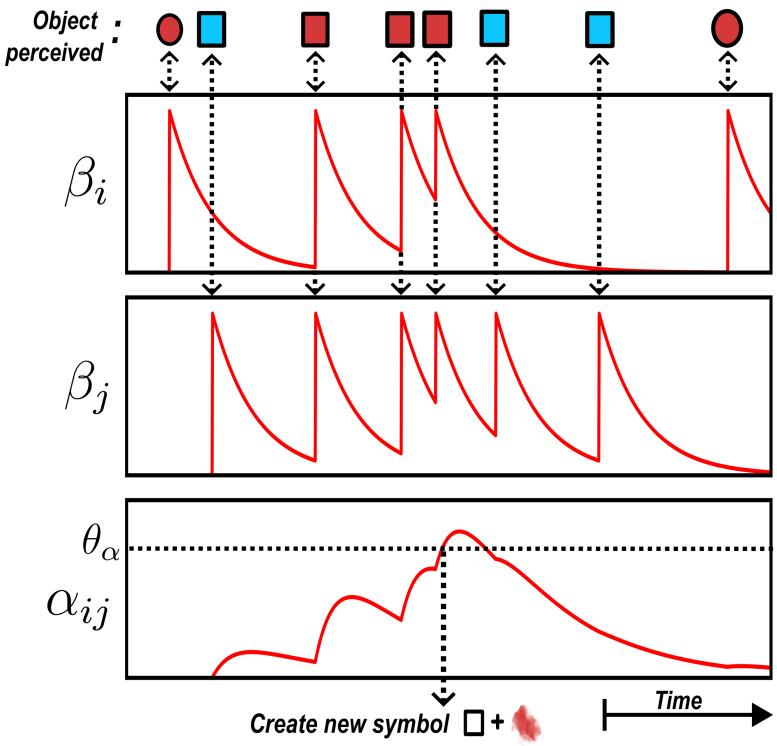


Figure 6.11: Example of the creation of a new association between meanings grounded on red color and cubic shape. The figure shows the temporal evolution of β_i and β_j that are *activated* and set to a value of 1 when the robot observes the red and cubic objects, respectively. It also plots the evolution of α_{ij} that results in the creation of the new association when its value surpassess a threshold θ_α. In all the graphics, the horizontal axis is the simulation time. The uppermost part of the figure shows the objects that the robot perceives at each time instant of the example.

describes the dynamics of the weights α_{ij},

$$\frac{\partial \alpha_{ij}(t)}{\partial t} = \lambda_\alpha \beta_i(t)\beta_j(t) - \frac{1}{\tau_\alpha}\alpha_{ij}(t), \tag{6.8}$$

where λ_α can be viewed as a learning rate, whereas τ_α is the decay time constant of the long-term memory. The weight dynamics in Equation 6.8 are based on the unsupervised Hebbian learning of the synapses' weights of artificial neural networks. Using this equation, a new association between two existing meanings $\mathbf{m}_i$ and $\mathbf{m}_j$ is created in the case that the value of α_{ij} exceeds some predefined threshold θ_α (hyper-parameter established by the researcher). The values of λ_α and τ_α determine the ease of the creation of new associations. Before the creation, it is verified that the new meaning does not already exits in the semiotic symbol set. Additionally, redundancy with existing meanings is also checked (e.g. to avoid having meanings grounded on red cubes that have low quality and low quality cubes that are red at the same time). When a new composed meaning resulting from $\mathbf{m}_i$ and $\mathbf{m}_j$ is finally created, its value is fixed to $\mathbf{m}_{ij} = \mathbf{m}_i + \mathbf{m}_j$ and its form is created randomly as a three syllable word. Finally its, referent is inherited from the referents of its parent meanings.

Formally, the creation of a new composed meaning is defined as in Equation 6.9

$$\alpha_{ij} > \theta_\alpha \text{ and } (i,j) \neq \mathcal{E} \implies \text{Create new association} \begin{cases} \nearrow & \mathbf{m}_{ij} = \mathbf{m}_i + \mathbf{m}_j \\ \searrow & \text{Add edge } (i,j) \text{ to } \mathcal{E} \end{cases} \tag{6.9}$$

Figure 6.11 shows an example of the creation of a new association between the meanings grounded on red color and cubic shape. It plots the temporal evolution of some β_i corresponding to the red color, β_j related to the cubic shape meaning, and α_{ij} representing their level of association. The values of β_i and β_j are set to 1 when the robot observes red, and a cubic objects, respectively (the objects being perceived by the agent at each time are shown in uppermost part of the figure). Thereafter, β_i and β_j decay exponentially until the referents are observed again. The spikes in β_i and β_j contribute to the growth of α_{ij}, which leads to the creation of the new association when its value surpasses the threshold θ_α.

6.2.7 Language games and cultural transmission

A custom variation of a naming game is used as the cultural transmission mechanism in which the robots participate to converge to a common lexicon. Such language game is defined by the following rules:

- Robots are initialized in the world with an empty lexicon. New words that are acquired through social interactions are coupled to existing or new semiotic symbols. That is, they are properly associated to meanings and grounded (directly or inherited) on external objects.
- Periodically and with some predefined probability, each agent acts as the *speaker* by broadcasting a random word in its lexicon. The transmitted message is received by all the robots on its local surroundings (closer than 1 meter).
- If the chosen word is *elementary* (i.e. associated with an *elementary* meaning), then it broadcasts the tuple (ξ_i, ω_i), where ξ_i is the prototype associated to the word ω_i (Figure 6.12). Such transmission of prototypes is used in this PhD Thesis as the joint attention mechanism, that allows hearers to acknowledge the topic or referent of the communication.
- Alternatively, in case of selecting a *composed* word (i.e. associated to a *composed* meaning) to be broadcasted, the robot would transmit the tuple $(\omega_{p1}, \omega_{p2}, \omega_i)$, where ω_{p1} and ω_{p2} are the parent words that are composed to produce ω_i (see also Figure 6.12). In this scenario, the *speaker* acts as a teacher, showing the *hearer* the new word through its definition and decomposition.
- When a *hearer* receives a new word (either elementary or composed), it updates its lexicon. This operation diverges depending on whether or not the word already exists in its own lexicon. If the word is unknown, then the *hearer* adds it to its lexicon and creates a new semiotic symbol:

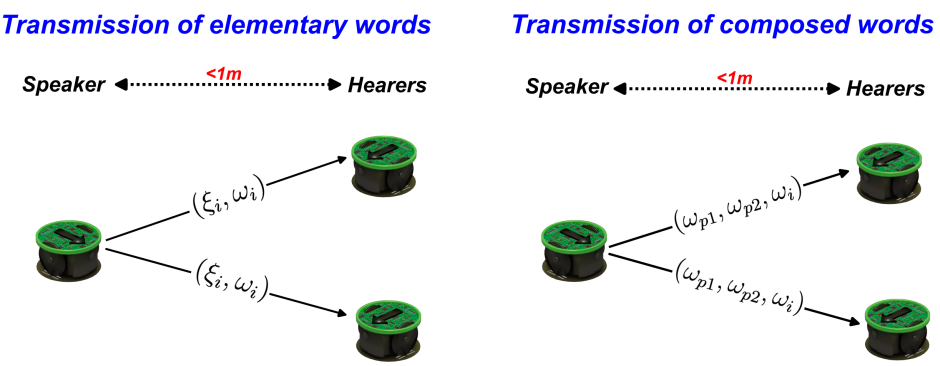


Figure 6.12: Overview of the customized naming game used in the experiment. Unidirectional interactions among a single speaker and multiple hearers during language games. It shows two possible types of transmissions with elementary and composed words.

(a) Unknown elementary words: the semiotic symbol is created using the received prototype ξ_i, creating a new associated meaning in $\mathcal{M}$, and adding a new input (unconnected) node to the MAN $\mathcal{G}$.

(b) Unknown composed words: the new word ω_i is only added if the lexicon already contains both ω_{p1} and ω_{p2}, that are composed to create ω_i. A new composed meaning $\mathbf{m}_i$ is created as the vector sum $\mathbf{m}_{p1} + \mathbf{m}_{p2}$, and through the creation of the new node and edges in the MAN. Very importantly, the meaning vectors $\mathbf{m}_{p1}$ and $\mathbf{m}_{p2}$, that are used to create $\mathbf{m}_i$, are the ones internally developed by the *hearer*. In fact, owing to the fact that meanings are private, the values of $\mathbf{m}_i$, $\mathbf{m}_{p1}$, and $\mathbf{m}_{p2}$ of *speaker* and *hearer* are different and share no correlation between them.

Alternatively, if the *hearer* does have the new word already stored in its lexicon, the semiotic symbol corresponding to the stored word is properly updated:

(a) Known elementary words: the robot changes the form of the semiotic symbol whose prototype is closest to the received ξ_i (joint attention). Because of its *private* nature, the corresponding meaning is not altered.

(b) Known composed words: the lexicon, meaning space, and MAN are updated with the new information.

- Each language game is played by a single *speaker* and multiple *hearers*. The number of hearers depends on the number of neighbors of the *speaker*.
- The conversations during the languages games are unidirectional, so that, upon reception of a message, the *hearer* is the only agent that updates its lexicon. This modification with respect to most naming games in the literature is required in SR because of limiting factors such as the large number of agents and the existence of multiple *hearers* [53, 213].

6.3 Results

6.3.1 Assessment metrics

The Lexicon Sharing Score (LSS) is defined as the assessment metric that indicates the degree of lexicon overlapping and sharing along the robot swarm. The LSS metric is defined in Equation 6.10 as,

$$\text{LSS} = \frac{2}{R \cdot (R-1)} \sum_{i=1}^{R} \sum_{i<j \leq R} \mathcal{J}(\mathcal{L}_i, \mathcal{L}_j) \tag{6.10}$$

where R is the number of robots in the swarm, $\mathcal{L}_i$ and $\mathcal{L}_j$ are the lexicons of the i-th and j-th agents. $\mathcal{J}(\mathcal{L}_i, \mathcal{L}_j)$ is the Jaccard similarity between the two sets [256], as showed in Equation 6.11,

$$\mathcal{J}(\mathcal{L}_1, \mathcal{L}_2) = \frac{|\mathcal{L}_1 \cap \mathcal{L}_2|}{|\mathcal{L}_1 \cup \mathcal{L}_2|} = \frac{\upsilon_{ij}}{L_1 + L_2 - \upsilon_{ij}}, \tag{6.11}$$

where $|\cdot|$ is the cardinal of a set, $\upsilon_{ij} = |\mathcal{L}_i \cap \mathcal{L}_j|$, and L_i and L_j are the sizes of the lexicons $\mathcal{L}_i$ and $\mathcal{L}_j$. An optimal value of LSS = 1 is obtained when all the agents share exactly the same words in their lexicons.

LSS does not reveal information about the size of the lexicon, therefore, this metric is combined with the total number of words in all the lexicons, υ (Equation 6.12).

$$\upsilon = \left| \bigcup_{i=1}^{R} \mathcal{L}_i \right|. \tag{6.12}$$

Furthermore, υ_{elem} and υ_{comp} are used to refer to the total lexicon size considering only elementary and composed words, respectively (such that $\upsilon = \upsilon_{elem} + \upsilon_{comp}$).

Additionally, the Compositionality Sharing Score (CSS) is defined as a metric to assess the degree of consensus along the robot swarm in terms of compositionality in their lexicons. Before the formal definition of the CSS, the set of associations $\mathcal{A}_i$ of the i-th agent in the swarm is introduced. An association belonging to some $\mathcal{A}$ is defined as the tuple $(\omega_{p1}, \omega_{p2}, \omega)$, where ω is some word in a lexicon and ω_{p1} and ω_{p2} are the parent words that give rise to ω. Thus, let $a_{ij} = |\mathcal{A}_i \cap \mathcal{A}_j|$ be the number of associations that two agents share, so that an association is shared provided that the following conditions are met:

(1) $\omega, \omega_{p1}, \omega_{p2} \in \mathcal{L}_i$,

(2) $\omega, \omega_{p1}, \omega_{p2} \in \mathcal{L}_j$, and

(3) ω results from the combination of ω_{p1} and ω_{p2} for both agents.

With these definitions in mind, the CSS is formulated in Equation 6.13,

$$\text{CSS} = \frac{2}{R \cdot (R-1)} \sum_{i=1}^{R} \sum_{i<j \leq R} \mathcal{J}(\mathcal{A}_i, \mathcal{A}_j) = \frac{2}{R \cdot (R-1)} \sum_{i=1}^{R} \sum_{i<j \leq R} \frac{a_{ij}}{A_i + A_j - a_{ij}}, \tag{6.13}$$

where A_i and A_j are the number of elements in the sets $\mathcal{A}_i$ and $\mathcal{A}_j$.

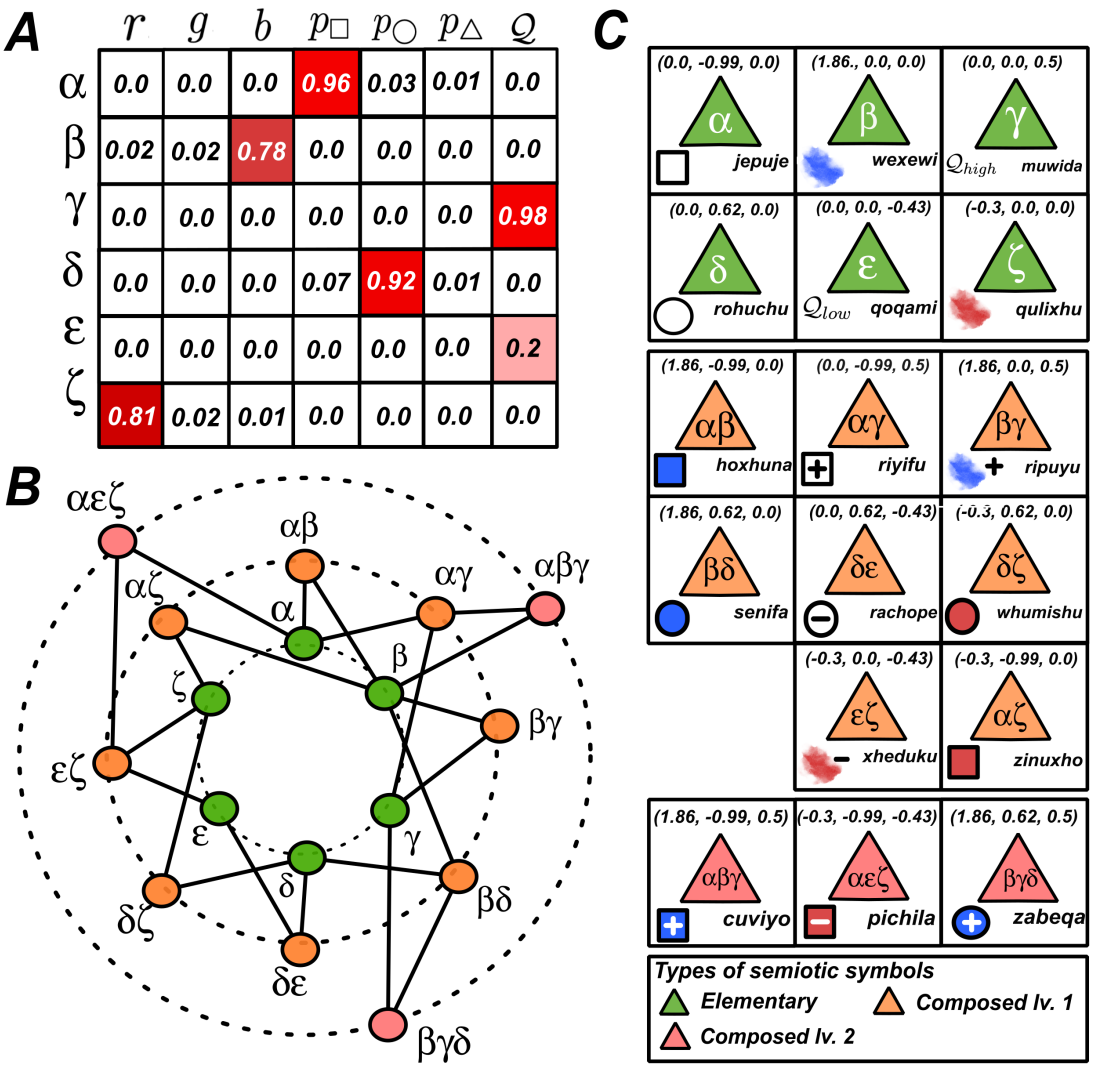


Figure 6.13: Results for 3 referents and 20 robots (focus on single robot). (**A**) Prototypes learnt by one of the robots in the swarm. (**B**) Meaning association network developed by the same robot, showing how symbols interconnect to give rise to incrementally more complex meanings. (**C**) Set of semiotic symbols of the agent, showing the referent (bottom left), the meaning (top), and the form (bottom right). In (**B**) and (**C**), the semiotic symbols are distinguished by color: the elementary symbols in green, and the composed symbols either in orange and red for combinations of two or three symbols, respectively.

6.3.2 Results with three referents

Firstly, the results of the experiment are assessed using three different types of objects: low quality red cubes, high quality blue spheres, and high quality blue cubes. These three composed referents ideally give rise to six elementary words, $\upsilon_{elem} = 6$ (blue color, red color, cubic shape, spherical shape, high quality, and low quality), and $\upsilon_{comp} = 11$ composed words resulting from the referents: blue cube, red cube, blue sphere, high quality blue color, low quality red color, low quality cube, high quality cube, high quality sphere, low quality red

cube, high quality blue sphere, and high quality blue cube. Consequently, the ideal size of the total lexicon is 17 ($\upsilon = \upsilon_{elem} + \upsilon_{comp}$). Four objects of each kind were randomly scattered in the arena at the beginning of each simulation. In this first experiment, the robot swarm is composed by 20 robots that are randomly positioned in the arena.

Figure 6.13 shows the results of the experiment with these three external objects, focusing on a single agent at the end of a simulation. Figure 6.13A illustrates the six prototypes learnt by the depicted agent, corresponding to the referents: cubic shape, blue color, high quality object, spherical shape, low quality object, and red color. Figure 6.13C depicts all the semiotic symbols developed by the robot, so that:

- the *referent* is sketched in the lower-right side of the square,
- the *meaning* is shown in the upper side as a three-dimensional vector,
- the *form* is shown as the three syllable word in the lower-left side,
- the color of the semiotic symbol denotes whether it is elementary, or composed,
- a tag inside semiotic symbol (e.g. α) is used for clarity to refer to each semiotic symbol.

For instance, the semiotic symbol tagged as $\alpha\beta$ corresponds to the external object `blue+cube`, the word `hoxhuna`, and the meaning $\mathbf{m}_{\alpha\beta} = (1.86, -0.99, 0.0)$. Moreover, notice that the meaning $\mathbf{m}_{\alpha\beta}$ is the vector sum of $\mathbf{m}_{\alpha}$ and $\mathbf{m}_{\beta}$. The prototypes (see the rows in Figure 6.13A) are not shown alongside the corresponding elementary semiotic symbol to ease the readability of the figure. However, the reader can easily associate prototypes and elemtary semiotic symbols by means of the tags α-ζ.

Additionally, Figure 6.13B shows the learnt MAN, showing how the semiotic symbols in Figure 6.13C are associated to create new concepts. The six prototypes of Figure 6.13A give rise to six elementary semiotic symbols (tagged as α-ζ and colored in green). Moreover, the elementary symbols are combined and associated to create eleven composed semiotic symbols according to the MAN in Figure 6.13B: eight resulting from the composition of two elementary semiotic symbols (tagged as $\alpha\beta$-$\alpha\zeta$ and colored in orange), and three from the composition of one elementary and one composed semiotic symbol (tagged as $\alpha\beta\gamma$, $\alpha\epsilon\zeta$ and $\beta\gamma\delta$, and colored in red).

