The
University
Of
Sheffield.

# Firefly-Inspired Synchronization in Swarms of Mobile Agents

**Fernando Perez Diaz**

*Supervisor:*

Dr Roderich Groß

A Thesis Submitted for the Degree of
Doctor of Philosophy

30th September 2016

A mi familia

As you set out for Ithaka
hope your road is a long one,
full of adventure, full of discovery.

*Ithaka* Konstantinos Kavafis

# Abstract

Synchronization can be a necessary prerequisite to perform coordinated actions or reach consensus in decentralized multi-agent systems, such as robotic swarms and sensor networks. One of the simplest distributed synchronization algorithms is firefly synchronization, also known as pulse-coupled oscillator synchronization. In this framework, each agent possesses an internal oscillator and the completion of oscillation cycles is signaled by means of short pulses, which can be detected by other neighboring agents. This thesis focuses on a realistic mode of interaction for practical implementations, in which agents have a restricted field of view used to detect pulses emitted by other agents. The effect of agent speed on the time required to achieve synchronization is studied. Simulations reveal that synchronization can be fostered or inhibited by tuning the agent (robot) speed, leading to distinct dynamical regimes. These findings are further validated in physical robotic experiments. In addition, an analysis is presented on the effect that the involved system parameters have on the time it takes for the ensemble to synchronize. To assess the effect of noise, the propagation of perturbations over the system is analyzed. The reported findings reveal the conditions for the control of clock or activity synchronization in swarms of mobile agents.

# Statement

I declare that this thesis is original work, apart from the exceptions indicated by a reference in the text, and that it has not been submitted to any other university for examination.

# Acknowledgments

I would like to express my sincere gratitude to my supervisor, Dr. Roderich Groß, for his invaluable support and guidance during the entirety of the PhD. I am also eternally in debt to my collaborator, Dr. Rüdiger Zillmer, whose input has been priceless.

I would like to appreciate the friendships with Brais, Umberto, Juan, Stefan, Yuri and Matt, who supported me through thick and thin. Moitísimas gracias, grazie mille, muchísimas gracias, danke von Herzen, muito obrigado, for the fun, the chats, the outings, and all the shared experiences. I would also like to thank my other colleagues in the Natural Robotics Lab, Melvin, Jianing, Wei, Chris, Gabriel, Shen, João and Anil; and fellow PhD students, Tiago, Christina, Nat and James, for accompanying me during this process.

Gracias a mis amigos españoles en Sheffield, Pablo, Paloma, Ferrán, Javi, Sergio, Graciela, Rubén, Sonia, Esther, Julián y Lucía, por vuestra amistad y por hacerme sentir como en casa y sentir algo menos de morriña.

A mis padres, Tere y Fernando, mi hermana Celia, y a mis abuelos, Guillermo, Marisa, Antonio y Luisa, os debo todo. Gracias por la educación y oportunidades brindadas, por el apoyo moral y económico, por todo vuestro amor y cariño, y por *aturarme* en lo bueno y en lo malo. Os quiero.

Хайрт Азаа минь, докторын зэргээ хамгаалах хугацаанд надтай хамт миний дэргэд байсанд баярлалаа. Би чамд хайртай.

# Contents

# List of Figures

# Chapter 1

# Introduction

Synchronization refers to a process by which two or more systems adjust one of their dynamical properties over time to achieve a common behavior by means of their interaction (Boccaletti et al., 2002). As such, synchronization constitutes a pervasive concept in the natural sciences, as well as in technological applications. Crickets chirping in unison, neurons of the brain firing together, cardiac pacemakers rhythmically signaling the pumping of the heart, the tidal locking of the Moon, or the coherence of laser beams are but a few examples of synchronization phenomena.

One widely studied example of synchronization is the rhythmic synchronous flashing by large groups of males of some firefly species, in particular, tropical fireflies in Southeast Asia (Buck, 1988). Each species of firefly may achieve this flash synchronization in a different manner, but the basic mechanism is as follows: each insect in the colony has an internal biological clock that indicates the timing of the flashes; by observing the flashes of other individuals, either the timing or frequency of its own flashes gets slightly shifted in an attempt to match these external stimuli (Murray, 2002). In this way, spontaneous order appears without a leader, in a process of self-organization. Synchronization of these kinds of systems, called pulse-coupled oscillators, is not unique to fireflies, but is ubiquitous in nature. This gave rise to intensive research in a multitude of scientific fields (Pikovsky et al., 2001; Arenas et al., 2008).