Figure 6.14 illustrates the results considering the whole robot swarm, instead of focussing on a single agent. Specifically, Figure 6.14A plots the temporal evolution of the LSS metric across the simulations, whereas Figure 6.14B depicts the evolution of the CSS. Upon convergence, both metrics reach nearly optimal values, suggesting that the emergence of a common lexicon is correctly reached. The steady state results in Figure 6.14B indicate that compositionality is also achieved and shared across swarms. Moreover, notice that the transient of the CSS metric is unstable at the early time instants of simulations. The main reason for this phenomenon is that agents first negotiate on elementary words, that are non-compositional. Compositionality only emerges when a solid basis of the elementary symbols is learnt.

Finally, Figure 6.14C shows the time evolution of υ, υ_{elem}, and υ_{comp} in red, green, and blue, respectively. The steady state value of υ correctly reaches, on average, the value of 17 shared words, which is the correct number of possible words resulting from the three objects of the

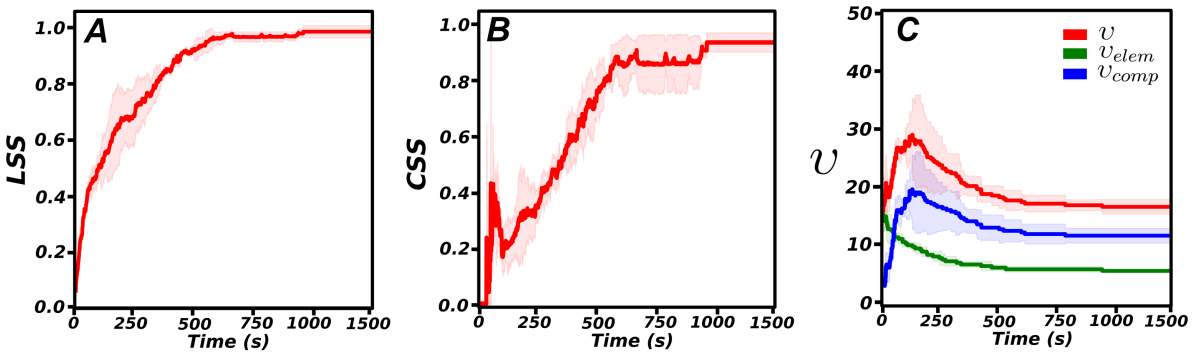


Figure 6.14: Results for 3 referents and 20 robots (focus on the whole robot swarm). (**A**) Time evolution of the LSS metric along the duration of a simulation. (**B**) Time evolution of the CSS metric along the duration of a simulation. (**C**) Time evolution of the number of elementary (υ_{elem}), composed (υ_{comp}) and total (υ) words shared by all the members of the swarm. In (**A** to **C**) 50 independent simulations are used to represent statistically significant results, showing the sample mean of the corresponding metric in darker colors and the standard deviations as the clear shadows.

experiment. Additionally, the steady state value of υ_{elem} reaches, on average, the value of 6, which is the exact number of elementary referents observable by the robots in the environment (which are `red`, `blue`, `cube`, `sphere`, `high quality`, and `low quality`). The average value of $\upsilon_{comp} = 11$ is also correctly reached.

6.3.3 Results with five referents

In this section, the lexicon production, sharing and compositionality are assessed with five composed referents, and robot swarms ranging from 10 to 100 individuals. The geometric objects scattered in the arena are low quality red cubes, high quality green cubes, high quality green spheres, low quality red spheres, and low quality red pyramids. Following a similar decomposition as in the results with three objects, these five composed referents ideally give rise to $\upsilon_{elem} = 7$, $\upsilon_{comp} = 19$, and $\upsilon = 26$.

Figure 6.15 collects the results of this experiment using robot swarms of 10, 20, 50, 75, and 100 individuals. Figure 6.15A depicts the sensory space $\mathcal{X}$ of dimension seven, that contains all the prototypes, visualized as a two-dimensional space using t-distributed Stochastic Neighbor Embedding (t-SNE) algorithm [257]. It plots the distribution of the learnt prototypes using a heatmap as the representation tool, and considering the sets of prototypes of all the robots in swarms of 50 robots along 50 independent trials (2500 data points in total). The aim of this figure is to show that the discrimination process is able to correctly and ubiquitously capture invariant properties of distal objects, regardless of the high randomness and variability in the way agents perceive the objects. The acquisition of such solid prototypes can be observed in Figure 6.15A in the number of data clusters, that correctly matches the number of different elementary referents (red, green, cube, sphere, pyramid, high quality, and low quality).

Figures 6.15B and C show the time evolution of the LSS and the CSS metrics. These metrics are illustrated for swarm sizes ranging from 10 to 100, shown in different colors, to assess the

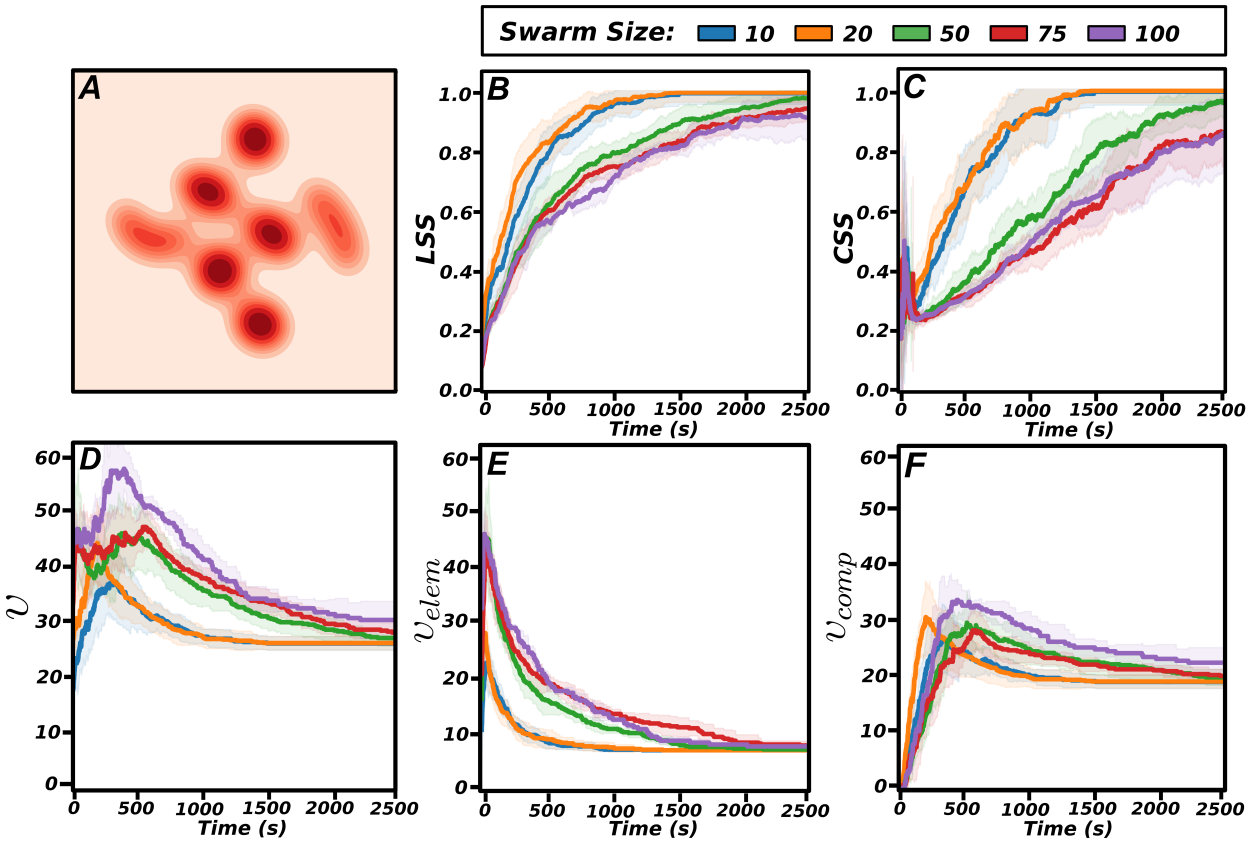


Figure 6.15: Results for 5 referents and swarm sizes ranging from 10 to 100 robots. (**A**) Sensory space $\mathcal{X}$ visualized in two dimensions using t-SNE algorithm. It depicts the heatmap of the distribution of prototypes learnt by all the robots in the swarms of 50 robots and using 50 independent simulations (2500 data points in total). (**B** and **C**) Time evolution of the LSS (**B**) and CSS (**C**) metrics along simulations of 2500 second duration. (**D** to **F**) Time evolution of υ (**D**), υ_{elem} (**E**), and υ_{comp} (**F**), that assess the number of words (total, elementary and composed) that are shared by all the members of the robot swarm. In (**B** to **F**), the results are shown for swarm sizes of 10, 20, 50, 75, and 100 robots, and 50 independent simulations with random initialization are collected and used to create the estimate of each metric. Specifically, each of the darker curves are the sample mean estimates of the corresponding metric, whereas the light contours are bounded by the standard deviation of the metric estimate.

scalability of the system. For each swarm size, 50 independent simulations are used to compute the sample mean estimates (darker curves) and standard deviations (light contours). Focusing on the steady state values, robots swarms with sizes of 10 and 20 achieve nearly optimal LSS and CSS results in about 1500 seconds of a simulation (corresponding to 30000 simulation cycles with a sampling period of 0.05 seconds). It implies a correct convergence to a common lexicon in most of the trials. Moreover, CSS values near 1.0 indicate that compositionality in the lexicon is also reached and spread across swarms. Increasing the number of robots results in a slight decrease of the steady state values of LSS and CSS. However, this degradation is notably low, with average LSS of 0.99, 0.95, and 0.92 for swarms of 50, 75, and 100 robots. The decrease of CSS is also minimal, being 0.97 for 50 robots and 0.86 for swarm sizes of 75 and 100. Regarding the transient periods, there is also a degradation of the convergence

time as the number of robots scale, being around 1500 seconds for swarms of 50 robots and between 2000 and 2500 seconds for swarm sizes of 75 and 100. The growth of the time elapsed to reach convergence is highly reasonable, compared to the 1500 seconds of lower swarm sizes, and, to some extent, unavoidable due to the large number of individuals and language games required to reach a common lexicon.

Figures 6.15D to F depict the time evolution of υ, υ_{elem}, and υ_{comp}, used to assess the sizes of the shared compositional lexicons. The optimal value of υ (corresponding to the number of possible words in the experiment) is 26, which is exactly reached on average by swarms of 10 and 20 robots (Figure 6.15D). In the case of 50, 75 and 100 individuals, the degradation with respect to the optimal value is, on average, of one, two, and four words, respectively. Figures 6.15E and F decompose the total size of the shared lexicon (υ) into total elementary words (υ_{elem}), and total composed words (υ_{comp}). The optimal values of υ_{elem} and υ_{comp} are 7 and 19, because of the number of elementary referents and possible compositions among them in this experiment. It can be observed that most of the errors and differences in the lexicons come from divergences in the composed words (Figure 6.15F). Steady state values of υ_{elem} are optimal, on average, for swarm sizes of 10 and 20 robots, and nearly optimal (with one or two different words) for swarms of 50, 75, and 100 agents. Finally, some interesting phenomena can be observed in Figures 6.15E and F regarding the transient period and the early stages of the simulations. During the initial seconds, the robot swarms generate a huge amount of elementary words (Figure 6.15E), because agents assign random words to new semiotic symbols learnt from their early experience. After some maximum value of υ_{elem} is reached, the great disparity of words is rapidly diminished because of the inter-robot interactions and engagement in language games. On the contrary, this scenario is completely different in the case of the transient period of υ_{comp} (Figure 6.15F), where the initial growth of the lexicon is notably slower, and the overshot is greatly reduced. This different behavior is caused because the creation of composed words requires a solid basis of elementary semiotic symbols to be produced.

6.4 Discussion

Addressing the SGP in SR with semiotics

The self-organization of a shared lexicon in societies of cognitive agents, tightly related to the Symbol Grounding Problem (SGP) [13, 14, 206], has been deeply addressed using semiotics [14, 221, 258, 187, 211]. The spotlight of most of these studies is to specifically address the SGP and to explore the origins, nature, and evolution of language and communication using Artificial Life. Here, an engineering perspective is considered, borrowing some of the concepts of semiotics and language games to apply them to the field of SR. As proposed by Cambier et al. [11], language games can be key element to overcome the stagnant state of emerged communication in SR systems (which generally results in signaling-based communications designed ad hoc for specific tasks), and pave the way for more complex language-like communications. Some studies have already applied minimal naming games to SR [240, 241, 53, 213]. However, these studies only tackle the problem considering the *social* substrate. In this PhD Thesis, the problem was attacked both from the *social* and

physical perspectives, creating perceptually grounded *semiotic symbols*, distinguishing between *meanings* and *forms*, and considering the main stages (iconization, discrimination, and identification) proposed by Harnad to solve the SGP [13, 14]. It was assessed that semiotics and language games can be successfully applied to SR, and that robot swarms can self-organize a common lexicon of an optimal size (considering the number of referents). Specifically, it was shown that an optimal lexicon size of 17 words is reached, on average, in swarms of 20 robots. Moreover, as a key property of SR systems, it was demonstrated that the system scales properly up to swarms of 100 robots. The results reveal a slight degradation of the mean value of the LSS score of 0.03 in swarms of 50 robots and of 0.14 in swarm sizes of 100, with respect to the optimal results reached on average in the smallest robot swarms.

Embodiment, exploration, and limited communication

A key feature to solve the SGP is that agents should be embodied and situated in the environment on which meanings and forms are grounded. In other words, the semiotic symbols should be grounded based on the individual experience and interaction with the external world. In SR, this feature is clearly compulsory, as robots are necessarily embodied and explore their local environment without the need of a central control. In the experiments of this chapter, mobile robots navigate and explore the arena in order to build their own and private set of meanings. However, meanings can only be grounded on objects detected by the robot's camera in its local surroundings. And this grounding is uniquely and subjectively built based on their experience and situation with the respect to the target object. The prototypes and the meaning spaces of each robot are, therefore, unique and diverge from the ones created by any other robot in the swarm.

A non-trivial problem arises when robots have to converge to the exact same set of words provided that the meanings of those words are private and internal. This problem becomes more complex when the number of agents increases drastically (as in SR), because the amount of different private meanings and subjective groundings grow likewise. It was proved that it is possible to address this problem using language games and semiotics, showing convergence to lexicons of 26 different words in swarms of up to 100 robots.

Emergence of compositionality

It was demonstrated that compositionality can emerge and be perceptually grounded in swarm robotics systems. My proposal of compositionality and its acquisition from zero is based on three main elements: the MAN, the meaning space, and the hebbian learning. As far as I am concerned, compositionality in a lexicon has not been addressed before in the SR field and using large swarms.

In line with the grounding of the semiotic symbols, robots create and define compositions between learnt meanings based on their own experience. Firstly, robots build a solid basis of elementary semiotic symbols, that are directly grounded on referents (e.g. red, cube, or low quality). Subsequently, these symbols are incrementally combined to create increasingly more complex composed meanings, that inherit the grounding of the basic ones. This is a highly relevant step to properly create symbolic representations [13]. A new meaning is created as

the composition of two prior symbols provided that the referents used in the composition are perceived at the same time by the robot. Furthermore, the new meaning is only created when co-occurrence of referents is repeatedly produced across time. More specifically, Hebbian learning is proposed as the building block that robots internally use to create compositionality. Alternatively, compositionality can be also produced through social interactions and language games, in which a robot shares a new word and the "parent" words that were used to compose it. It resembles the teaching of a new word by defining it (e.g. an individual teaches the word zebra by describing it as a horse with stripes [13]). Very importantly, this teaching process does not fall into the Searle's Chinese room argument [259] because an agent only learns a new word provided that the constituent parts of its definition are already known and properly grounded (i.e. the new word zebra can only be learnt once the words horse and stripes have been already assimilated and grounded).

Towards language-like communications

Communication systems that emerge in most SR studies fall under the category of signaling communication, being typically far removed from language-like communications. Moreover, such stagnation in the evolution of communication resembles some aspects of animal communication. More precisely, even though some animal communication systems exhibit some specific properties of human language in isolation, all of them are considered as signaling systems (with a wide spectrum of complexity in their mechanics, semantics and evolutionary history). In Chapter 3.2, it was reviewed that some of the most relevant and representative characteristics of language are symbolic communication and compositionality, duality of patterning, arbitrariness, displaced communication and cultural transmission. In this chapter, the creation of a shared lexicon, which is one of the fundamentals of language-like communication, was addressed. The emerged lexicon fulfills, at some extent, all of the language properties enumerated above:

(1) *Cultural transmission*: in the experiments of this chapter, the robots negotiate and converge to a common lexicon through a cultural transmission process based on language games.

(2) *Duality of patterning*: the words in the robots' lexicons are created as the combination of three randomly chosen syllables. Such syllables lack of meaning and physical grounding by their own. It is their combination what creates a word, which has an associated meaning grouded on external geometric objects.

(3) *Arbitrariness*: the process of identification, proposed by Harnad in [13] as one of the fundamental steps to solve the SGP, is considered in this chapter as the mechanism that maps prototypes and meanings. On the one hand, prototypes are iconic representations that belong to the same vector space as the input signals measured by the sensors. On the other hand, meanings are symbolic representations that have no direct relation with the associated prototype or referent on which they are grounded. Therefore, the meaning component of the semiotic symbols fulfills the *arbitrariness* property. Similarly, words are also arbitrary, because they are created randomly and share no relation with meanings and referents.

(4) *Symbolic communication and compositionality*: the obtained lexicon clearly exhibits compositionality in terms of word compositions to create new terms. Moreover, such compositionality is acquired from zero through lifetime learning and cultural transmission. However, even though it is a noticeable milestone towards language-like communications, the level of compositionality tackled in this chapter only considers adjective-noun syntactic relations. Therefore, there are still many (and more complex) aspects of compositionality and grammar to be explored, opening up an important line of future research.

(5) *Displaced communication*: the type of communication presented in this chapter exhibits some characteristics of displaced communication. Robots engaged in language games learn and negotiate new vocabulary whose grounding referent is potentially not present neither in space nor in time. For example, two robots may agree upon the word used to refer to the shape cube when there is no cubic object nearby.

Even though these properties are displayed, only the acquisition, negotiation, and production of the compositional lexicon is addressed in this PhD Thesis. In order to truly approach to a language-like comunication, robots must use the acquired vocabulary and semiotic symbols to communicate relevant information during the execution of some task. Moreover, more complex aspects of grammar, such as new syntactic categories, predication, or deixis, among many others, have to be addressed.

6.5 Summary and conclusions

In this chapter, a robot swarm that, without any prior knowledge, can self-organize and converge to a common compositional lexicon to name different geometrical objects scattered along the environment was proposed. Moreover, the perceptually grounded symbol grounding problem in embodied agents is considered. During their lifetime, the robots develop *semiotic symbols*, which are tetyrads composed by a referent (external object), a meaning (inner representation), a form (word in lexicon), and a prototype. *Semiotics* is used for the creation of private meanings, while cultural evolution, based on language games, tackles the spreading of the lexicon along the robot swarm. Furthermore, it was shown that compositionality (the meaning of a word is created based on the meaning of its constituent parts) can emerge in perceptually grounded lexicons shared in robot swarms. The performance and scalability of the proposed system were assessed with swarms up to 100 robots, revealing that convergence to common compositional lexicon of 26 different words can be achieved with high accuracy.

Several future lines of research can be considered from the work of this chapter. Firstly, it can be extended to real robot swarms and non-simulated environments. Secondly, other forms of compositionality, such as grammar rules or syntactical relations, can be considered. Finally, in this chapter, words are perceptually grounded on physical objects and concepts in the environment. An interesting extension of this process would be to ground symbols on actions and behaviors of the robots. A semiotic symbol could be grounded, for instance, on the behaviors "*Explore the environment*" or "*Approach to a light source*".

Chapter 7

Discussion and synthesis of results

This PhD Thesis has addressed two of the main challenges of the evolution of communication in SR systems: (1) the transferability of the same communication system in multiple independent tasks, and (2) the emergence of complex communications beyond signaling. Problem (1) is, in turn, splited into two levels and addressed separately using a *divide-and-conquer* strategy. Specifically, two levels of transferability are defined: (a) *weak transferability* and (b) *strong transferability*. The former refers to SR systems that can be used in multiple tasks without its re-design, but requiring re-evolving the system for each task. In contrast, the latter one goes a step beyond, being defined as the capability to transfer the same previously evolved communication to multiple tasks.

Chapter 4 sheds some light to the problem of *weak transferability*, presenting a minimal communication system that can be used in (at least) three independent SR tasks with very minor desing changes. The robots were endowed with a minimal communication system based on IR technology, that uses symbolic messages to encode the information being transmitted. Moreover, the robots were controlled by the same CTRNN-based architecture, that was evolved independently for each task using a GA. The communication system of this Chapter was assessed in three different SR primitives, namely, the election of a leader of the group, the identification of the swarm perimeter, and the heading orientation consensus. Even though these tasks can be defined as low-level behaviors, all of them required some degree of communication to be successfully solved. In line with most evolutionary SR research, an exhaustive statistical analysis of the evolved communication revealed that a signaling-based communication emerged in all the tasks. However, it was surprisingly shown that opposite communication semantics were evolved for each task:

- *Abstract semantics*, in which only the abstract message was important in the communication, evolved in the leader election task.
- *Situated semantics*, in which only the context information rather than the abstract message were communicative, did emerge in the identification of the swarm perimeter.
- *Purely situated semantics*, in which both the abstract message and the associated context were communicative, were evolved in the heading orientation consensus task.

This zoo of emerged semantics reinforced the *weak transferability* of the system, showing a tight adaptation of the evolutionary trajectories to the target task, using the same robot controller's architecture and design. Furthermore, it was demonstrated that the SR system and the operation of the evolved communication are scalable and robust in all of the three tasks. The scalability was assessed separately in each task using the best-performing GA individual of the evolved population and swarms of up to 50 robots.

In Chapter 5, the *strong transferability* in SR communications was tackled. For this aim, a novel communication module that, once evolved, can be used in multiple and utterly different SR tasks without the need to re-optimize was proposed. The communication module was based on *virtual state spaces*, denoted as *communication spaces*, which are independent to the real environment where physical robots exist, and it is where swarm communication and swarm organization takes place. The communication module was composed by *virtual regions*, each associated with some innately defined semantics that establish the robot's behavior in the physical world. A communication controller, based on CTRNNs, was evolved using a NEAT algorithm. The evolved controller defined the communication among the robots in the virtual space. The communication module and its *strong transferability* were evaluated in three popular SR tasks: an aggregation into groups of fixed sizes, a formation of target swarm geometries, and a basic foraging with robots' batteries. The exact same evolved communication was successfully used in all of these tasks. The aggregation in groups of desired sizes was correctly achieved, demonstrating the effectiveness of the communication module across multiple number of groups and group members. In all cases, groups were correctly formed according to the specified target size. Furthermore, distinct groups tended to repel each other, to avoid group merging. The formation control was also successfully solved using the previously evolved communication module. Different swarm formation geometries, i.e. triangle, square, rhombus, circle, and pentagon formations, were obtained with high accuracy and low distortion. It was shown that collective motion of the robot swarm while preserving the target formation geometry can be achieved as well. Moreover, the communication module allowed the switch of the swarm formation in real time during the same simulation. Specifically, it is shown an execution in which the robot swarm switches its formation geometry thirce, starting with a square formation, followed by a circle structure, and, lastly, iterating to a rhombus geometry. The communication module was also applied successfully to the foraging task. The results revealed that the robot swarm effectively performed a dynamic division of labour. Specifically, using the communication module, the following target division of labour was achieved: (i) 25% of the robots foraged from one food source, (ii) 25% of the swarm foraged from the remaining food source, and (iii) 50% of the robots stayed idle in the nest. Moreover, idle robots in the nest were able to correctly replace foragers that ran out of battery, successfully ensuring the desired 25% allocation to each food source. Scalability was also assessed in all of the three SR tasks. It was demonstrated that the proposed system is scalable at least up to swarms of 40 robots in the group aggregation and the formation control, with small degradations of the assessment errors. The communication module also scales suitably when the foraging experiment is tackled, revealing a correct achievement of the task goals with swarms up to 60 robots. All the scalability evaluations were accomplished using a statistically significant sample of fifty independent simulations for each swarm size and SR task.

The emergence of more complex forms of communication beyond signaling in SR was treated in Chapter 6. Specifically, it was shown that a swarm is able to self-organize and converge to a common compositional lexicon to refer to the different geometrical objects (of diverse shapes, colors and qualities) in the environment. The robots started with no prior knowledge neither about the external objects nor the words that are used to name them. Instead, they discovered them through exploration and interaction with the environment, and converged to a common lexicon using language games, local communication, and *semiotics*. The acquired words and meanings were perceptually grounded, satisfying the SGP. Moreover, the emerged lexicon displayed compositionality properties, implying that new words were created as the combination of previously acquired ones. Several metrics were carefully designed to fully evaluate the effectiveness and correct acquistion of a unique lexicon that is shared by all the robots. These metrics were the Lexicon Sharing Score (LSS), the Compositionality Sharing Score (CSS), and the number of total words v. The LSS and the CSS measured the degree of lexicon and compositionality overlapping and sharing along the swarm, respectively. Firstly, an experiment using a swarm of 20 robots and with three types of external objects (low quality red cubes, high quality blue spheres, and high quality blue cubes) was accomplished. The results revealed that the robot swarm was perfectly capable of acquiring the shared lexicon and set of semiotic symbols (see Section 6.2 for further details). Specifically, LSS and CSS metrics near 1 (optimal value) were obtained on average using fifty independent executions with random initializations of the position of robots and objects. Moreover, the total number of words suitably converged, on average, to the correct values of 6 elementary words (that do not emerge through compositionality) and 11 composed words (that emerge through compositionality). A second experiment was carried out with five external objects (low quality red cubes, high quality green cubes, high quality green spheres, low quality red spheres, and low quality red pyramids), and with an additional focus on the scalability of the system. Specifically, the emergence of the compositional lexicon was assessed with swarm sizes ranging from 10 up to 100 robots. For every swarm size, fifty independent simulations with independent initial conditions were collected and analysed. The results showed the correct scalability of the system up to (at least) swarms of 100 robots. Nearly optimal values of LSS and CSS were obtained for the lower swarm sizes, whereas the degradation of these metrics was considerably small for the large robot swarms. Moreover, such deterioration was mostly caused in terms of compositionality acquisition, so that the self-organization of elementary words was nearly optimal even in the larger swarms.

Chapter 8

Conclusions and Future Work

This Chapter summarizes and concludes this PhD Thesis. In Section 8.1, the main conclusions of the Thesis are detailed. Subsequently, Section 8.2 presents the future improvement and lines of research that result from this PhD Thesis. Finally, the author contributions are described in Section 8.3, reviewing the journal and congress papers that were published or submitted by the author under the scope of this PhD Thesis.

8.1 Conclusions

The SR field has reached an stagnation point in the emergence of inter-robot communication using evolutionary algorithms. Firstly, this stagnation is caused by a lack of standardization and generalization to multiple tasks. In other words, most of the SR studies tackling the evolution and origins of communication in societies of robots are designed ad hoc for specific tasks, environemnts, or communication systems. Moreover, even though many studies used CTRNN-based robot controllers and GAs as the optimization algorithm, there are not standaridized design and evolutionary principles that ease the evolution of communication from an engineering perspective. For instance, there is not a consensus on the communication technology, the size and encoding of the message (i.e. based on symbols or signals), or the architecture of the robot controller. Moreover, most of the SR studies targeting the evolution of communication result in simple a signaling communications. Thus, there is a strong barrier to reach the emergence of more complex communications, such as language.

In this PhD Thesis, these challenging issues are addressed in detail. Two different SR communication systems that use evolutionary algorithms to reach the emergence of its mechanics and semantics were proposed. Moreover, the main focus of these communication frameworks is to achieve *transferability* — a property that is defined as the capability to use the same communication in multiple SR tasks. One of the proposed communication systems only presented *weak transferability*, so that the same communication system was applied to three different SR tasks without its re-design, but requiring re-evolving the system for each task. In addition, a second SR framework was proposed and evolved, with the last aim of achieving *strong transferability*. Strong transferability is the property of an SR communication system to be applied to multiple tasks without requiring neither its re-design

nor its re-evolution. We demonstrated that the proposed framework is strongly transferable to threee popular SR tasks — aggregation, formation control, and foraging. Lastly, this PhD Thesis also tackles the strong barrier that hinders the achievement of language-based communications. Specifically, the problem of the self-organization of a shared lexicon in robot swarms was addressed and solved. The proposed lexicon emergence system is in line with the Symbol Grounding Problem, so that words and symbols are perceptually grounded on external objects based on the robot's experience. To address this challenging problem, language games, hebbian learning, and semiotics techniques were used.

8.2 Future Work

Multiple lines of future research emerged from this PhD Thesis. Firstly, the following general improvements, related to all the Thesis chapters, are considered:

- **Implementation in real robots**: all the experiments of this PhD Thesis were accomplished using robotics simulation tools and techniques. Even thought these simulations mimic the real physics, kinematics and dynamics accurately, there is still a critical *reality gap* that should be considered as future work. Specifically, the experiments should be replicated with real robots in controlled environments, to fully assess the effectiveness of the presented models and techniques in real-world missions.
- **Migration to other robots and environments**: only terrestrial mobile robots that resemble the e-puck robot [107] are considered in this work. Therefore, an extension of this PhD Thesis would be to assess the performance of the evolved communications using other types of terrestrial robots (e.g. hexapod or humanoid robots, among many others). Furthermore, more complex environments that hinder the robots' navigation and task accomplishment can be conceived. For example, rugged or uneven terrains can provide simulations that reflect more faithfully real-world applications. Moreover, migration to underwater and aerial environments, using swarms of drones and Unmanned Underwater Vehicles (UUVs), respectively, can be also an important extension of this PhD Thesis.
- **New neural architectures and optimization algorithms**: other neuron models and ANN architectures can be considered in the future. In this PhD Thesis, all the neural controllers are based on CTRNN models, which offer a low-level and simple solution that simplify how biological neurons work. However, other less realistic yet more computationally powerful ANNs could provide more complex emerged communications. Specifically, deep learning models, such as Long Short-Term Mermory (LSTM) neural networks or transformers, would be highly promising alternatives. Furthermore, the application of more advanced evolutionary algorithms can be an important improvement to reach more elaborate communications.
- **Reinforcement learning**: RL is left as future work as an alternative to evolutionary algorithms for reaching the emergence of communication in SR. In contrast to the evolution of communication, in RL the communication semantics are learnt during the robots' lifetime. Moreover, a highly primising research line to be considered in the future would be to combine evolution and lifetime learning (*EVO-DEVO*), so that the

emergence of communication can simulaneously coexist at two time scales.

The following future improvements resulting from the evolution of signaling communications in SR primitives are considered:

- **Incremental evolution and multitasking**: *strong transferability* cannot be achieved in the communication system of Chapter 4. However, an ambitious future extension of this chapter would be evolve a robot's neural controller capable of solving multiple tasks sequentially via incremental evolution. Firstly, the robot controller would be evolved to successfully perform a single task (as in the experiments of the Chapter). However, once the first task has been succesfully optimized and an effective communication has evolved, the evolutionary process would be redirected towards tackling a different task. However, the traits and communication semantics evolved for the original task should not be lost, and, instead, the robot swarm should ideally retain the capability to solve both tasks.
- **Assessment in more complex tasks**: the tasks that are solved in Chapter 4 correspond to *primitive* collective behaviors. Therefore, an extension of the proposed communication system to more complex and elaborate SR tasks (e.g. flocking, foraging, or formation control) should be considered as future work.
- **Fitness function optimization**: the fitness function is one of most important design steps that will define the outcome of the evolutionary algorithm. It can completely alter the evolved communication semantics. Therefore, future work can be devoted to the improvement of the fitness functions of this chapter, using all the newly acquired experience and knowledge acquired during the overall development of the PhD Thesis.

The following future improvements resulting from the evolution of transferable and self-organized communication modules are considered:

- **Very constrained and low-range communications**: the most immediate extension of Chapter 5 would be to assess the limits of the communication range and constrains. Specifically, the amount of tolerable communication noise and interferences as well as the feasibility of the communication module in very low range communicaitons should be evaluated.
- **Improvement of robot's positioning**: as it was not the focus of the PhD Thesis, the calculations and estimations of the positioning of the robots in some of the SR tasks of this chapter were simplified. Therefore, future improvements should be devoted to such in-depth estimation using advanced techniques such as *social odometry*, multi-agent localization, or SLAM.
- **Generalization to n-dimensional communication spaces**: The communication spaces used in this PhD Thesis are restricted to one and two dimensions. The generalization to n dimensions requires additional mathematical development on the distance, angle, and virtual region definitions. This extension is left as future work.
- **Learning of the communication space topology**: the communication spaces of Chapter 5, either one-dimensional or two-dimensional, are manually designed by the

researcher. That is, the number of regions and landmarks are fixed before the execution of the experiments. Therefore, a highly promising line of future research would be to initiate with a single virtual region, and, thereafter, acquire new ones through a process of lifetime learning during the experiment's execution. Owing to the fact that all the robots should share the same virtual regions and communication space topology, this learning could be combined with cultural transmission. It should be remarked that NEAT would still be responsible for the evolution of the communication controller (f_{comm}) and the virtual navigation of the robots.

- **Learning of primitive behaviors**: in line with the previous improvement, the *primitive behaviors* and the *primitive selector* could also be acquired through the lifetime learning and cultural transmission of the robot swarm. There are two levels of complexity of this future line of research:

  (1) The *primitive behaviors* are handcrafted and collected in the *primitive pool*, and it is only the *primitive selector* (the mapping between virtual regions and primitives) the building block that is learnt.

  (2) In addition to the previous level (1), the primitive behaviors themselves are also learnt or evolved.

  In both cases, the learning of behaviors should be simulateous to the learning of the communication space topology, described above.

- **Other communication controller techniques**: in this Chapter, the virtual navigation of the robots in the communication space is done using the communication controller, which is an evolved CTRNN. Other control techniques can be considered as an alternative method. For example, the virtual navigation can be performed using a physics-based approach, so that robots are *virtually* represented as particles in the communication space, that are attracted or repelled by landmarks and other particles.

The following future improvements resulting from the emergence and self-organization of a shared and perceptually grounded lexicon are considered:

- **Use of lexicon in robot conversations**: the self-organization of a shared compositional lexicon in robot swarms was achieved in Chapter 6. However, the use of the lexicon and its compositionality in actual conversations between robots is not explored. This new line of research can only be assessed when a solid basis of words and, thus, semiotic symbols, has been acquired and agreed across the swarm. Furthermore, it requires more complex goal tasks beyond the exploration and object identification tackled in this chapter.

- **Exploration of more complex compositional rules**: in this PhD Thesis, compositionality is considered from the perspective of *meanings* that are combined to create other, more complex, *meanings*. Even though this symbol composition is of essence for the emergence of language-like communications, other forms of compositionality can be also considered. The exploration of grammar rules and other syntactical relations are promising line of research that should be considered in the future. Furthermore, these new compositional rules should be also learnt and be culturally transimited across the

robot swarm.

- **Other types of external referents**: words and meanings are perceptually grounded on geometrical objects with different colors, shapes, and qualities. An interesting extension of this process would be to ground symbols on other external referents beyond the ones of this PhD Thesis. Semiotic symbols could be grounded on more abstract concepts such as actions and behaviors of the robots. A new word can be created, for example, for the motor actions "*Rotate right wheel counter clockwise*" or "*Rotate left wheel counter clockwise*", and, in turn, these new concepts can be combined to create a word grounded on the behavior "*Move straight forward*".
- **Connection between the transferable communication module and the emergence of lexicon**: the same algorithms and techniques of Chapter 6, employed to achieve the self-organization of a shared lexicon, can be used to improve the communication module of Chapter 5. Specifically, it can be applied to learn from zero the communication space topology, primitive behaviors, and primitive selector. That is, the self-organized semiotic symbols, grounded on learnt primitive behaviors, can be associated to virtual regions in the communication space. Such grounding and associative process would be accomplished in an online way during the robots' lifetime, using language games, semiotics, and Hebbian learning.
- **Large Language Models**: the emergence of communication and the self-orgnanization of a common lexicon can be combined with Large Language Models (LLMs). For instance, LLMs can be used as the cognition behind the planning, actuation, communication, and self-organization of the robot swarm, or to study and analyze some previously emerged communication semantics.

8.3 Review of Contributions

This dissertation describes original research carried out by the author. It has not been previously submitted to the Universidad Politécnica de Madrid, nor to any other university for the awarding of any degree. In this Section, all the papers published or under review for publication by the author (2 papers published in journals, 2 papers under review in journals, and 2 conferences), are explained and linked with this PhD Thesis. The corresponding publications are detailed in the following.

The early work published by the author was in 2021, entitled as *Evolution of situated and abstract communication in leader selection and borderline identification swarm robotics problems* [25] and published in the *Applied Sciences* journal. This paper studied how the same communication system and robot controller can lead to the emergence of different kinds of signaling communications within the robot swarm depending on the task being faced. Specifically, the paper tackles both the leader election and the swarm frontier identification SR tasks. In addition, a conference paper entitled as *Emergence of Communication Through Artificial Evolution in an Orientation Consensus Task in Swarm Robotics* [260] was published in 2023 in the International Conference on Artificial Intelligence Applications and Innovations. In addition, a journal paper entitied as *Emergence of flocking behaviors transferring previously*

evolved alignment robot controllers [261] was recently published in the *Evolving Systems* journal in 2025. Both of these works, [260] and [261], extend the previously published paper in [25]. Specifically, these publications respectively focus on different aspects of an evolved robot alignment behavior. On the one hand, the publication in [260] is related to the evolved communication itself and an exhaustive analysis of its emerged semantics. On the other hand, [261] is devoted to the transferability of the evolved communication and robot controller, assessing its connection and importance within flocking behaviors. A conference paper was also published in 2023 in the XLIV Jornadas de Automática [262], evaluating the robotics simulation tool used along this PhD Thesis in two robotics tasks that are evolved and performed by the robots sequentially. All of these four publications (two journal papers and two conference papers) correspond to Chapter 4 of this PhD Thesis.

In October 2024, a paper entitled as *Evolution of transferable and self-organized communication modules for solving multiple swarm robotics tasks* was submitted to the *IEEE Transactions on Cybernetics* journal. The submitted manuscript is under review at the moment of the submission of this PhD dissertation. This paper corresponds to Chapter 5 and it presents a novel *communication module* to be used in SR systems to plan and coordinate the robot's physical behavior collectively. The proposed system can be transfered to multiple independent SR tasks without the need to re-evolve the system. It is successfully assessed in three popular SR tasks: group aggregation, formation control, and foraging. Finally, the contents of Chapter 6 were submitted as a journal paper in December 2024 to the *IEEE Transactions on Evolutionary Computation* journal. This manuscript is entitled as *Cultural evolution of perceptually grounded compositional lexicons in swarm robotics systems*, and it is also under review at the moment of the submission of this PhD dissertation. It proposes the self-organization and agreement of a common lexicon in a swarm of mobile robots to name multiple geometric objects, of different colors and shapes, in the environment. The acquisition of the emerged lexicon is fully automatic and distributed using language games and semiotics. The words in the agent's lexicon satisfy the Harnad's Symbol Grounding Problem [13], as they are physically grounded based on the robots' own experience. Moreover, the self-organized lexicon displays compositionality properties, that are obtained through Hebbian learning.