Most of the research on the synchronization of pulse-coupled oscillators focuses on static networks of oscillators. Recently, however, there has been increasing interest in the study of synchronization in networks of mobile oscillators (Fujiwara et al., 2011; Prignano et al., 2012, 2013), more specifically, in networks whose nodes represent physical agents that move in an environment and interact with local neighbors. These studies aim to understand what role topological changes, due to the mobility of the oscillators, play on the synchronization of these systems.

## 1.1 Motivation

The motivation for this thesis stems from the potential applications of firefly synchronization in swarms of mobile agents.

The self-organizing behavior of populations of pulse-coupled oscillators is akin to that of swarm-intelligent systems in the field of Artificial Intelligence. Swarm intelligence research addresses the design of intelligent multi-agent systems drawing inspiration from the collective behavior of social animals, like colonies of insects, flocks of birds or schools of fish (Blum and Groß, 2015). These systems, made of a number of relatively simple and limited individuals, can achieve a complex collective behavior through local interactions without any centralized control (Bonabeau et al., 1999; Beni, 2005; Nouyan et al., 2009).

Swarm robotics and, to a lesser extent, mobile sensor networks, could be considered applications of swarm intelligence, aiming at exploiting the robustness provided by the self-organization of swarm-intelligent systems. These systems typically consist of simple or limited robots with restricted or local sensing or actuating capabilities. Like social insects, their strength is in their numbers and not in any particular individual. A swarm-intelligent behavior of its own, firefly synchronization can also be beneficial for robotic systems and sensor networks, enabling them to reach consensus, coordinate their actions, merge their sensory information, or optimize communication and battery usage (Ren et al., 2007). On the other hand, desynchronization, or temporal incoherence, has also been utilized for task distribution over time as it could help agents to not interfere with each other (Simeone et al., 2008).

As pulse-coupled oscillators operate at the physical layer by transmitting simple identical pulses, rather than packet messages, they are especially suitable in noisy or communication-limited environments (Wang et al., 2012, 2014). In addition, due to its simplicity, synchronization of pulse-coupled oscillators is inherently decentralized and robust. These properties are what makes this kind of synchronization particularly relevant to systems formed by a large number of simple individuals that operate in a distributed manner, such as sensor networks and robot swarms.

## 1.2 Problem Definition

Most of the previous work on mobile pulse-coupled oscillators (MPCOs) showed that synchronization occurs monotonically faster with higher agent speed if each agent influences others lying within a certain range (Prignano et al., 2012). At high speeds all agents interact with each other frequently, whereas at low speeds the neighborhood of any particular agent remains unchanged for long periods of time, leading to a rapid local synchrony but needing a longer time to achieve global synchronization. In contrast, if the agents influence only their nearest neighbor a regime of intermediate speeds is observed, in which synchronization is inhibited, while both the slow and fast regimes remain the same (Prignano et al., 2013).

The dramatic difference between the effect of nearest neighbor interaction (non-monotonic dependence of synchronization time with respect to agent speed) compared to finite radius interaction (monotonic dependence of synchronization time with respect to agent speed) is the point of departure for this thesis. This work explores the effects of the interaction rule and agent speed on the time it takes for the system to synchronize (synchronization time) with a view towards employing this knowledge in swarm robotics applications. As mentioned in the preceding section, both synchronization and desynchronization have uses in real-world applications. One could exploit MPCO synchronization for this purpose by better understanding the dependence of the synchronization time on the agents' speed and further characterizing the emergence of the inhibitory regime in practical settings.

## 1.3 Aims and Objectives

In light of the above problem definition, the aim of this thesis is to advance the state of the art on mobile pulse-coupled oscillator synchronization, and apply the acquired knowledge to the swarm robotic field.

The specific objectives are:

- To conduct a literature review of the current state of MPCO synchronization and its applications to practical domains. This review will contextualize the present work, and the results of this thesis will be compared and contrasted with existing approaches.