Bibliography

Bibliography

[1] C. Ampatzis, E. Tuci, V. Trianni, and M. Dorigo, "Evolution of signaling in a multi-robot system: Categorization and communication," *Adaptive Behavior*, vol. 16, no. 1, pp. 5–26, 2008.

[2] E. Tuci, "An investigation of the evolutionary origin of reciprocal communication using simulated autonomous agents," *Biological Cybernetics*, vol. 101, no. 3, pp. 183–199, 2009.

[3] G. Beni, "From Swarm Intelligence to Swarm Robotics," in *Swarm Robotics*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005, pp. 1–9.

[4] E. Şahin, "Swarm Robotics: From Sources of Inspiration to Domains of Application," in *Swarm Robotics*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005, pp. 10–20.

[5] A. Campo, A. Gutiérrez, S. Nouyan, C. Pinciroli, V. Longchamp, S. Garnier, and M. Dorigo, "Artificial pheromone for path selection by a foraging swarm of robots," *Biological Cybernetics*, vol. 103, no. 5, pp. 339–352, 2010.

[6] J. P. Hecker, K. Letendre, K. Stolleis, D. Washington, and M. E. Moses, "Formica ex machina: Ant swarm foraging from physical to virtual and back again," in *Swarm Intelligence*, M. Dorigo, M. Birattari, C. Blum, A. L. Christensen, A. P. Engelbrecht, R. Groß, and T. Stützle, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 252–259.

[7] Á. Gutiérrez, E. Tuci, and A. Campo, "Evolution of Neuro-Controllers for Robots' Alignment using Local Communication," *International Journal of Advanced Robotic Systems*, vol. 6, no. 1, p. 6, 2009.

[8] E. Tuci and C. Ampatzis, "Evolution of acoustic communication between two cooperating robots," in *Advances in Artificial Life*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 395–404.

[9] R. O'Grady, A. L. Christensen, and M. Dorigo, "SWARMORPH: Multirobot Morphogenesis Using Directional Self-Assembly," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 738–743, Jun. 2009.

[10] F. Ducatelle, G. A. Di Caro, C. Pinciroli, and L. M. Gambardella, "Self-organized cooperation between robotic swarms," *Swarm Intelligence*, vol. 5, no. 2, p. 73, Mar.

2011.

[11] N. Cambier, R. Miletitch, V. Frémont, M. Dorigo, E. Ferrante, and V. Trianni, "Language Evolution in Swarm Robotics: A Perspective," *Frontiers in Robotics and AI*, vol. 7, 2020.

[12] N. Nedjah and L. S. Junior, "Review of methodologies and tasks in swarm robotics towards standardization," *Swarm and Evolutionary Computation*, vol. 50, p. 100565, 2019.

[13] S. Harnad, "The symbol grounding problem," *Physica D: Nonlinear Phenomena*, vol. 42, no. 1-3, pp. 335–346, 1990.

[14] P. Vogt, "The physical symbol grounding problem," *Cognitive Systems Research*, vol. 3, no. 3, pp. 429–457, 2002, situated and Embodied Cognition.

[15] C. F. Hockett and C. D. Hockett, "The origin of speech," *Scientific American*, vol. 203, no. 3, pp. 88–97, 1960.

[16] C. Lyon, C. L. Nehaniv, and A. Cangelosi, *Emergence of communication and language*. Springer Science & Business Media, 2010.

[17] J. Goldstein, "Emergence as a construct: History and issues," *Emergence*, vol. 1, no. 1, pp. 49–72, 1999.

[18] T. De Wolf and T. Holvoet, "Emergence versus self-organisation: Different concepts but promising when combined," in *Engineering Self-Organising Systems*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005, pp. 1–15.

[19] M. B. L. Dempster, "A self-organizing systems perspective on planning for sustainability," Ph.D. dissertation, Citeseer, 1998.

[20] D. J. T. Sumpter, *Collective Animal Behavior*. Princeton, NJ: Princeton University Press, 2010.

[21] Y. Ozkan-Aydin and D. I. Goldman, "Self-reconfigurable multilegged robot swarms collectively accomplish challenging terradynamic tasks," *Science Robotics*, vol. 6, no. 56, p. eabf1628, 2021.

[22] K. Yasui, S. Takano, T. Kano, and A. Ishiguro, "Adaptive centipede walking via synergetic coupling between decentralized control and flexible body dynamics," *Frontiers in Robotics and AI*, vol. 9, 2022.

[23] M. Rubenstein, A. Cornejo, and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm," *Science*, vol. 345, no. 6198, pp. 795–799, 2014.

[24] V. Karpov and I. Karpova, "Leader election algorithms for static swarms," *Biologically Inspired Cognitive Architectures*, vol. 12, pp. 54–64, 2015.

[25] R. Sendra-Arranz and Á. Gutiérrez, "Evolution of Situated and Abstract Communication in Leader Selection and Borderline Identification Swarm Robotics Problems," *Applied Sciences*, vol. 11, no. 8, p. 3516, 2021.

[26] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.

[27] H. L. Kwa, J. Philippot, and R. Bouffanais, "Effect of swarm density on collective tracking performance," *Swarm Intelligence*, vol. 17, no. 3, pp. 253–281, 2023.

[28] A. F. Winfield, "Foraging robots." *Encyclopedia of complexity and systems science*, vol. 6, pp. 3682–3700, 2009.

[29] E. O. Wilson, *The insect societies.* Cambridge, MA: Harvard University Press, 1971.

[30] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems.* Oxford University Press, 10 1999.

[31] S. Camazine, J.-L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraula, and E. Bonabeau, "Self-organization in biological systems," in *Self-Organization in Biological Systems.* Princeton studies in complexity. Princeton university press, Princeton, 2003.

[32] R. Beckers, J. Deneubourg, and S. Goss, "Trails and u-turns in the selection of a path by the ant lasius niger," *Journal of Theoretical Biology*, vol. 159, no. 4, pp. 397–415, 1992.

[33] K. v. Frisch, *The dance language and orientation of bees.* Harvard University Press, 1993.

[34] K. E. Gardner, T. D. Seeley, and N. W. Calderone, "Do honeybees have two discrete dances to advertise food sources?" *Animal Behaviour*, vol. 75, no. 4, pp. 1291–1300, 2008.

[35] R. M. Seyfarth, D. L. Cheney, and P. Marler, "Monkey responses to three different alarm calls: Evidence of predator classification and semantic communication," *Science*, vol. 210, no. 4471, pp. 801–803, 1980.

[36] J. T. Emlen, "Flocking behavior in birds," *The Auk*, vol. 69, no. 2, pp. 160–170, 1952.

[37] I. L. Bajec and F. H. Heppner, "Organized flight in birds," *Animal Behaviour*, vol. 78, no. 4, pp. 777–789, 2009.

[38] Wilson, Edward O., *Sociobiology: The new synthesis.* Harvard University Press, 2000.

[39] J. H. Reif and H. Wang, "Social potential fields: A distributed behavioral control for autonomous robots," *Robotics and Autonomous Systems*, vol. 27, no. 3, pp. 171–194, 1999.

[40] W. M. Spears, D. F. Spears, J. C. Hamann, and R. Heil, "Distributed, physics-based control of swarms of vehicles," *Autonomous robots*, vol. 17, no. 2-3, pp. 137–162, 2004.

[41] W. Kerr, D. Spears, W. Spears, and D. Thayer, "Two formal gas models for multi-agent sweeping and obstacle avoidance," in *International Workshop on Formal Approaches to Agent-Based Systems.* Springer, 2004, pp. 111–130.

[42] Q. Chen, S. M. Veres, Y. Wang, and Y. Meng, "Virtual spring-damper mesh-based

formation control for spacecraft swarms in potential fields," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 3, pp. 539–546, 2015.

[43] B. Khaldi and F. Cherif, "Swarm robots circle formation via a virtual viscoelastic control model," in *2016 8th International Conference on Modelling, Identification and Control (ICMIC)*, 2016, pp. 725–730.

[44] B. Khaldi, F. Harrou, F. Cherif, and Y. Sun, "Flexible and efficient topological approaches for a reliable robots swarm aggregation," *IEEE Access*, vol. 7, pp. 96 372–96 383, 2019.

[45] E. Ferrante, A. E. Turgut, M. Dorigo, and C. Huepe, "Collective motion dynamics of active solids and active crystals," *New Journal of Physics*, vol. 15, no. 9, p. 095011, sep 2013.

[46] M. Raoufi, A. E. Turgut, and F. Arvin, "Self-organized collective motion with a simulated real robot swarm," in *Towards Autonomous Robotic Systems: 20th Annual Conference, TAROS 2019, London, UK, July 3–5, 2019, Proceedings, Part I 20.* Springer, 2019, pp. 263–274.

[47] M. Bahaidarah, F. Rekabi-Bana, O. Marjanovic, and F. Arvin, "Swarm flocking using optimisation for a self-organised collective motion," *Swarm and Evolutionary Computation*, vol. 86, p. 101491, 2024.

[48] X. Wang, X. Wang, D. Zhang, and L. Shen, "A liquid sphere-inspired physicomimetics approach for multiagent formation control," *International Journal of Robust and Nonlinear Control*, vol. 28, no. 15, pp. 4565–4583, 2018.

[49] Z. Firat, E. Ferrante, Y. Gillet, and E. Tuci, "On self-organised aggregation dynamics in swarms of robots with informed robots," *Neural Computing and Applications*, vol. 32, no. 17, pp. 13 825–13 841, 2020.

[50] O. Soysal and E. Sahin, "Probabilistic aggregation strategies in swarm robotic systems," in *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, Jun. 2005, pp. 325–332.

[51] O. Soysal and E. Şahin, "A macroscopic model for self-organized aggregation in swarm robotic systems," in *Swarm Robotics*, E. Şahin, W. M. Spears, and A. F. T. Winfield, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 27–42.

[52] L. Bayindir and E. Sahin, "Modeling self-organized aggregation in swarm robotic systems," in *2009 IEEE Swarm Intelligence Symposium*, 2009, pp. 88–95.

[53] N. Cambier, D. Albani, V. Frémont, V. Trianni, and E. Ferrante, "Cultural evolution of probabilistic aggregation in synthetic swarms," *Applied Soft Computing*, vol. 113, p. 108010, 2021.

[54] A. Font Llenas, M. S. Talamali, X. Xu, J. A. R. Marshall, and A. Reina, "Quality-sensitive foraging by a robot swarm through virtual pheromone trails," in *Swarm Intelligence*, M. Dorigo, M. Birattari, C. Blum, A. L. Christensen, A. Reina, and V. Trianni, Eds. Cham: Springer International Publishing, 2018, pp. 135–149.

[55] J. P. Hecker and M. E. Moses, "Beyond pheromones: evolving error-tolerant, flexible, and scalable ant-inspired robot swarms," *Swarm Intelligence*, vol. 9, no. 1, pp. 43–70, 2015.

[56] G. M. Fricke, J. P. Hecker, A. D. Griego, L. T. Tran, and M. E. Moses, "A distributed deterministic spiral search algorithm for swarms," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4430–4436.

[57] A. Gutiérrez, A. Campo, F. Monasterio-Huelin, L. Magdalena, and M. Dorigo, "Collective decision-making based on social odometry," *Neural Computing & Applications*, vol. 19, no. 6, pp. 807–823, 2010.

[58] M. Bahaidarah, O. Marjanovic, F. Rekabi-Bana, and F. Arvin, "An optimised robot swarm flocking with genetic algorithm," in *2023 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2023, pp. 1823–1828.

[59] M. B. Bezcioglu, B. Lennox, and F. Arvin, "Self-organised swarm flocking with deep reinforcement learning," in *2021 7th International Conference on Automation, Robotics and Applications (ICARA)*, 2021, pp. 226–230.

[60] M. Dorigo, V. Trianni, E. Şahin, R. Groß, T. H. Labella, G. Baldassarre, S. Nolfi, J.-L. Deneubourg, F. Mondada, D. Floreano, and L. M. Gambardella, "Evolving self-organizing behaviors for a swarm-bot," *Autonomous Robots*, vol. 17, p. 223–245, 2004.

[61] J. Gomes, P. Urbano, and A. L. Christensen, "Evolution of swarm robotics systems with novelty search," *Swarm Intelligence*, vol. 7, no. 2, pp. 115–144, 2013.

[62] G. Vásárhelyi, C. Virágh, G. Somorjai, T. Nepusz, A. E. Eiben, and T. Vicsek, "Optimized flocking of autonomous drones in confined environments," *Science Robotics*, vol. 3, no. 20, p. eaat3536, 2018.

[63] H. Celikkanat, "Optimization of felf-organized flocking of a robot swarm via evolutionary strategies," in *2008 23rd International Symposium on Computer and Information Sciences*, 2008, pp. 1–4.

[64] R. P. Ramos, S. M. Oliveira, S. M. Vieira, and A. L. Christensen, "Evolving flocking in embodied agents based on local and global application of Reynolds' rules," *PLOS ONE*, vol. 14, no. 10, pp. 1–16, Oct. 2019, publisher: Public Library of Science.

[65] H. Kwasnicka, U. Markowska-Kaczmar, and M. Mikosik, "Flocking behaviour in simple ecosystems as a result of artificial evolution," *Applied Soft Computing*, vol. 11, no. 1, pp. 982–990, 2011.

[66] L. Ma, W. Bao, X. Zhu, M. Wu, Y. Wang, Y. Ling, and W. Zhou, "O-flocking: Optimized flocking model on autonomous navigation for robotic swarm," in *Advances in Swarm Intelligence*, Y. Tan, Y. Shi, and M. Tuba, Eds. Cham: Springer International Publishing, 2020, pp. 628–639.

[67] J. Ericksen, M. Moses, and S. Forrest, "Automatically evolving a general controller for robot swarms," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017, pp. 1–8.

[68] K. Morihiro, T. Isokawa, H. Nishimura, and N. Matsui, "Emergence of flocking behavior based on reinforcement learning," in *Knowledge-Based Intelligent Information and Engineering Systems*, B. Gabrys, R. J. Howlett, and L. C. Jain, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 699–706.

[69] M. Durve, F. Peruani, and A. Celani, "Learning to flock through reinforcement," *Phys. Rev. E*, vol. 102, p. 012601, Jul 2020.

[70] I. Navarro, E. Di Mario, and A. Martino, "Distributed vs. centralized particle swarm optimization for learning flocking behaviors," ser. Artificial Life Conference Proceedings, vol. ECAL 2015: the 13th European Conference on Artificial Life, 07 2015, pp. 302–309.

[71] H. Qiu and H. Duan, "A multi-objective pigeon-inspired optimization approach to uav distributed flocking among obstacles," *Information Sciences*, vol. 509, pp. 515–529, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025518305103

[72] S. Nolfi and D. Floreano, *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*. Cambridge, MA, USA: MIT press, 2000.

[73] D. Floreano, P. Dürr, and C. Mattiussi, "Neuroevolution: from architectures to learning," *Evolutionary Intelligence*, vol. 1, no. 1, p. 47–62, 2008.

[74] V. Trianni, R. Groß, T. H. Labella, E. Şahin, and M. Dorigo, "Evolving aggregation behaviors in a swarm of robots," in *Advances in Artificial Life*, W. Banzhaf, J. Ziegler, T. Christaller, P. Dittrich, and J. T. Kim, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 865–874.

[75] Q. Lu, M. E. Moses, and J. P. Hecker, "A scalable and adaptable multiple-place foraging algorithm for ant-inspired robot swarms," *arXiv preprint arXiv:1612.00480*, 2016.

[76] Q. Lu, J. P. Hecker, and M. E. Moses, "The mpfa: A multiple-place foraging algorithm for biologically-inspired robot swarms," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 3815–3821.

[77] M. Quinn, "Evolving communication without dedicated communication channels," in *Advances in Artificial Life*. Springer, 2001, pp. 357–366.

[78] D. Marocco and S. Nolfi, "Origins of communication in evolving robots," in *From Animals to Animats 9. SAB 2006. Lecture Notes in Computer Science()*, vol. 4095. Springer, 2006, pp. 789–803.

[79] D. Marocco and S. Nolfi, "Emergence of communication in embodied agents evolved for the ability to solve a collective navigation problem," *Connection Science*, vol. 19, no. 1, pp. 53–74, 2007.

[80] D. Marocco, A. Cangelosi, and S. Nolfi, "The emergence of communication in evolutionary robots," *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 361, no. 1811, pp. 2397–2421, 2003.

[81] D. Floreano, S. Mitri, S. Magnenat, and L. Keller, "Evolutionary conditions for the emergence of communication in robots," *Current biology*, vol. 17, no. 6, pp. 514–519, 2007.

[82] A. Bernard, S. Wischmann, D. Floreano, and L. Keller, "The evolution of behavioral cues and signaling in displaced communication," *PLOS Computational Biology*, vol. 19, no. 3, p. e1010487, 2023.

[83] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002.

[84] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[85] L. Buşoniu, R. Babuška, and B. De Schutter, *Multi-agent Reinforcement Learning: An Overview*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 183–221.

[86] H. Iima and Y. Kuroe, "Swarm reinforcement learning method for a multi-robot formation problem," in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, 2013, pp. 2298–2303.

[87] C. Speck and D. J. Bucci, "Distributed uav swarm formation control via object-focused, multi-objective sarsa," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 6596–6601.

[88] B. Jin, Y. Liang, Z. Han, and K. Ohkura, "Generating collective foraging behavior for robotic swarm using deep reinforcement learning," *Artificial Life and Robotics*, vol. 25, no. 4, pp. 588–595, 2020.

[89] W. Luo, Q. Tang, C. Fu, and P. Eberhard, "Deep-sarsa based multi-uav path planning and obstacle avoidance in a dynamic environment," in *Advances in Swarm Intelligence*, Y. Tan, Y. Shi, and Q. Tang, Eds. Cham: Springer International Publishing, 2018, pp. 102–111.

[90] R. Brooks, "A robust layered control system for a mobile robot," *IEEE journal on robotics and automation*, vol. 2, no. 1, pp. 14–23, 1986.

[91] J. C. Varughese, H. Hornischer, P. Zahadat, R. Thenius, F. Wotawa, and T. Schmickl, "A swarm design paradigm unifying swarm behaviors using minimalistic communication," *Bioinspiration & Biomimetics*, vol. 15, no. 3, p. 036005, mar 2020.

[92] M. Duarte, S. M. Oliveira, and A. L. Christensen, "Evolution of hybrid robotic controllers for complex tasks," *Journal of Intelligent & Robotic Systems*, vol. 78, no. 3-4, pp. 463–484, 2015.

[93] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '87. New York, NY, USA: Association for Computing Machinery, 1987, pp. 25–34.

[94] R. Jeanson, C. Rivault, J.-L. Deneubourg, S. Blanco, R. Fournier, C. Jost, and G. Ther-

aulaz, "Self-organized aggregation in cockroaches," *Animal Behaviour*, vol. 69, no. 1, pp. 169–180, 2005.

[95] S. Garnier, C. Jost, R. Jeanson, J. Gautrais, M. Asadpour, G. Caprari, and G. Theraulaz, "Aggregation behaviour as a source of collective decision in a group of cockroach-like-robots," in *Advances in Artificial Life*, M. S. Capcarrère, A. A. Freitas, P. J. Bentley, C. G. Johnson, and J. Timmis, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 169–178.

[96] J. Vanualailai and B. N. Sharma, "A lagrangian-based swarming behavior in the absence of obstacles," in *Proceedings of the Workshop on Mathematical Control Theory*, 2010, pp. 119–135.

[97] A. Gasparri, A. Priolo, and G. Ulivi, "A swarm aggregation algorithm for multi-robot systems based on local interaction," in *2012 IEEE International Conference on Control Applications*, 2012, pp. 1497–1502.

[98] A. Borzì and S. Wongkaew, "Modeling and control through leadership of a refined flocking system," *Mathematical Models and Methods in Applied Sciences*, vol. 25, no. 02, pp. 255–282, 2015.

[99] O. Tuzel, G. Marcon dos Santos, C. Fleming, and J. A. Adams, "Learning based leadership in swarm navigation," in *Swarm Intelligence. ANTS 2018. Lecture Notes in Computer Science*, vol. 11172. Cham: Springer International Publishing, 2018, pp. 385–394.

[100] M. S. Güzel and H. Kayakökü, "A collective behaviour framework for multi-agent systems," in *Mechatronics and Robotics Engineering for Advanced and Intelligent Manufacturing*, D. Zhang and B. Wei, Eds. Cham: Springer International Publishing, 2017, pp. 61–71.

[101] O. Gigliotta, M. Mirolli, and S. Nolfi, "Communication based dynamic role allocation in a group of homogeneous robots," *Natural Computing*, vol. 13, pp. 391–402, 2014.

[102] A. E. Turgut, H. Çelikkanat, F. Gökçe, and E. Şahin, "Self-organized flocking in mobile robot swarms," *Swarm Intelligence*, vol. 2, no. 2, pp. 97–120, 2008.

[103] E. Ferrante, A. E. Turgut, N. Mathews, M. Birattari, and M. Dorigo, "Flocking in Stationary and Non-stationary Environments: A Novel Communication Strategy for Heading Alignment," in *Parallel Problem Solving from Nature, PPSN XI*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2010, pp. 331–340.

[104] S. W. Ekanayake and P. N. Pathirana, "Formations of robotic swarm: An artificial force based approach," *International Journal of Advanced Robotic Systems*, vol. 7, no. 3, p. 23, 2010.

[105] L. Qin, Y. Zha, Q. Yin, and Y. Peng, "Formation control of robotic swarm using bounded artificial forces," *The Scientific World Journal*, vol. 2013, no. 1, p. 194280, 2013.

[106] L. E. Barnes, M. A. Fields, and K. P. Valavanis, "Swarm formation control utilizing

elliptical surfaces and limiting functions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 6, pp. 1434–1445, 2009.

[107] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J. C. Zufferey, D. Floreano, and A. Martinoli, "The e-puck, a robot designed for education in engineering," *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions (IPCB, 2009)*, vol. 1, p. 59–65, 2009.

[108] A. Gutierrez, A. Campo, M. Dorigo, J. Donate, F. Monasterio-Huelin, and L. Magdalena, "Open e-puck range & bearing miniaturized board for local communication in swarm robotics," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3111–3116.

[109] B. G. Elkilany, A. A. Abouelsoud, and A. M. Fathelbab, "Adaptive formation control of robot swarms using optimized potential field method," in *2017 IEEE International Conference on Industrial Technology (ICIT)*, 2017, pp. 721–725.

[110] T. Balch and R. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, 1998.

[111] D. Xu, X. Zhang, Z. Zhu, C. Chen, and P. Yang, "Behavior-based formation control of swarm robots," *Mathematical Problems in Engineering*, vol. 2014, no. 1, p. 205759, 2014.

[112] K.-H. Tan and M. Lewis, "Virtual structures for high-precision cooperative mobile robotic control," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS '96*, vol. 1, 1996, pp. 132–139 vol.1.

[113] M. A. Lewis and K.-H. Tan, "High Precision Formation Control of Mobile Robots Using Virtual Structures," *Autonomous Robots*, vol. 4, no. 4, pp. 387–403, Oct. 1997.

[114] X. Wenlong, S. Jianbo, and L. Zongli, "New coordination scheme for multi-robot systems based on state space models," *Journal of Systems Engineering and Electronics*, vol. 19, no. 4, pp. 722–734, Aug. 2008.

[115] J. Su and W. Xie, "Motion Planning and Coordination for Robot Systems Based on Representation Space," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 1, pp. 248–259, Feb. 2011.

[116] C. Ruizhi and S. Jianbo, "Motion planning of multi-robot formation based on representation space," in *Proceedings of the 31st Chinese Control Conference*, Jul. 2012, pp. 6389–6394.

[117] G. Xiang and J. Su, "Interactive Natural Motion Planning for Robot Systems Based on Representation Space," *International Journal of Social Robotics*, vol. 12, no. 2, pp. 345–354, May 2020.

[118] A. Askari, M. Mortazavi, and H. A. Talebi, "Uav formation control via the virtual structure approach," *Journal of Aerospace Engineering*, vol. 28, no. 1, p. 04014047, 2015.

[119] I. Navarro and F. Matía, "A survey of collective movement of mobile robots," *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, p. 73, 2013.

[120] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks, "Collective memory and spatial sorting in animal groups," *Journal of Theoretical Biology*, vol. 218, no. 1, pp. 1–11, 2002.

[121] F. Cucker and S. Smale, "Emergent behavior in flocks," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 852–862, 2007.

[122] F. Cucker and J.-G. Dong, "Avoiding collisions in flocks," *IEEE Transactions on Automatic Control*, vol. 55, no. 5, pp. 1238–1243, 2010.

[123] J. Ma and E. M.-K. Lai, "Finite-time flocking control of a swarm of cucker-smale agents with collision avoidance," in *2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, 2017, pp. 1–6.

[124] Y. Han, D. Zhao, and Y. Sun, "Finite-time flocking problem of a cucker-smale-type self-propelled particle model," *Complexity*, vol. 21, pp. 354–361, 2016.

[125] F. Cucker and E. Mordecki, "Flocking in noisy environments," *Journal de Mathématiques Pures et Appliquées*, vol. 89, no. 3, pp. 278–296, 2008.

[126] R. Erban, J. Haškovec, and Y. Sun, "A cucker–smale model with noise and delay," *SIAM Journal on Applied Mathematics*, vol. 76, no. 4, pp. 1535–1557, 2016.

[127] J. Yan and X. Yin, "Flocking of the hybrid cucker–smale model," *Journal of the Franklin Institute*, vol. 360, no. 6, pp. 4016–4030, 2023.

[128] L. E. Beaver and A. A. Malikopoulos, "An overview on optimal flocking," *Annual Reviews in Control*, vol. 51, pp. 88–99, 2021.

[129] T. Balch and M. Hybinette, "Social potentials for scalable multi-robot formations," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 1, 2000, pp. 73–80 vol.1.

[130] D. Kim, H. Wang, G. Ye, and S. Shin, "Decentralized control of autonomous swarm systems using artificial potential functions: analytical design guidelines," in *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, vol. 1, 2004, pp. 159–164 Vol.1.

[131] A. Ruiz-Esparza-Rodriguez, M. S. Gonzalez-Chavez, and V. J. Gonzalez-Villela, "A flocking behavior model with artificial potential fields for the coordinated displacement of robotic swarms," in *2020 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE)*, 2020, pp. 65–70.

[132] A. Stranieri, E. Ferrante, A. E. Turgut, V. Trianni, C. Pinciroli, M. Birattari, and M. Dorigo, "Self-organized flocking with an heterogeneous mobile robot swarm." in *Advances in artificial life, ECAL 2011*, 2011, pp. 789–796.

[133] A. E. Turgut, F. Gokce, H. Celikkanat, L. Bayindir, and E. Sahin, "Kobot: A mo-

bile robot designed specifically for swarm robotics research," *Middle East Technical University, Ankara, Turkey, METU-CENG-TR Tech. Rep*, vol. 5, no. 2007, 2007.

[134] J. L. Deneubourg, S. Aron, S. Goss, and J. M. Pasteels, "The self-organizing exploratory pattern of the argentine ant," *Journal of Insect Behavior*, vol. 3, no. 2, pp. 159–168, 1990.

[135] A. Campo and M. Dorigo, "Efficient multi-foraging in swarm robotics," in *Advances in Artificial Life*, F. Almeida e Costa, L. M. Rocha, E. Costa, I. Harvey, and A. Coutinho, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 696–705.

[136] R. Groß and M. Dorigo, "Towards group transport by swarms of robots," *Int. J. Bio-Inspired Comput.*, vol. 1, no. 1/2, p. 1–13, 2009.

[137] R. Fujisawa, S. Dobata, K. Sugawara, and F. Matsuno, "Designing pheromone communication in swarm robotics: Group foraging behavior mediated by chemical substance," *Swarm Intelligence*, vol. 8, no. 3, pp. 227–246, 2014.

[138] R. Russell, "Heat trails as short-lived navigational markers for mobile robots," in *Proceedings of International Conference on Robotics and Automation*, vol. 4, 1997, pp. 3534–3539 vol.4.

[139] K. Sugawara, T. Kazama, and T. Watanabe, "Foraging behavior of interacting robots with virtual pheromone," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 3074–3079 vol.3.

[140] S. Garnier, M. Combe, C. Jost, and G. Theraulaz, "Do ants need to estimate the geometrical properties of trail bifurcations to find an efficient route? a swarm robotics test bed," *PLOS Computational Biology*, vol. 9, no. 3, pp. 1–12, 2013.

[141] A. Reina, A. J. Cope, E. Nikolaidis, J. A. R. Marshall, and C. Sabo, "Ark: Augmented reality for kilobots," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1755–1761, 2017.

[142] Y. Song, X. Fang, B. Liu, C. Li, Y. Li, and S. X. Yang, "A novel foraging algorithm for swarm robotics based on virtual pheromones and neural network," *Applied Soft Computing*, vol. 90, p. 106156, 2020.

[143] Q. Lu, A. D. Griego, G. M. Fricke, and M. E. Moses, "Comparing physical and simulated performance of a deterministic and a bio-inspired stochastic foraging strategy for robot swarms," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 9285–9291.

[144] G. M. Viswanathan, V. Afanasyev, S. V. Buldyrev, E. J. Murphy, P. A. Prince, and H. E. Stanley, "Lévy flight search patterns of wandering albatrosses," *Nature*, vol. 381, no. 6581, pp. 413–415, 1996.

[145] A. Mårell, J. P. Ball, and A. Hofgaard, "Foraging and movement paths of female reindeer: insights from fractal analysis, correlated random walks, and lévy flights," *Canadian Journal of Zoology*, vol. 80, no. 5, pp. 854–865, 2002.

[146] G. M. Fricke, J. P. Hecker, J. L. Cannon, and M. E. Moses, "Immune-inspired search strategies for robot swarms," *Robotica*, vol. 34, no. 8, p. 1791–1810, 2016.

[147] N. Johannes, K. Yara, and S. Pieter, "Hybrid foraging in patchy environments using spatial memory," *J. R. Soc. Interface*, vol. 17, p. 20200026, 2020.

[148] A. Schroeder, S. Ramakrishnan, M. Kumar, and B. Trease, "Efficient spatial coverage by a robot swarm based on an ant foraging model and the lévy distribution," *Swarm Intelligence*, vol. 11, no. 1, pp. 39–69, 2017.

[149] A. Gutiérrez, A. Campo, F. C. Santos, C. Pinciroli, and M. Dorigo, "Social odometry in populations of autonomous robots," in *Ant Colony Optimization and Swarm Intelligence, 6th International Conference, ANTS 2008, Brussels, Belgium, September 2008, Proceedings*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer-Verlag, 2008, pp. 371–378.

[150] A. Gutiérrez, A. Campo, F. C. Santos, F. Monasterio-Huelin, and M. Dorigo, "Social odometry: Imitation based odometry in collective robotics," *International Journal of Advanced Robotic Systems*, vol. 6, no. 2, pp. 129–136, 2009.

[151] E. Ferrante, A. E. Turgut, E. Duéñez-Guzmán, M. Dorigo, and T. Wenseleers, "Evolution of self-organized task specialization in robot swarms," *PLOS Computational Biology*, vol. 11, no. 8, pp. 1–21, 2015.

[152] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[153] M. W. Oliphant, "Formal approaches to innate and learned communication: Laying the foundation for language," PhD thesis, University of California, San Diego, 1997.

[154] M. Oliphant and J. Batali, "Learning and the emergence of coordinated communication," *Center for research on language newsletter*, vol. 11, no. 1, pp. 1–46, 1997.

[155] J. W. Bradbury, S. L. Vehrencamp *et al.*, *Principles of animal communication*. Sinauer Associates Sunderland, MA, 1998, vol. 132.

[156] Y. U. Cao, A. S. Fukunaga, and A. Kahng, "Cooperative mobile robotics: Antecedents and directions," *Autonomous Robots*, vol. 4, no. 1, pp. 7–27, 1997.

[157] J. Gomes, P. Urbano, and A. L. Christensen, "Introducing Novelty Search in Evolutionary Swarm Robotics," in *Swarm Intelligence*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2012, pp. 85–96.

[158] M. H. M. Alkilabi, A. Narayan, and E. Tuci, "Cooperative object transport with a swarm of e-puck robots: robustness and scalability of evolved collective strategies," *Swarm Intelligence*, vol. 11, no. 3, pp. 185–209, Dec. 2017.

[159] I. Wagner, M. Lindenbaum, and A. Bruckstein, "Distributed covering by ant-robots using evaporating traces," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 5, pp. 918–933, 1999.

[160] M. H. Christiansen and S. Kirby, *Language evolution*. OUP Oxford, 2003.

[161] Christiansen, Morten H and Kirby, Simon, "Language evolution: Consensus and controversies," *Trends in cognitive sciences*, vol. 7, no. 7, pp. 300–307, 2003.

[162] A. Cangelosi and D. Parisi, *Computer Simulation: A New Scientific Approach to the Study of Language Evolution*. London: Springer London, 2002, pp. 3–28.

[163] M. Mirolli and D. Parisi, "How producer biases can favor the evolution of communication: An analysis of evolutionary dynamics," *Adaptive Behavior*, vol. 16, no. 1, pp. 27–52, 2008.

[164] S. Nolfi, "Emergence of communication in embodied agents: co-adapting communicative and non-communicative behaviours," *Connection Science*, vol. 17, no. 3-4, pp. 231–248, 2005.

[165] J. M. Smith, *The theory of evolution*. Cambridge University Press, 1993.

[166] J. L. Scott, A. Y. Kawahara, J. H. Skevington, S.-H. Yen, A. Sami, M. L. Smith, and J. E. Yack, "The evolutionary origins of ritualized acoustic signals in caterpillars," *Nature communications*, vol. 1, no. 1, p. 4, 2010.

[167] H. Yan and J. Liebig, "Genetic basis of chemical communication in eusocial insects," *Genes & development*, vol. 35, no. 7-8, pp. 470–482, 2021.

[168] J. C. Nieh, "The evolution of honey bee communication: Learning from asian species," *Formos. Entomol*, vol. 31, pp. 1–14, 2011.

[169] D. H. Ackley and M. L. Littman, "Altruism in the evolution of communication," in *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*. The MIT Press, 1994.

[170] M. Mirolli and D. Parisi, "How can we explain the emergence of a language that benefits the hearer but not the speaker?" *Connection Science*, vol. 17, no. 3-4, pp. 307–324, 2005.

[171] T. C. Scott-Phillips and S. Kirby, "Language evolution in the laboratory," *Trends in cognitive sciences*, vol. 14, no. 9, pp. 411–417, 2010.

[172] T. J. Ord and E. P. Martins, "The evolution of behavior: phylogeny and the origin of present-day diversity," *Evolutionary behavioral ecology*, pp. 108–128, 2010.

[173] K. Støy, "Using situated communication in distributed autonomous mobile robotics," in *Proceedings of the Seventh Scandinavian Conference on Artificial Intelligence*, ser. SCAI '01. NLD: IOS Press, 2001, p. 44–52.

[174] G. E. Hinton, S. J. Nowlan *et al.*, "How learning can guide evolution," *Complex systems*, vol. 1, no. 3, pp. 495–502, 1987.

[175] H. A. Simon, "A mechanism for social selection and successful altruism," *Science*, vol. 250, no. 4988, pp. 1665–1668, 1990.

[176] S. Wischmann, D. Floreano, and L. Keller, "Historical contingency affects signaling strategies and competitive abilities in evolving populations of simulated robots," *Pro-*

ceedings of the National Academy of Sciences, vol. 109, no. 3, pp. 864–868, 2012.

[177] J. M. Smith and D. Harper, *Animal signals*. Oxford University Press, 2003.

[178] T. C. Scott-Phillips, R. A. Blythe, A. Gardner, and S. A. West, "How do communication systems emerge?" *Proceedings of the Royal Society B: Biological Sciences*, vol. 279, no. 1735, pp. 1943–1949, 2012.

[179] N. Chomsky, *Aspects of the Theory of Syntax*. MIT press, 1965.

[180] ——, *Reflections on language*. London : Fontana, 1975.

[181] J. A. Fodor *et al.*, *The language of thought*. Harvard university press Cambridge, MA, 1975, vol. 5.

[182] B. Skinner, *Verbal Behavior*. NY: Prentice Hall, 1957.

[183] N. Chomsky, "A review of bf skinner's verbal behavior," *The Language and Thought Series*, pp. 48–64, 1980.

[184] S. Kirby and J. R. Hurford, *The Emergence of Linguistic Structure: An Overview of the Iterated Learning Model*. London: Springer London, 2002, pp. 121–147.

[185] S. Kirby, M. Dowman, and T. L. Griffiths, "Innateness and culture in the evolution of language," *Proceedings of the National Academy of Sciences*, vol. 104, no. 12, pp. 5241–5245, 2007.

[186] A. Newell, "Physical symbol systems," *Cognitive Science*, vol. 4, no. 2, pp. 135–183, 1980.

[187] L. Steels, "The symbol grounding problem has been solved. so what's next," *Symbols and embodiment: Debates on meaning and cognition*, pp. 223–244, 2008.

[188] G. Lakoff, *Women, Fire, and Dangerous Things: What Categories Reveal About the Mind*. University of Chicago Press, 1987.

[189] R. Pfeifer and C. Scheier, *Understanding intelligence*. Cambridge, MA: MIT press, 1999.

[190] R. A. Brooks, "Elephants don't play chess," *Robotics and Autonomous Systems*, vol. 6, no. 1, pp. 3–15, 1990, designing Autonomous Agents.

[191] C. S. Peirce, *Collected papers. Volume I-VIII*. Harvard University Press. (The volumes were published from 1931 to 1958)., 1931.

[192] L. Steels and F. Kaplan, "Situated grounded word semantics," in *Proceedings of IJCAI 99. Morgan Kaufmann.*, 1999.

[193] F. De Saussure, "Course in general linguistics," *Literary theory: An anthology*, vol. 2, pp. 59–71, 2004.

[194] L. Steels, "Perceptually grounded meaning creation," in *Proceedings of the International Conference on Multiagent Systems (ICMAS-96)*, 1996, pp. 338–44.

[195] P. Vogt, "The emergence of compositional structures in perceptually grounded language

games," *Artificial Intelligence*, vol. 167, no. 1, pp. 206–242, 2005, connecting Language to the World.

[196] L. Steels, *The Talking Heads experiment: Origins of words and meanings*. Language Science Press, 2015, vol. 1.

[197] P. Gardenfors, *Conceptual spaces: The geometry of thought*. MIT press, 2004.

[198] ——, *The geometry of meaning: Semantics based on conceptual spaces*. MIT press, 2014.

[199] A. Puglisi, A. Baronchelli, and V. Loreto, "Cultural route to the emergence of linguistic categories," *Proceedings of the National Academy of Sciences*, vol. 105, no. 23, pp. 7936–7940, 2008.