- To understand the factors that influence the time required for a system of MPCOs to synchronize. In particular, this thesis will analyze: the effect of agent speed on synchronization time; the effect of the system's parameters, such as number of agents and size of the environment; the effect of different interaction rules, specifically those applicable to real-world scenarios; and the effect of different oscillator and interaction models.

- To translate and verify the above results into a swarm of real robots. Special focus will be put on the dependence of the synchronization time on agent speed, and on determining whether the inhibitory regime can emerge in practical contexts.

- To study the robustness of the observed behaviors in the presence of perturbations and jitter. Real-world systems are subject to imperfections. Studying their impact on synchronization can help understand how to minimize their effect or even exploit it to our advantage.

## 1.4 Preview of Contributions

The contributions of this thesis are:

- An in-depth study of mobile pulse-coupled oscillator synchronization with a more practical mode of interaction for real applications: a cone of vision. Abstract simulations, in which the oscillators are point-like particles, demonstrated that one can switch between the monotonic and nonmonotonic behaviors by appropriately tuning the dimensions of such a cone. Namely, for a small or narrow cone the nonmonotonic dependence of synchronization time as a function of agent speed is found. As the cone widens the nonmonotonicity gradually disappears and the monotonic behavior is retrieved.

- An experimental validation of the above results in a swarm of real robots, despite the effects of embodiment. That is, collisions between robots are taken into account as well as occlusion of the firing signals. This is first validated in robotic simulations and further confirmed in a swarm of real robots. In this work, robots emitting pulses by momentarily flashing LEDs are considered. These flashes can, in turn, be perceived by other robots using a directional camera.

- The characterization of two key features of the nonmonotonic curve and the understanding of how these features are affected by various system parameters in a robotic simulation. These parameters include the number of agents, the size of the environment, the period of oscillation and the coupling strength between oscillators.

- A metric that predicts the appearance of the nonmonotonic behavior. The impact on the synchronization time of different responses to a pulse found in the literature is investigated. In particular, it is found that only one family of responses produces the described nonmonotonicity.

- A study of mobile pulse-coupled oscillator synchronization with two other neighborhoods: cone of emission and K-nearest neighbor interaction. Similar to the cone of vision, in both cases, the monotonic and nonmonotonic behaviors can be retrieved by tuning the neighborhood size. The effect of the cone dimensions in the cone of emission is qualitatively equivalent to that of the cone of vision. Small or narrow cones result in a nonmonotonic behavior and, which gradually becomes monotonic with increasing cone dimensions. Similarly, for K-nearest neighbors connectivity, it is observed that by increasing $K$ the nonmonotonicity gradually disappears. This prompts to conclude that it is the average rate of neighbor change that leads to the nonmonotonic behavior.

- An in-depth study of the robustness of the different dynamical regimes to perturbations or jitter. It is observed that if agents move at fast speeds synchronization is resilient to perturbations. This is also true, although to a lesser extent, when agents move at slow speeds. However, in the intermediate, synchronization-hindering, regime, an initially completely synchronized system can be totally disrupted by a small perturbation in a single oscillator. Furthermore, an analogy between the behavior in this intermediate regime and stable chaos is drawn.

## 1.5 Publications

This thesis represents the author's own work, and includes a number of original contributions to scientific knowledge. The work presented herein is built around the following peer-reviewed papers that I produced as first author during my PhD:

1. **Perez-Diaz, F., Zillmer, R., and Groß, R.** (2015). *Firefly-inspired synchronization in swarms of mobile agents.* In Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS '15, pp. 279—286. IFAAMAS.

2. **Perez-Diaz, F., Zillmer, R., and Groß, R.** (2016). *Emergence and inhibition of synchronization in robot swarms.* In Proceedings of the 13th Interna-

tional Symposium on Distributed Autonomous Robotic Systems (to appear). Springer.

3. **Perez-Diaz, F., Zillmer, R., and Groß, R.** (2016). *Control of synchronization regimes in networks of mobile interacting agents.* Physical Review Applied (in prep.).

Publication 1 was presented orally as full paper by the author himself at the corresponding conference, held in Istanbul, Turkey.

The material in Publications 1, 2 and 3 corresponds to the contents of Chapters 3, 4 and 5 of this thesis, respectively. The introductory and related work material from all these three publications has contributed to the contents of Chapters 1 and 2.