[200] A. Baronchelli, T. Gong, A. Puglisi, and V. Loreto, "Modeling the emergence of universality in color naming patterns," *Proceedings of the National Academy of Sciences*, vol. 107, no. 6, pp. 2403–2407, 2010.

[201] T. Riga, A. Cangelosi, and A. Greco, "Symbol grounding transfer with hybrid self-organizing/supervised neural networks," in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, vol. 4, 2004, pp. 2865–2869 vol.4.

[202] S. D. Levy and S. Kirby, "Evolving distributed representations for language with self-organizing maps," in *Symbol Grounding and Beyond*, P. Vogt, Y. Sugita, E. Tuci, and C. Nehaniv, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 57–71.

[203] S. M. de Paula and R. R. Gudwin, "Evolving conceptual spaces for symbol grounding in language games," *Biologically Inspired Cognitive Architectures*, vol. 14, pp. 73–85, 2015.

[204] N. Rasheed, S. H. Amin, U. Sultana, and A. R. Bhatti, "Hopfield net spreading activation for grounding of abstract action words in cognitive robot," *Biologically Inspired Cognitive Architectures*, vol. 21, pp. 37–46, 2017.

[205] R. Schulz, G. Wyeth, and J. Wiles, "Lingodroids: socially grounding place names in privately grounded cognitive maps," *Adaptive Behavior*, vol. 19, no. 6, pp. 409–424, 2011.

[206] A. Cangelosi, "The grounding and sharing of symbols," *Pragmatics &; Cognition*, vol. 14, no. 2, pp. 275–285, 2006.

[207] L. Wittgenstein, *Philosophical Investigations*, G. E. M. Anscombe, Ed. New York, NY, USA: Wiley-Blackwell, 1953.

[208] L. Steels, "A Self-Organizing Spatial Vocabulary," *Artificial Life*, vol. 2, no. 3, pp. 319–332, 1995.

[209] L. Steels and F. Kaplan, "Spontaneous lexicon change," in *Annual Meeting of the Association for Computational Linguistics*, 1998.

[210] A. Baronchelli, M. Felici, V. Loreto, E. Caglioti, and L. Steels, "Sharp transition towards shared vocabularies in multi-agent systems," *Journal of Statistical Mechanics: Theory*

and Experiment, vol. 2006, no. 06, p. P06014, 2006.

[211] L. Steels, M. Loetzsch *et al.*, "The grounded naming game," *Experiments in cultural language evolution*, vol. 3, pp. 41–59, 2012.

[212] P. Wellens, *Adaptive Strategies in the Emergence of Lexical Systems*. VUBPress, 2012.

[213] R. Miletitch, A. Reina, M. Dorigo, and V. Trianni, "Emergent naming conventions in a foraging robot swarm," *Swarm Intelligence*, vol. 16, no. 3, pp. 211–232, 2022.

[214] S. Kirby, "Spontaneous evolution of linguistic structure-an iterated learning model of the emergence of regularity and irregularity," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 2, pp. 102–110, 2001.

[215] H. Brighton, "Compositional Syntax From Cultural Transmission," *Artificial Life*, vol. 8, no. 1, pp. 25–54, 01 2002.

[216] A. Cangelosi, "Modeling the evolution of communication: From stimulus associations to grounded symbolic associations," in *Advances in Artificial Life*, D. Floreano, J.-D. Nicoud, and F. Mondada, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 654–663.

[217] H. Ebara, T. Nakamura, A. Taniguchi, and T. Taniguchi, "Multi-agent reinforcement learning with emergent communication using discrete and indifferentiable message," in *2023 15th International Congress on Advanced Applied Informatics Winter (IIAI-AAI-Winter)*, 2023, pp. 366–371.

[218] P. Van Eecke, K. Beuls, J. Botoko Ekila, and R. Rădulescu, "Language games meet multi-agent reinforcement learning: A case study for the naming game," *Journal of Language Evolution*, vol. 7, no. 2, pp. 213–223, 2023.

[219] L. Steels *et al.*, "Synthesising the origins of language and meaning using co-evolution, self-organisation and level formation," *Evolution of Human Language. Edinburgh University Press, Edinburgh*, 1997.

[220] J. Pollack, M. A. Bedau, P. Husbands, R. A. Watson, and T. Ikegami, *A Computational Framework to Simulate the Coevolution of Language and Social Structure*, 2004, pp. 158–163.

[221] P. Vogt, *How mobile robots can self-organise a vocabulary*. Language Science Press, 2015.

[222] T. Belpaeme, J. Van Looveren, and L. Steels, "The construction and acquisition of visual categories," in *Learning Robots*, A. Birk and J. Demiris, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 1–12.

[223] A. Cangelosi, "Grounding language in action and perception: From cognitive agents to humanoid robots," *Physics of Life Reviews*, vol. 7, no. 2, pp. 139–151, 2010.

[224] S. Harnad, S. Hanson, and J. Lubin, "Categorical perception and the evolution of supervised learning in neural nets," in *Working papers of the AAAI Spring Symposium on Machine Learning of Natural Language and Ontology*, D. W. Powers and L. Reeker,

Eds., 1991, pp. 65–74.

[225] A. G. Angelo Cangelosi and S. Harnad, "From robotic toil to symbolic theft: Grounding transfer from entry-level to higher-level categories1," *Connection Science*, vol. 12, no. 2, pp. 143–162, 2000.

[226] A. Cangelosi and S. Harnad, "The adaptive advantage of symbolic theft over sensorimotor toil: Grounding language in perceptual categories," *Evolution of Communication*, vol. 4, no. 1, pp. 117–142, 2001.

[227] A. Greco, T. Riga, and A. Cangelosi, "The acquisition of new categories through grounded symbols: An extended connectionist model," in *Artificial Neural Networks and Neural Information Processing — ICANN/ICONIP 2003*, O. Kaynak, E. Alpaydin, E. Oja, and L. Xu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 763–770.

[228] A. Cangelosi, "Symbol grounding in connectionist and adaptive agent models," in *New Computational Paradigms*, S. B. Cooper, B. Löwe, and L. Torenvliet, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 69–74.

[229] M. A. Nowac, J. B. Plotkin, and V. A. A. Jansen, "The evolution of syntactic communication," *Nature*, vol. 404, no. 6777, p. 495–498, 2000.

[230] K. Zuberbühler, "A syntactic rule in forest monkey communication," *Animal Behaviour*, vol. 63, no. 2, pp. 293–299, 2002.

[231] T. C. Scott-Phillips and R. A. Blythe, "Why is combinatorial communication rare in the natural world, and why is language an exception to this trend?" *Journal of the Royal Society Interface*, vol. 10, no. 88, p. 20130520, 2013.

[232] J. F. Fontanari and L. I. Perlovsky, "Evolving compositionality in evolutionary language games," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 758–769, 2007.

[233] R. Cann, *Formal semantics: an introduction.* Cambridge University Press, 1993.

[234] S. Kirby, *Syntax Without Natural Selection: How Compositionality Emerges from Vocabulary in a Population of Learners.* Cambridge University Press, 2000, p. 303–323.

[235] P. Vogt, "Variation, competition and selection in the self-organisation of compositionality," *The mind, the body and the world: Psychology after cognitivism*, pp. 233–256, 2007.

[236] P. Vogt, "Iterated learning and grounding: from holistic to compositional languages," in *Language Evolution and Computation, 15th European Summer School in Logic, Language and Information.* Technical University Vienna, 2003, pp. 76–86.

[237] P. Vogt, "Stability conditions in the evolution of compositional languages: issues in scaling population sizes," in *Proceedings of the European Conference on Complex Systems*, P. Bourgine, F. Kepes, and M. Schoenauer, Eds., 2005.

[238] P. Vogt, "Overextensions and the emergence of compositionality," in *The Evolution of*

Language: Proceedings of the 6th international conference on the evolution of language. World Scientific Press, 2006, pp. 364–371.

[239] P. Vogt, "Language evolution and robotics: issues on symbol grounding and language acquisition," in *Artificial cognition systems*. IGI Global, 2007, pp. 176–209.

[240] N. Cambier, V. Frémont, and E. Ferrante, "Group-size regulation in self-organised aggregation through the naming game," in *International Symposium on Swarm Behavior and Bio-Inspired Robotics (SWARM 2017)*, 2017.

[241] N. Cambier, V. Frémont, V. Trianni, and E. Ferrante, "Embodied evolution of self-organised aggregation by cultural propagation," in *Swarm Intelligence*, M. Dorigo, M. Birattari, C. Blum, A. Christensen, Anders L.and Reina, and V. Trianni, Eds. Cham: Springer International Publishing, 2018, pp. 351–359.

[242] E. H., "Alpha shapes - a survey," *Tessellations in the Sciences*, 2010.

[243] P. Zahadat, S. Hahshold, R. Thenius, K. Crailsheim, and T. Schmickl, "From honeybees to robots and back: division of labour based on partitioning social inhibition," *Bioinspiration & biomimetics*, vol. 10, no. 6, p. 066005, 2015.

[244] P. Zahadat and T. Schmickl, "Division of labor in a swarm of autonomous underwater robots by improved partitioning social inhibition," *Adaptive Behavior*, vol. 24, no. 2, pp. 87–101, 2016.

[245] F. L. Lewis, H. Zhang, K. Hengster-Movric, and A. Das, *Cooperative Control of Multi-Agent Systems: Optimal and Adaptive Design Approaches*, ser. Communications and Control Engineering. London: Springer, 2014.

[246] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[247] R. D. Beer and J. C. Gallagher, "Evolving dynamical neural networks for adaptive behavior," *Adaptive Behavior*, vol. 1, no. 1, pp. 91–122, 1992.

[248] M. Duarte, A. L. Christensen, and S. Oliveira, "Towards Artificial Evolution of Complex Behaviors Observed in Insect Colonies," in *Progress in Artificial Intelligence*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2011, pp. 153–167.

[249] R. Chang and S. Worlanyo, "Evolving swarm communication with neat," *Dept. Comp. Sci., Swarthmore College, Swarthmore, PA*, 2015.

[250] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2021.

[251] G. Gao, Y. Mei, Y.-H. Jia, W. N. Browne, and B. Xin, "Adaptive coordination ant colony optimization for multipoint dynamic aggregation," *IEEE Transactions on Cybernetics*, vol. 52, no. 8, pp. 7362–7376, 2022.

[252] G. Gao, Y. Mei, B. Xin, Y.-H. Jia, and W. N. Browne, "Automated coordination strategy design using genetic programming for dynamic multipoint dynamic aggregation," *IEEE*

Transactions on Cybernetics, vol. 52, no. 12, pp. 13 521–13 535, 2022.

[253] J. Wu, C. Luo, Y. Luo, and K. Li, "Distributed uav swarm formation and collision avoidance strategies over fixed and switching topologies," *IEEE Transactions on Cybernetics*, vol. 52, no. 10, pp. 10 969–10 979, 2022.

[254] A. Bono, G. Fedele, and G. Franzè, "A swarm-based distributed model predictive control scheme for autonomous vehicle formations in uncertain environments," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 8876–8886, 2022.

[255] D. O. Hebb, *The organization of behavior.* New York: Wiley, 1949.

[256] P. Jaccard, "Étude comparative de la distribution florale dans une portion des alpes et des jura," *Bull Soc Vaudoise Sci Nat*, vol. 37, pp. 547–579, 1901.

[257] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[258] L. Steels, M. Loetzsch, and M. Spranger, "Semiotic dynamics solves the symbol grounding problem," *Nature Proceedings*, vol. 1, no. 1, 2007.

[259] J. R. Searle, "Minds, brains, and programs," *Behavioral and brain sciences*, vol. 3, no. 3, pp. 417–457, 1980.

[260] R. Sendra-Arranz and Á. Gutiérrez, "Emergence of communication through artificial evolution in an orientation consensus task in swarm robotics," in *Artificial Intelligence Applications and Innovations*, I. Maglogiannis, L. Iliadis, J. MacIntyre, and M. Dominguez, Eds. Cham: Springer Nature Switzerland, 2023, pp. 515–526.

[261] ——, "Emergence of flocking behaviors transferring previously evolved alignment robot controllers," *Evolving Systems*, vol. 16, no. 1, p. 28, 2025.

[262] ——, "Evolution of robot controllers for solving multiple tasks sequentially," in *XLIV Jornadas de Automática*, 2023, pp. 738–743.

[263] G. B. Ermentrout and D. H. Terman, *Mathematical foundations of neuroscience*. Springer New York, NY, 2010, vol. 35.

[264] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning.* USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[265] L. J. Eshelman and J. D. Schaffer, "Real-coded genetic algorithms and interval-schemata," in *Foundations of Genetic Algorithms*, ser. Foundations of Genetic Algorithms, L. D. WHITLEY, Ed. Elsevier, 1993, vol. 2, pp. 187–202.

Appendices

A Evolutionary computation

A.1 Genetic algorithm

A Genetic Algorithm (GA) [264] is a population-based algorithm that, inspired by biological evolution, mimics how natural selection and survival of the fittest operate, in order to address optimization problems. The GA was originally conceived as a binary-coded algorithm, so that the genes that compose a genotype encode the parameters being optimized as binary numbers. In contrast, in a real-coded GA, the genes are real numbers that are generally constrained to some search space. Real-coded GAs are a more suitable option in ER, when the parameters being evolved are the free parameters of hyperparameters of an ANN. Such is the case of this PhD Thesis, in which only real-coded GAs are considered.

The most basic form of a GA works as follows (see Figure A.1):

(1) **Initialization**: Candidate solutions are collected in a population of individuals or

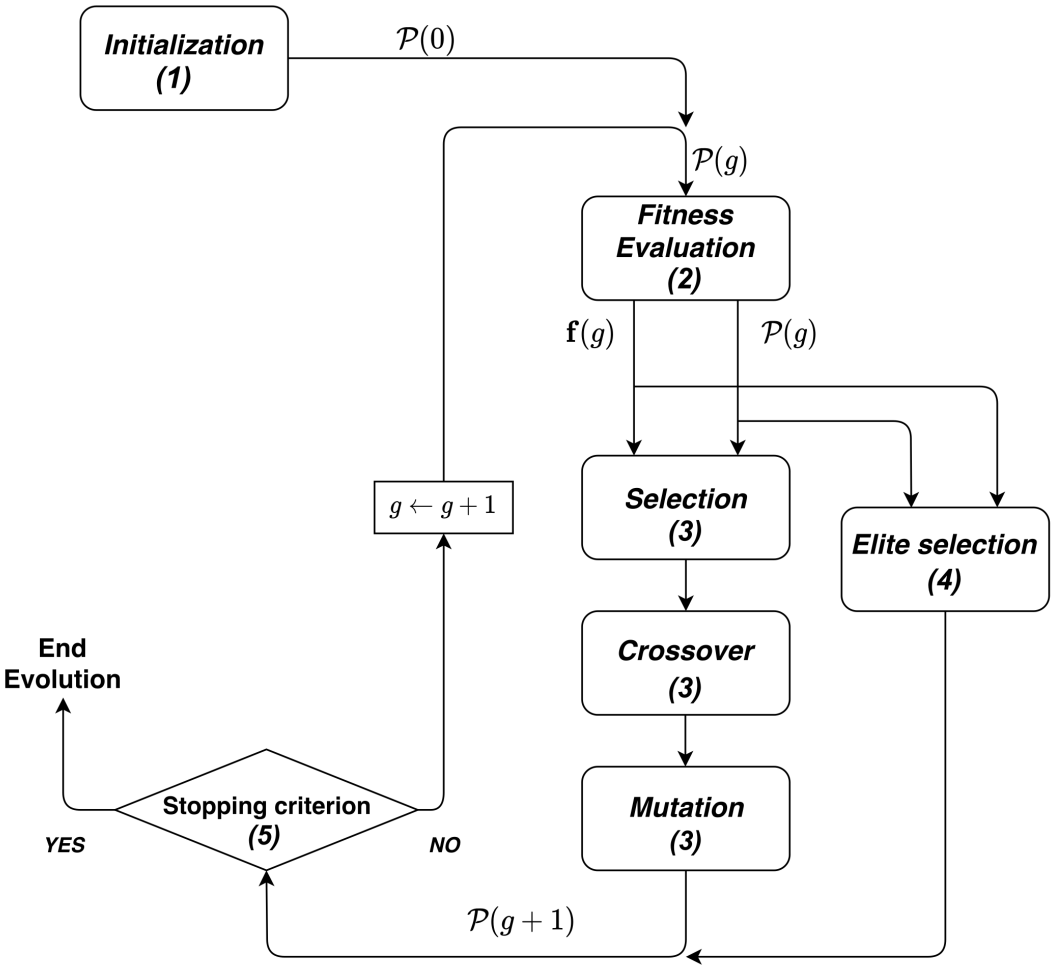


Figure A.1: General diagram of a genetic algorithm. The variable g is the generation counter, and $\mathcal{P}(g)$ is the population at a specific generation g.

genotypes. Such population is iteratively updated until a suitable solution is reached.

(2) **Evaluation**: Each individual in the population is evaluated, and a fitness score that quantifies its performance is computed. Under the frame of ER, the evaluation process typically requires the execution of the corresponding robotics simulations. Moreover, the computation of the final fitness score of an individual is refered as fitness function, and it has to be designed ad hoc for each specific task being addressed.

(3) **Operation**: The evaluated individuals are subject to a series of genetic operators that, ultimately, give rise to the next population. The main genetic operators of a basic GA are the following:

- *Selection operator*: selects the subset of individuals in the population that will be used to produce the next generation. The selection process is driven by the fitness scores of the genotypes, so that the fittest individuals normally have the highest probability of being selected. Some popular selection operators are the *truncation selection*, the *rank selection*, and, the *roulette-wheel selection*.

- *Crossover operator*: combines pairs of genotypes (parents) to create new candidate solutions (offspring). The resulting genotypes inherit the traits of both parents as a combination of their genes. Some popular crossover operators are the *one-point*, *multi-point*, *uniform* and *BLX-*α crossovers.

- *Mutation operator*: slightly alters the genes of an individual with the aim of exploring new areas of the search space. The mutation operator is applied with a very low probability to the genes of each genotype. The most common mutation operator in real-coded GAs is the *Gaussian mutation*, that adds some Gaussian noise to those genes that are subject to the mutation.

(4) **Elitism**: In addition to the new individuals resulting from the application of genetic operators, the very best-performing individuals of the previous generation are directly included in the next population. It is called *elite selection*, and it is purely based on the fitness scores.

(5) **Finalization**: A stopping criterion is established and assessed at the end of every generation. The artificial evolution concludes in the case that such stopping criterion is met. Some examples of stopping criteria are: (i) a maximum number of generations is elapsed, (ii) a sufficiently good population fitness value (either average or maximum fitness) is reached, or (iii) the average fitness score of the population becomes stagnant or decreases for several generations.

Figure A.1 illustrates the data flow of a GA, highlighting the main steps accomplished in each generation g.