During the course of my PhD studies, in addition to the aforementioned work, I have also contributed to other projects that are not featured in this thesis. These contributions have led to the following publications:

1. **Berdan, R., Prodromakis, T., Khiat, A., Salaoru, I., Toumazou, C., Perez-Diaz, F., and Vasilaki, E.** (2013). *Temporal processing with volatile memristors.* In 2013 IEEE International Symposium on Circuits and Systems, pp. 425—428. IEEE.

2. **Doyle, M. J., Xu, X., Gu, Y., Perez-Diaz, F., Parrott, C., and Groß, R.** (2016). *Modular hydraulic propulsion: A robot that moves by routing fluid through itself.* In 2016 IEEE International Conference on Robotics and Automation, pp. 5189—5196. IEEE.

## 1.6 Thesis Outline

This thesis is structured as follows:

- Chapter 2 provides an overview of background material and related work that places this thesis in context. The chapter starts in Section 2.1 with a historical introduction to synchronization. This introduction presents different examples of the phenomenon, since its first scientific discovery until the 20th century, when the science of synchronization commenced to be formalized. Section 2.2 defines synchronization. Its basic ingredients, "oscillators" and "coupling" are described in detail in Sections 2.2.1 and 2.2.2, respectively. Section 2.3 concerns "pulse-coupled oscillators", the type of oscillators studied in this thesis, and provides a formal definition. Section 2.4 reviews previous approaches and results in "mobile pulse-coupled oscillators", that is, pulse-coupled oscillators where the oscillators are agents moving in space. Section 2.5 discusses the existing applications of mobile-coupled oscillators in technical domains. In particular, Section 2.5.1 defines "sensor networks", describes the importance of synchronization in these systems, and presents the related research on pulse-coupled oscillators' synchronization in mobile sensor networks. Section 2.5.2, in turn, defines "swarm intelligence" and "swarm robotics", discusses the uses of synchronization in these systems, and describes previous approaches on synchronization through pulse-coupled oscillators in swarms of mobile robots.

- Chapter 3 introduces the framework developed and utilized in this thesis and presents our first approaches to understanding the effect of a cone of vision interaction in MPCO synchronization both in a particle and a robotic simulation. Section 3.1 describes the models used, such as the oscillator (Sec. 3.1.1) and neighborhood (Sec. 3.1.2) models; the details of the particle (Sec. 3.1.3) and robotic (Sec. 3.1.4) simulations; and the employed metric of synchrony (Sec. 3.1.5). Section 3.2 presents the results obtained in the particle simulations. It investigates the dependence of the synchronization time on agent speed and on the dimensions of the cone of vision (i.e. its radius and angle).

Section 3.3 presents the corresponding results in the robotic simulations. It shows an initial validation of presence of both the monotonic and nonmonotonic behaviors despite the embodiment of the agents. Section 3.4 summarizes the chapter.

- Chapter 4 extends the results of the previous chapter with more realistic simulations of the robotic environment and a validation of the results on a swarm of real robots. Section 4.1 presents the simulation environment and the chosen robots, as well as the employed algorithms. Section 4.2 describes the simulation setup and the obtained results, which display the dependence of the synchronization time on the dimensions of the cone and the speed of agents. Moreover, this section also finds the relationship between key features of the synchronization curve and various system parameters. Section 4.3 presents a validation of the simulation results on a swarm of real robots. It shows that the nonmonotonic behavior is observed also in reality and it is not an artifact of the simulations. Section 4.4 summarizes the chapter.

- Chapter 5 returns to an abstract particle simulation in order to further investigate all the factors involved in the emergence of the nonmonotonic behavior and the lack thereof. It also presents an in-depth study of the effect of perturbations and the robustness of synchronization regimes against disturbances. Section 5.1 presents the utilized methods, which extend those of Chapter 4 by providing a more generalized oscillator model with various responses to a pulse, and describing two other interaction models, K-nearest neighbors and cone of emission interaction. Section 5.2 presents the effect of the response function, which is the part of the oscillator model that characterizes the response to a pulse. Section 5.3 studies the effect of the interaction model and extends the results found with a cone of vision to K-nearest neighbors and cone of emission connectivities. Section 5.4 analyzes the effect of perturbations on the system, from the propagation of small perturbations applied initially to a single oscillator (Sec. 5.4.1) to a generalized clock jitter (Sec. 5.4.2). Section

5.5 summarizes the chapter.

- Chapter 6 concludes this thesis with a discussion of the presented results together with their limitations, and provides a number of potential directions for future developments.

# Chapter 2

# Background and Related Work

> *"... when we suspended two clocks so constructed from two hooks imbedded in the same wooden beam, the motions of each pendulum in opposite swings were so much in agreement that they never receded the last bit from each other [...] Further, if this agreement was disturbed by some interference, it reestablished itself in a short time. For a long time I was amazed at this unexpected result, but after a careful examination I finally found that the cause of this is due to the motion of the beam, even though this is hardly perceptible"*
>
> *Christiaan Huygens*, 1673

## 2.1 Synchronization: A Historical Introduction

In 1665, Dutch mathematician, physicist and inventor of the clock pendulum, Christiaan Huygens, realized an *"admirable effect which no one could have ever thought of"*: When two pendulum clocks were hanging side by side, they started oscillating together, without alteration. Struck by this phenomenon, he set a series of experiments, aiming to understand what caused that *"kind of sympathy"*, as he called it. He concluded that the imperceptible vibrations of the common support that held the clocks was providing a means for their interaction (Fig. 2.1). Huygens had given the first accurate description and understanding of **synchronization**.

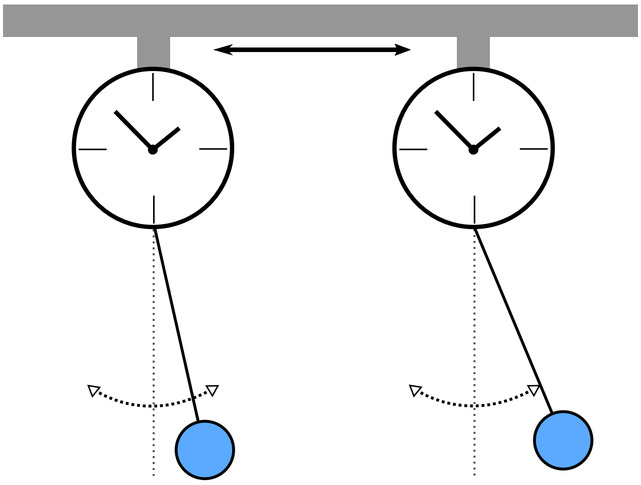


Figure 2.1: Diagram of Huygens' experiment. Two pendulum clocks are fixed to a common support. The motion of the pendula makes this support vibrate slightly. As a result, the clocks can interact with each other and synchronize their oscillations.

From that moment onwards, descriptions of other synchronization phenomena continued to appear in the scientific literature. In the early 18th century, the French astronomer and mathematician, Jean-Jacques Dortous de Mairan, recounted how the leaves of the haricot bean would move up and down with the changes of light during the day. The movement of the leaves would continue even if the plant was put in a dark room. It is now known that many other biological clocks can adjust their rhythms to the Earth's 24-hour day cycle, in what is known as the *circadian rhythm* (Winfree, 2001). In the 19th century Lord Rayleigh described the synchronization of acoustical systems, as he observed that two similar pipes would sound in unison despite small differences between them. In 1920, W.H. Eccles and J.H. Vincent discovered that two connected triode generators, devices capable of producing a periodic alternating current, would vibrate with a common frequency. This discovery became of great importance as one of the essential components of radio communication systems.

Over time it gradually became apparent that these, and many other, apparently unrelated phenomena, observed in a variety of fields, all followed some common laws and could be described by a unified theory. From the mid 20th century, led by the seminal paper of Winfree (1967), the science of synchronization started to blossom, and a vast body of research in mathematics and physics has emerged since then.

## 2.2 What is Synchronization?

Synchronization is a process whereby two or more oscillators adjust their rhythms until they achieve a common behavior due to their interaction(s) (Pikovsky et al., 2001; Boccaletti et al., 2002). For instance, in Huygens' clock experiment, each pendulum on its own exhibits an independent cyclical behavior. However, when two or more clocks are put on the same stand, their individual oscillations would influence each other's cycles, resulting in all the clocks ticking in unison.