Selection operator

The selection operator that is used in the GA implementations of this PhD Thesis is the tournament selection. The tournament selection procedure is shown in Algorithm 3, and its outcome is the set of selected genotypes, $\mathcal{P}_{sel}$. This set is initially empty and, at the end

Algorithm 3 Tournament selection

$\mathcal{P} \leftarrow \{\mathbf{g}_1, \ldots, \mathbf{g}_P\}$ ▷ Population of size P ($\mathbf{g}_k$ is the k-th genotype).
$\mathcal{P}_{sel} \leftarrow \{\}$ ▷ Selected genotypes after the tournament selection (initially empty).
for $i = \{1, \ldots, \mu\}$ **do**
    $T_i \leftarrow \{\}$ ▷ Set of genotypes of the i-th tournament (initially empty)
    for $j = \{1, \ldots, K\}$ **do**
        Select randomly a genotype $\mathbf{g}_j$ from $\mathcal{P}$.
        Add $\mathbf{g}_j$ to T_i.
    end for
    Winner of the tournament, $\mathbf{g}_{win}$, as the genotype in T_i with highest fitness score.
    Add $\mathbf{g}_{win}$ to $\mathcal{P}_{sel}$.
end for

of the selection process is composed by μ individuals. In addition, let, $\mathcal{P}$ be the current population, of size P, that is subject to this selection operator. Moreover, each genotype in $\mathcal{P}$ is denoted as $\mathbf{g}_k$, and, in real-coded GAs, it is generally a real vector encoding the real-valued parameters. In the tournament selection, μ tournaments are iteratively accomplished with the aim of selecting the resulting μ individuals. Each tournament is independent from each other, and it is composed by K genotypes, that are randomly selected from the population (with same probabilities). Among the K tournament members, the winner is the one with highest fitness score, and it is directly included to the selection set $\mathcal{P}_{sel}$.

Crossover operator

BLX-α is the crossover operator that is considered in the implementations of the GA in this PhD Thesis. BLX-α [265] is a popular crossover technique in real-coded GAs, as it reduces the premature convergence through a random sampling of new offspring in the neighborhood of the parent genes. In order to describe the crossover mechanics, let $\mathbf{g}_a$ and $\mathbf{g}_b$ be the genotypes that are used as parent in the crossover transformation. Moreover, let $\mathbf{g}_a^{'}$ and $\mathbf{g}_b^{'}$ be the offspring genotypes that result from such recombination. All the individuals $\mathbf{g}_a$, $\mathbf{g}_b$, $\mathbf{g}_a^{'}$, and $\mathbf{g}_b^{'}$ have the same number of genes, denoted as d, and the k-th gene within some genotype $\mathbf{g}$ is defined as $\mathbf{g}(k)$.

With this notation in mind, Algorithm 4 defines the main steps that are accomplished during the recombination of two parent genotypes, $\mathbf{g}_a$ and $\mathbf{g}_b$, using the BLX-α. In addition to $\mathbf{g}_a$ and $\mathbf{g}_b$, it receives the parameter α, which establishes the trade-off between exploration and exploitation. A value of $\alpha = 0$ implies that the recombination is fully exploitaive, whereas larger values of α imply a greater degree of exploration. In [265], the authors proposed a value of $\alpha = 0.5$, which is also used in the experiments of this PhD Thesis. BLX-α randomly samples the new genes of the offspring, $\mathbf{g}_a^{'}(k)$ and $\mathbf{g}_b^{'}(k)$, from a uniform distribution $\mathcal{U}(A_k, B_k)$ in the segment $[A_k, B_k]$. This sampling segment is computed as in Algorithm 4, and it is also illustrated in Figure A.2. It depicts the sampling segment divided into two types of zones:

- *Exploitation zone* $\rightarrow [g_{min,k},\ g_{max,k}]$
- *Exploration zone* $\rightarrow [A_k,\ g_{min,k}) \cup (g_{max,k},\ B_k]$

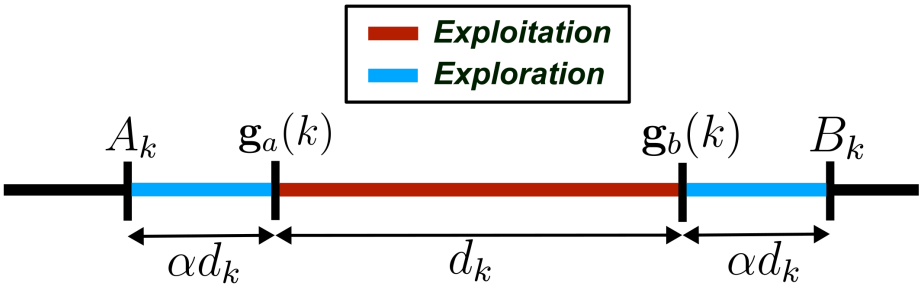


Figure A.2: Illustration of the sampling segments of new genes using the crossover operator BLX-α. It represents the crossover process using the k-th gene of two parent genotypes, $\mathbf{g}_a(k)$ and $\mathbf{g}_b(k)$. The sampling segment is split into an exploration zone, in blue, and an exploitation segment, in red.

Mutation operator

In this PhD Thesis the Gaussian mutation is considered as the main gene mutation operator. With some low probability p_{mut}, the Gaussian mutation adds some normally distributed noise to the gene subject to the mutation. For every gene $\mathbf{g}(k)$, the mutation mechanism works as in Equation 1,

$$\mathbf{g}_{mut}(k) = \mathcal{N}(\mathbf{g}(k), \sigma^2_{mut}), \tag{1}$$

where $\mathcal{N}(\mathbf{g}(k), \sigma^2_{mut})$ is the normal distribution centered at the gene value before the mutation and with a mutation power σ_{mut}. Additionally, $\mathbf{g}_{mut}(k)$ is the k-th gene after the mutation.

A.2 Neuroevolution of augmenting topologies

In addition to GA, NeuroEvolution of Augmenting Topologies (NEAT) [83] is also used in some experiments of this PhD Thesis (specifically in Chapter 5). NEAT algorithm is a population-based neuroevolutionary algorithm that is widely used to automatically design robot controllers in ER because it evolves not only the ANN parameters but also its neural topology itself. In NEAT, a population ($\mathcal{P}$) of P candidate solutions (genotypes) is iteratively updated and optimized. The goal of such evolution is to maximize the fitness scores achieved

Algorithm 4 BLX-α

procedure BLX-α($\mathbf{g}_a$, $\mathbf{g}_b$, α)
  for $k = \{1, \ldots, d\}$ **do** ▷ For every gene.
    $g_{min,k} \leftarrow \min\{\mathbf{g}_a(k), \mathbf{g}_b(k)\}$
    $g_{max,k} \leftarrow \max\{\mathbf{g}_a(k), \mathbf{g}_b(k)\}$
    $d_k \leftarrow |\mathbf{g}_a(k) - \mathbf{g}_b(k)|$
    $A_k \leftarrow g_{min,k} - \alpha d_k$
    $B_k \leftarrow g_{max,k} + \alpha d_k$
    $\mathbf{g}'_a \leftarrow \mathcal{U}(A_k, B_k)$
    $\mathbf{g}'_b \leftarrow \mathcal{U}(A_k, B_k)$
  end for
end procedure

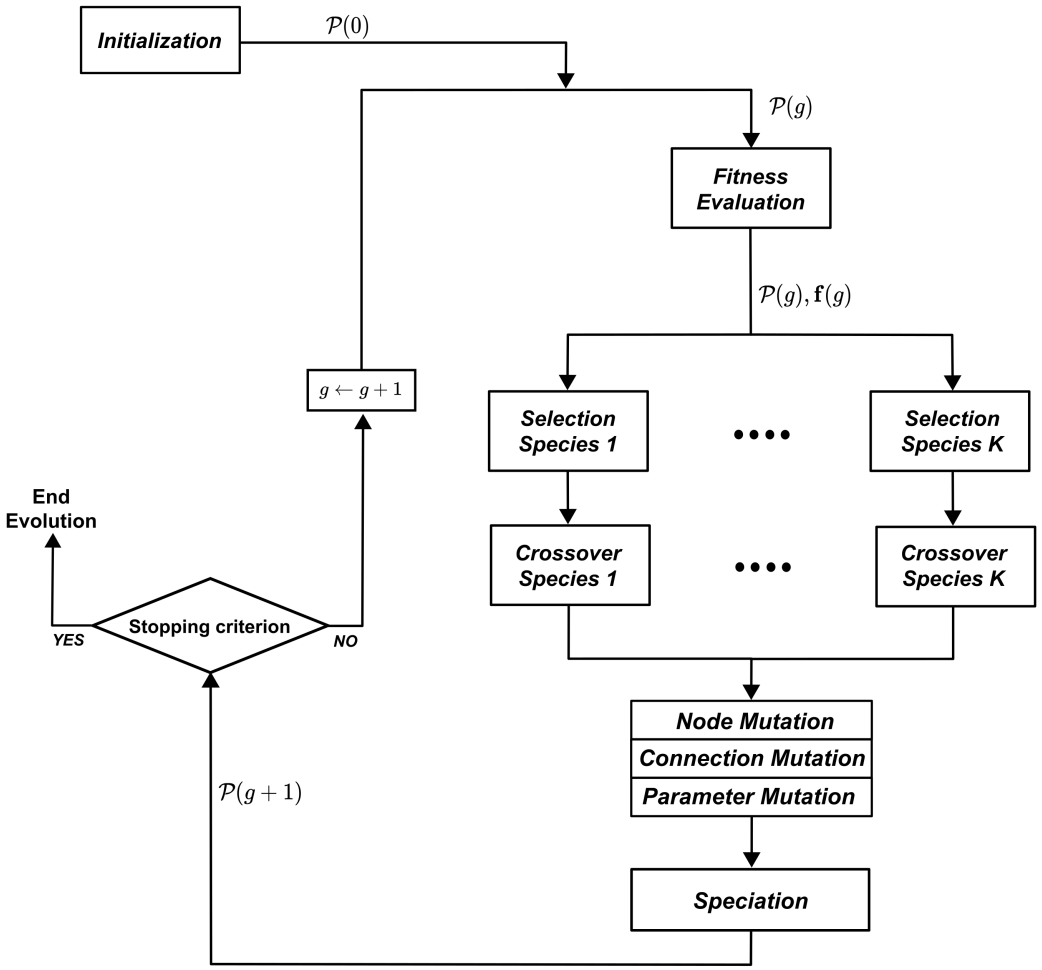


Figure A.3: General diagram of a NEAT algorithm. The variable g is the generation counter, and $\mathcal{P}(g)$ is the population at a specific generation g.

by the NEAT's individuals in the task being addressed. In contrast to the GA, NEAT genotypes are no longer real-valued vectors. Instead, the genes of an individual encode within the same structure both the numeric parameters and the neural architecture (neurons and synapses) of the phenotype ANN. Consequently, two types of genes can be distinguished in NEAT genotypes:

- *Node genes*: encode information from the neurons of the ANN. A single node gene can carry information about multiple parameters of a single neuron (e.g. membrane time constants, neuron's bias, and activation function). Additionally, it can also include other relevant information, such as the type of neuron or its associated layer.
- *Connection genes*: encode information from the synapses of the ANN. It contains the numeric weight parameter of the corresponding synapse, the pre-synptic neuron, and the post-synaptic neuron.

Genotypes are initialized enconding a very simple ANN that directly connects the input nodes with the output neurons. The artificial evolution is the process responsible for designing

more complex and elaborate ANN topologies, if the goal task requires it. In contrast to the topology initialization, all the ANN parameters subject to the evolution process are initialized randomly. In each generation g, genotypes are evaluated in order to obtain individual-wise fitness scores. Node and connection genes are subject to selection, crossover, and mutation operators. Moreover, NEAT algorithm implements *speciation* techniques, so that the population is divided into niches to protect innovations in the ANN topologies. Figure A.3 shows the diagram and overall data flow of the NEAT algorithm. Note that the selection and crossover operations are applied separately for each species. Additionally, at the end of each generation, a speciation step is performed to reassign each individual to the most appropriate niche, and to create new niches if necessary.

Mutation operator

Three different types of mutations can be distinguished in NEAT. One of them is a parameter mutation, that simply alters the value of some parameter of the ANN. Conversely, two of them are structural mutations that result in a growth of the ANN topology (see Figure A.4). The functing of these complementary mutations is the following:

- *Connection mutation*: a new connection is added with a low probability to the ANN architecture encoded in each genotype. The pre-synaptic and post-synaptic neurons of the new connection are selected randomly among the existing neurons of the ANN.

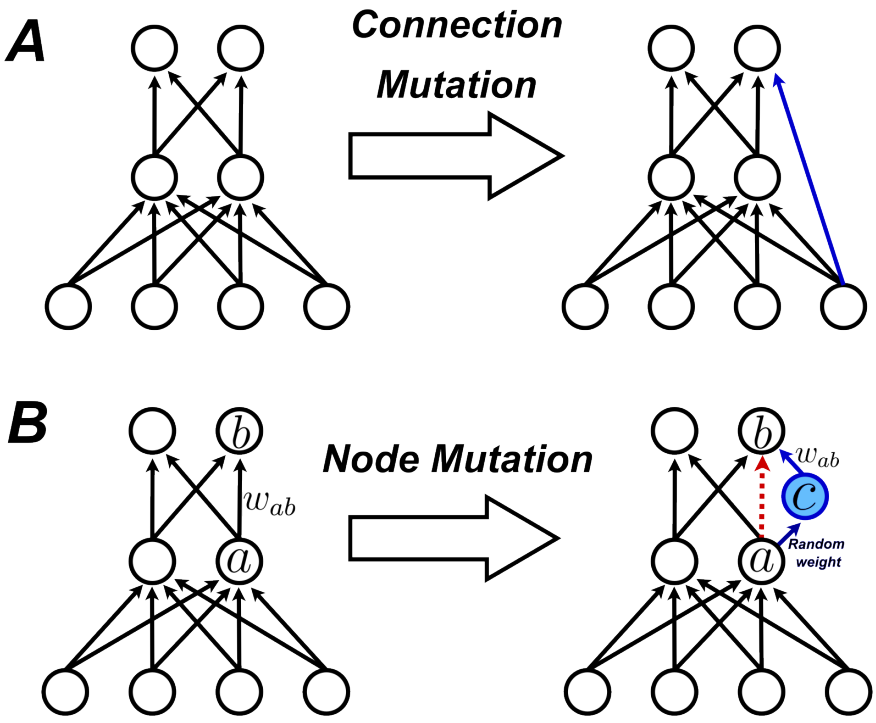


Figure A.4: Example of the NEAT mutation operator. (**A**) Mutation of a connection gene. (**B**) Mutation of a node gene. The parts of the ANN topology that were added during the mutation are highlighted in blue, and the parts that were disabled because of the mutation are colored in red.

However, the candidate neurons should fulfill the following conditions: (i) there should not be an existing connection between them and (ii) the post-synaptic node cannot be an input node. The weight of the new synapse is initialized randomly using a normal distribution with an standard deviation of 0.1. An illustrative example of the connection mutation process is shown in Figure A.4A.

- *Node mutation*: new nodes can also be attached to the evolving ANN topologies with a low probability (generally different from the connection mutation probability). The node mutation works as follows: firstly, a synapse is randomly selected among the already existing ones. To ease the explanation, let a and b be the pre-synaptic and post-synaptic nodes of the selected synapse, respectively. Moreover, let w_{ab} be the weight of such synapse. Then, the selected synapse joining a and b is disabled and, instead, it is replaced by (1) a new neuron c, (2) a synapse joining a and c, and (3) a synapse joining c and b. The parameters of the new node c are initialized randomly. Similarly, the weight of the synapse a-c is also sampled randomly using a normal distribution centered at zero. However, the weight of the synapse joining c and b is the same as the weight of the disabled synapse, w_{ab}. Figure A.4B illustrates an example of a node mutation. It highlights in red the former connection that is disabled and in blue the new node and connections resulting from the mutation.
- *Parameter mutation*: apart from adding new nodes and connections, parameters of existing neurons and synapses can be also subject to mutations. Their functioning is very similar to the Gaussian mutation operator of a GA. Essentially, the value of a parameter is modified, with a low probability, by adding a normally distributed noise to it.

Competing conventions problem and innovation numbers

A complex problem that emerges in NEAT, and in other evolutionary algorithms that evolve the topology of an ANN is the *competing conventions problem*. The *competing conventions problem* essentially states that the same phenotype structure (i.e. an ANN) can be encoded by multiple genotypes. NEAT solves this issue through historical markings associated to the genes, called *innovation numbers*. An innovation number is attached to every gene once it is created, either at the beginning of the artificial evolution or through a mutation. The innovation number serves as a unique identifier of a gene, so that any two genes with the same innovation number express the exact same trait. Moreover, the lower the innovation number, the more time has passed since the gene was originally created. Innovation numbers only register topological traits and, thus, node and connection parameters are not tracked throughout the evolution history.

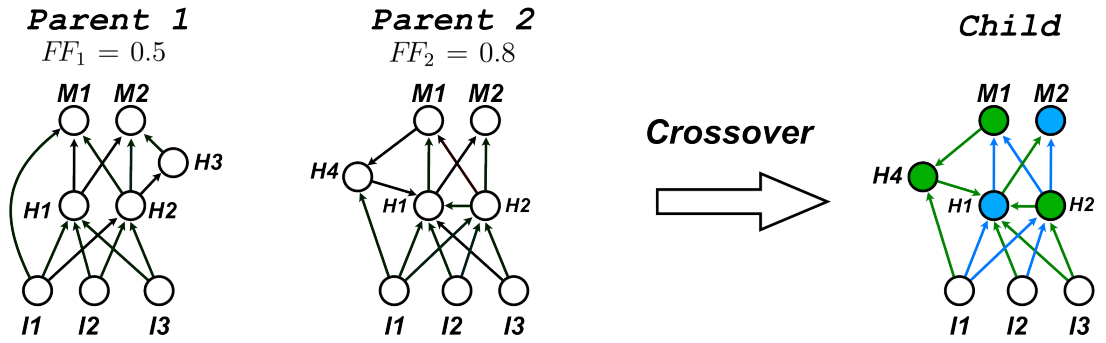
Crossover operator

Performing a recombination of two genotypes that encode structurally different ANNs is a complex task. In the NEAT algorithmm, a crossover operator that uses innovation numbers to perform efficient genotype recombinations was proposed. The authors distinguished three types of genes during the crossover process: *matching*, *disjoint*, and *excess* genes. Matching

genes are present in the genotypes of both parents involved in the crossover. Conversely, disjoint and excess genes do only exist in the genotype of one of the parents. The type of gene is easily identified using innovation numbers, so that genes with the same identifier are categorized as matching genes. Therefore, the NEAT crossover works as follows:

- The genes of both parents are identified as either matching, disjoing or excess genes.
- Offspring inherit all the matching genes from the parents. The parameters of the matching genes (i.e. weights and neuron paramters) are randomly inherited from one of the parents, with equal probability.
- Disjoint and excess genes are inherited only from the fittest parent. If such gene is inherited, the corresponding gene parameters are copied as well.
- Innovation numbers are always attached to inherited genes (matching, disjoint, and excess genes).
- Disabled connection genes can be also inherited, following the same crossover rules described above. Moreover, an inherited connection can be enabled again with some fixed probability.

A



B

P1	Innov=1 Pre=I1 Post=H1 w=0.5	Innov=2 Pre=I1 Post=H2 w = -0.2	Innov=3 Pre=I2 Post=H1 w=2	Innov=4 Pre=I2 Post=H2 w=0.15	Innov=5 Pre=I3 Post=H1 w=0.3	Innov=6 Pre=I3 Post=H2 w=2	Innov=7 Pre=H2 Post=H3 w=-0.35				Innov=11 Pre=H1 Post=M1 w=0.65	Innov=12 Pre=H1 Post=M2 w=-0.01	Innov=13 Pre=H2 Post=M1 w=-0.8	Innov=14 Pre=H2 Post=M2 w=0.12	Innov=15 Pre=H3 Post=M2 w=1.05		Innov=17 Pre=I1 Post=M1 w=3
P2	Innov=1 Pre=I1 Post=H1 w=0.2	Innov=2 Pre=I1 Post=H2 w=1	Innov=3 Pre=I2 Post=H1 w=0.1	Innov=4 Pre=I2 Post=H2 w=0.25	Innov=5 Pre=I3 Post=H1 w=-1	Innov=6 Pre=I3 Post=H2 w=3		Innov=8 Pre=H2 Post=H1 w=0.05	Innov=9 Pre=I1 Post=H4 w=0.67	Innov=10 Pre=H4 Post=H1 w=-0.2	Innov=11 Pre=H1 Post=M1 w=1.4	Innov=12 Pre=H1 Post=M2 w=-0.6	Innov=13 Pre=H2 Post=M1 w=-1.5	Innov=14 Pre=H2 Post=M2 w=1.2		Innov=16 Pre=M1 Post=H4 w=-0.85	
CHILD	Innov=1 Pre=I1 Post=H1 w=0.5	Innov=2 Pre=I1 Post=H2 w=-0.2	Innov=3 Pre=I2 Post=H1 w=0.1	Innov=4 Pre=I2 Post=H2 w=0.15	Innov=5 Pre=I3 Post=H1 w=-1	Innov=6 Pre=I3 Post=H2 w=3		Innov=8 Pre=H2 Post=H1 w=0.05	Innov=9 Pre=I1 Post=H4 w=0.67	Innov=10 Pre=H4 Post=H1 w=-0.2	Innov=11 Pre=H1 Post=M1 w=0.65	Innov=12 Pre=H1 Post=M2 w=-0.6	Innov=13 Pre=H2 Post=M1 w=-0.8	Innov=14 Pre=H2 Post=M2 w=0.12		Innov=16 Pre=M1 Post=H4 w=-0.85	

Figure A.5: Example of the NEAT crossover operator. (**A**) ANN architectures of the parents genotypes and the child resulting from the crossover. For each parent the hypothetical fitness score FF_i is displayed, showing that the fittest parent is `Parent 2`. The color of the neurons and synapses of the child ANN indicate the parent from whom the corresponding gene was inherited (blue for `Parent 1` and green for `Parent 2`). According to [83], topological genes (connections and nodes) are inherited from the parent with highest fitness score (`Parent 2` in this example). (**B**) List of connection genes of both parents and the resulting child.

Figure A.5 provides an illustrative example of the crossover process in NEAT. In Figure A.5A, the two parents' topologies and the child ANNs that result from the recombination are shown. In this example, the `Parent 2` has a higher fitness score than `Parent 1`. The connections and nodes colored in green denote that the corresponding gene was inherited from `Parent 2` (the fittest). In contrast, the color green indicates that the gene was inherited from `Parent 1`. Figure A.5B shows the connection genes of the parents and the child. Each gene shows the innovation number, the weight of the synapse, and the pre-synaptic and post-synaptic neurons. The genes are aligned based on their innovation numbers, so that matching, excess and disjoint genes are clearly identified. Moreover, note that the offspring genotype only inherits the excess and disjoint genes from `Parent 2`, because its hypothetical fitness score is greater than the one of `Parent 1`.

Speciation

A challenging issue that should be addressed in neuroevolutionary algorithms that evolve the topology of the ANN is the preservation of structural mutations. That is, when a new neuron or connection is created through some mutation, it will generally not result in an immediate fitness growth. Such topology upgrade requires some generations to be behaviorally effective through the fine-tuning of its parameters. NEAT algoritm implements the speciation technique to preserve and protect new topological innovations. The individuals are grouped into niches or species, so that the members of each niche are genetically similar.

NEAT algorithm proposes a *similarity distance*, $\delta(\mathbf{g}_i, \mathbf{g}_j)$, between pairs of genotypes to determine their kinship. It is described in Equation 2,

$$\delta(\mathbf{g}_i, \mathbf{g}_j) = c_1 E_{ij} + c_2 D_{ij} + c_3 \frac{1}{d_c} \|\Theta_i - \Theta_j\|_1 \tag{2}$$

where Θ_i is the vector of numeric parameters (i.e. weights and neuron parameters) of genotype $\mathbf{g}_i$, $\|\cdot\|_1$ is the L_1 norm, and E_{ij}, D_{ij} are the excess and disjoint genes between genotypes i and j. Moreover, d_c is the number of matching genes and c_1, c_2, and c_3 are hyperparameters that adjust the importance of each term for the final similarity distance. Note that the lower the value of $\delta(\mathbf{g}_i, \mathbf{g}_j)$, the stronger the kinship between the genotypes. The authors of NEAT proposed a compatibility threshold, δ_t, so that two genotypes, $\mathbf{g}_i$, $\mathbf{g}_j$, are similar enough provided that $\delta(\mathbf{g}_i, \mathbf{g}_j) \leq \delta_t$.

Every niche has a representative genotype that is updated every generatation at the end of the speciation process. Therefore, the speciation phase, in which every individual in the population is assigned to a niche, operates as follows:

(1) Every genotype, $\mathbf{g}_k$, in the population is compared to each of the representative individuals of each niche. Such comparison is performed using the similarity metric δ. Subsequently, the most similar representative individual (and, thus, the closest niche) is selected as the one with minimum similarity distance.

(2) If the similarity distance to the closest niche representative is lower than δ_t then $\mathbf{g}_k$ is directly associated to that species. Conversely, if the similarity distance is higher than δ_t, then the genotype is not similar enough to any species, and, thus, a new niche is created (with $\mathbf{g}_k$ as the only member and representative genotype).

(3) After the niche allocation process, a new representative is assigned in each species as the niche's genotype with lowest similarity distance to the prior representative individual.

As it was already introduced in Figure A.3, selection and crossover operators are separately applied for each species. Moreover, each niche has a different number of individuals to be picked during the selection operation. Specifically, Equation 3 displays the number of individuals, $n_{off,i}$, to be selected for the i-th species,

$$n_{off,i} = \left\lfloor \frac{\overline{F}_i}{F_{tot}} P \right\rfloor \tag{3}$$

where $\overline{F}_i$ is the average fitness score of the i-th species, and F_{tot} is the total fitness score of the population (Equation 4).

$$F_{tot} = \sum_{k=1}^{P} FF_k \tag{4}$$

B Continuous-time recurrent neural networks

A Recurrent Neural Network (RNN) is a type of ANN with feed back connections that allows the past events to be retained. Continuous-Time Recurrent Neural Networks (CTRNNs) [247] are RNNs with continuous-time dynamics. In this Appendix, CTRNNs and its constituent components are formally described. Firstly, its most basic building block, namely, the neuron, is introduced. Thereafter, the CTRNN is defined and formalized as a complete neural network.

B.1 The neuron model

In this PhD Thesis, the neuron dynamics that are considered in all the experiments are based on the firing rate model. Firing rate models (see e.g. [263]) provide a simplified model of spiking neuron models with rate coding schemes (i.e. information is encoded in the rate of action potentials produced by neurons, rather than their precise firing timings).

Considering a rate coding, firing rate models simplify spiking neurons using a non-linear mapping that converts the neuron's membrane voltage ($v(t)$) to the firing rate of the neuron ($u(t)$). Therefore, the output of the neuron is no longer a spike train, but rather a continuous signal. The dynamics of the firing rate model is shown in Equations 5 and 6,

$$\tau \frac{dv(t)}{dt} = -v(t) + \mathrm{I}(t), \tag{5}$$

$$u(t) = F(g(v(t) + \beta)). \tag{6}$$

Firstly, $\mathrm{I}(t)$ is the total input current injected to the neuron through synaptic connections. The source of the current can be either a external stimulus (e.g. sensor readings), or it can be produced by the activation of other connected neurons. τ is the decaying time constant of the neuron, that establishes how quickly the membrane voltage converges to stable fixed points in response to the injected current. The constant β is the offset value of the neuron. Additionally, g is the gain factor of the neuron, that defines how quickly the neuron can change from being innactive to firing with maximum rate, and vice versa. $u(t)$ is the firing rate of the neuron (also known as *activation*), and $\mathrm{F}(\cdot)$ is the neuron's activation function that maps the membrane voltage, $v(t)$, to the neuron's firing rate, $u(t)$. Two commonly used activation functions, and the ones used in this PhD Thesis, are the sigmoid function (Equation 7) and the hyperbolic tangent function (Equation 8).

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{7}$$

$$\tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \tag{8}$$

Notice that $u(t)$ takes values in $(0, 1)$ when a sigmoid function is applied, whereas it is constrained to $(-1, 1)$ when a hyperbolic tangent activation is used.

Figure B.1 shows the effect that the variables τ, β, g, and I have on the neuron dynamics. Specifically, the figures show the time evolution of the membrane voltage ($v(t)$) and the neuron's activation ($u(t)$) when different values of these variables are considered. In all the figures, a constant input currect of 3A is injected to the neuron.

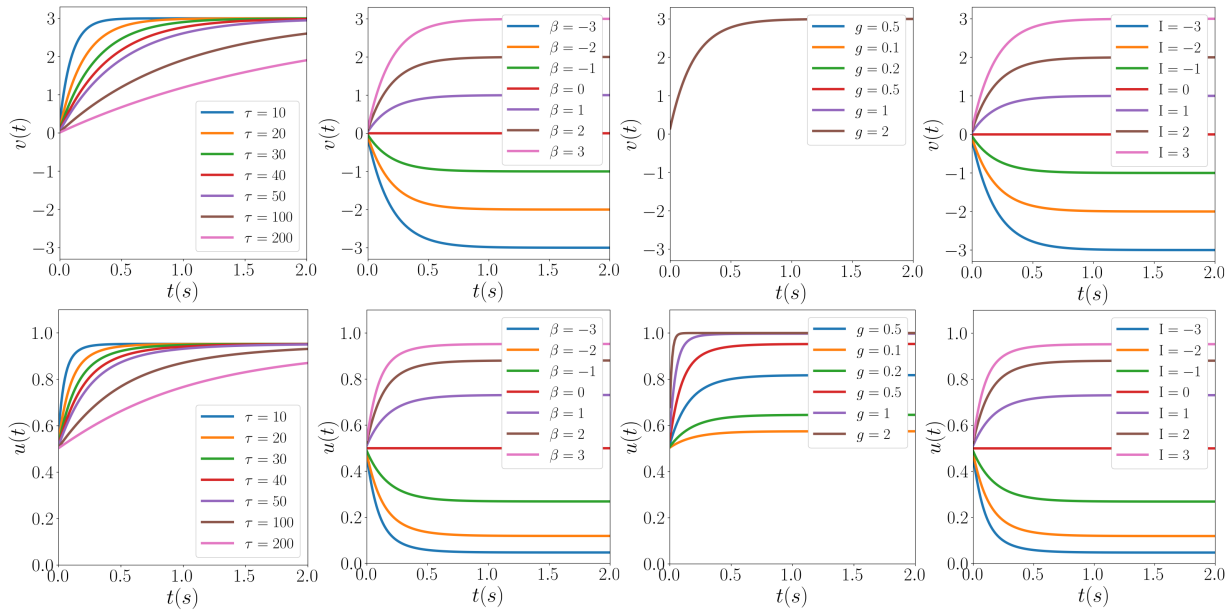


Figure B.1: Response of a firing rate neuron's dynamics ($v(t)$ and $u(t)$) when different values of the neuron's parameters are used. The input in all the scenarios is an input current of 3A.

B.2 The artificial neural network

In this PhD Thesis, a CTRNN is defined as a weighted directed graph $\mathcal{G}_{ann}(\mathcal{V}_{ann}, \mathcal{E}_{ann})$, so that the nodes of the graph in $\mathcal{V}_{ann}$ are firing rate point neurons, and the edges in $\mathcal{E}_{ann}$ are the synapses that connect pairs of neurons. Moreover, the neuron set is partitioned as in Equation 9,

$$\mathcal{N}_{ann} = \mathcal{I}_{ann} \cup \mathcal{N}_{ann} = \mathcal{I}_{ann} \cup \left(\mathcal{N}^{h}_{ann} \cup \mathcal{N}^{o}_{ann}\right) \tag{9}$$

so that $\mathcal{I}_{ann}$ are the input nodes and $\mathcal{N}_{ann}$ are the firing rate neurons of the CTRNN. For completeness, $\mathcal{N}_{ann}$ is also split into $\mathcal{N}^{h}_{ann}$, for the hidden neurons, and $\mathcal{N}^{o}_{ann}$, for the output neurons. The input nodes in $\mathcal{I}_{ann}$ are not actual neurons, but they rather represent the input signals that are used to feed the CTRNN model.

A weight is attached to every synapse in $\mathcal{E}_{ann}$, representing the strength of such connection. Let $w^{\mathcal{N}}_{ki}$ be the weight of the synapse connecting pre-synaptic neuron i with post-synaptic neuron k. Moreover, let $w^{\mathcal{I}}_{kj}$ be the weight that connects the j-th input node to the post-synaptic neuron k. The superindex indicates whether the pre-synaptic node is another firing rate neuron ($\mathcal{N}$) or an input node ($\mathcal{I}$). Using this notation, the total current injected to some neuron k is defined as in Equation 10,

$$\mathrm{I}_k(t) = \sum_{i \in \mathcal{N}_{ann}} w^{\mathcal{N}}_{ki} u_i(t) + \sum_{j \in \mathcal{I}_{ann}} w^{\mathcal{I}}_{kj} \phi_j(t), \tag{10}$$

where ϕ_j is the input signal injected to the j-th input node (normally originating from some sensor reading).

The synapse weights can be collected as matrix $\mathbf{W}$, defined in Equation 11.

$$\mathbf{W} = \left(\begin{array}{c|c} \mathbf{O}_{I\times I} & \mathbf{O}_{I\times N} \\ \hline \mathbf{W}_{\mathcal{I}} & \mathbf{W}_{\mathcal{N}} \end{array}\right) = \left(\begin{array}{ccc|ccc} 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \\ \hline w^{\mathcal{I}}_{11} & \cdots & w^{\mathcal{I}}_{1I} & w^{\mathcal{N}}_{11} & \cdots & w^{\mathcal{N}}_{1N} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ w^{\mathcal{I}}_{N1} & \cdots & w^{\mathcal{I}}_{NI} & w^{\mathcal{N}}_{N1} & \cdots & w^{\mathcal{N}}_{NN} \end{array}\right) \tag{11}$$

The matrix $\mathbf{W}$ is, in turn, the adjacency matrix of the directed graph $\mathcal{G}_{ann}$.

An illustrative example of a very simple CTRNN is provided in Figure B.2, showing a neural architecture with two input nodes (green), two hidden neurons (blue), and one output neuron (red). The value attached to each arrow indicates the weight of the corresponding synapse, resulting in the following CTRNN weight matrix:

$$\mathbf{W} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -0.25 & 0.5 & -0.1 & -0.5 & 0 \\ 0.5 & -0.25 & 0.5 & 0.1 & 0 \\ 0 & 0 & 0.7 & 0.7 & 0 \end{pmatrix}$$

Moreover, the CTRNN is fed by two sine waves of different frequencies. The execution of such CTRNN results in the output signal depicted at the right side of Figure B.2. Figure B.3 shows the two input sine waves (upper), and the mebrane voltage of the three neurons (lower).

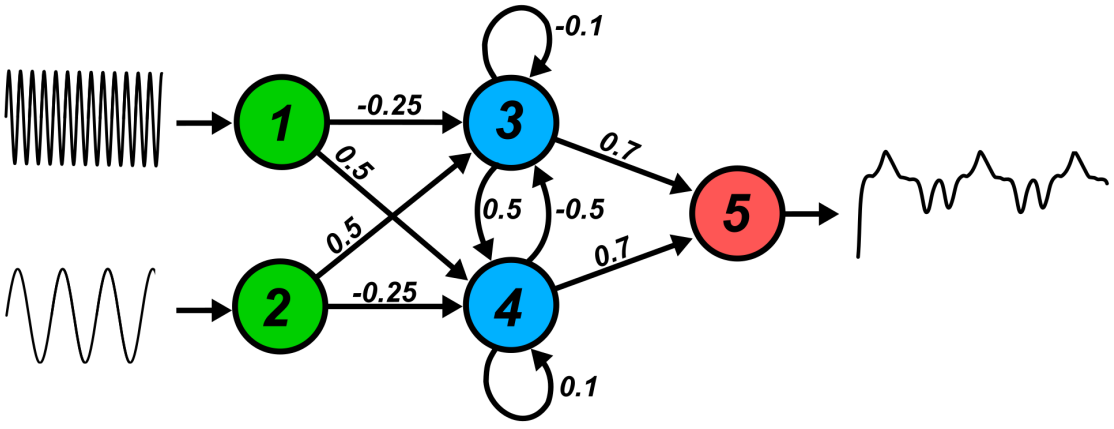


Figure B.2: Illustrative example of a simple CTRNN. It is composed by two input nodes, in green, two hidden neurons, in blue, and one output neuron, in red. Two sine waves at different frequencies are injected in each of the input nodes. The resulting output activity signal is shown on the rightmost side of the figure. All the neurons are implemented using the firing rate neuron and the weights of the synapses are defined next to the corresponding arrow.

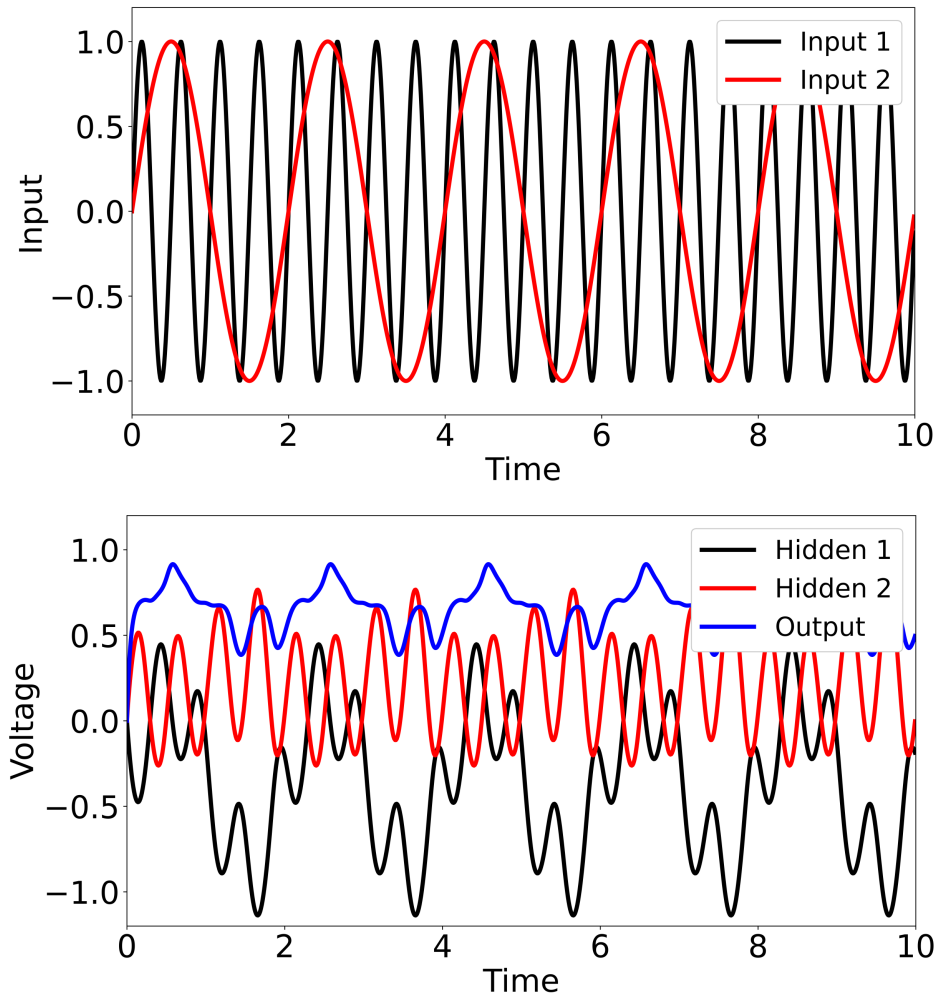


Figure B.3: Time evolution of input signals and neuron's membrane voltages of the illustrative example in Figure B.2.