### 2.2.1 Oscillators

A condition for synchronization is that oscillations must be *self-sustained.* That is, the oscillator is an active system that converts an internal source of energy into a rhythmic movement or oscillation. This oscillation persists over time until the source of energy expires, or a strong disturbing force[1] is applied to the system (Pikovsky et al., 2001). The movement of the haricot leaves described above is a perfect example of self-sustained oscillation, as the plant performs its daily motion even if it is hidden from the Sun. On the other hand, tides cannot be considered self-sustained. In the absence of the Moon, the oscillation of the ocean would vanish. Therefore, one cannot claim that the sea level synchronizes with the Moon's position on the sky.

Another important characteristic of oscillations is their *periodicity.* That is, the system repeats its motion cyclically with period $T$. In Huygens' clock example, the coordinates of the tip of the pendulum follow a periodic motion.

The space formed by the coordinates that uniquely represent all possible states of a dynamical system is called its *phase space.* The behavior of the system can be described by the trajectory of its coordinates in phase space. For a pendulum clock, the phase space can be formed by its angle with respect to the vertical, and its angular velocity. This example has a two-dimensional phase space but more

[1] If the magnitude of the perturbation is small, the oscillation could return to its original shape.

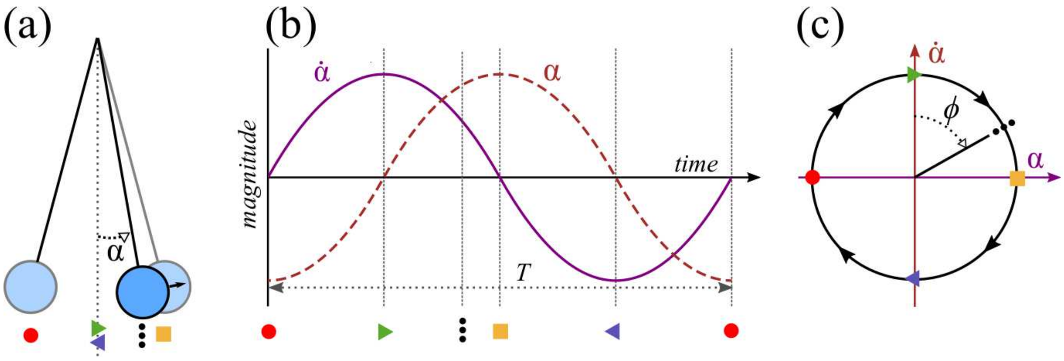


Figure 2.2: Example of a phase oscillator. (a) A pendulum clock swings back and forth. Five points along its trajectory are marked: ◯ and □ correspond to its leftmost and rightmost points, respectively; ◁ and ▷ represent the passing across the vertical when swinging left and right, respectively; and ⁝ is the current position of the pendulum. (b) Angular position $\alpha$ and angular velocity $\dot{\alpha}$ of the pendulum over time. (c) Trajectory of the oscillator in phase space (the plane formed by $\alpha$ and $\dot{\alpha}$). The phase $\phi$ represents the current point along the oscillation cycle.

complex ones can exist. Given its periodicity, an oscillator describes a closed curve in phase space called a *limit cycle* (Strogatz, 2014). Given an oscillator's limit cycle, its **phase** is a quantity that defines at which point along the trajectory the system is located at a particular time. The oscillator's phase $\phi$, is thus a periodic function of time,

$$\phi(t) = \phi(t + T)\,. \tag{2.1}$$

Figure 2.2(a) presents a pendulum clock as an example of a phase oscillator. Figure 2.2(b) plots the evolution of the pendulum's coordinates (angular position and velocity) as a function of time, whereas Figure 2.2(c) represents the oscillator's limit cycle (i.e. its closed trajectory in phase space).

Before a group of oscillators becomes synchronized, they could differ from each other in several ways: they could each follow slightly different trajectories (e.g. having different periods of oscillation); they could all have the same trajectory but start at different points along it (i.e. having different initial phases); or a combination of the two. After synchronizing, all oscillators will describe in unison a common trajectory in phase space (Figure 2.3). Naturally, if the initial individual trajectories are radically different the group of oscillators may not be able to synchronize.

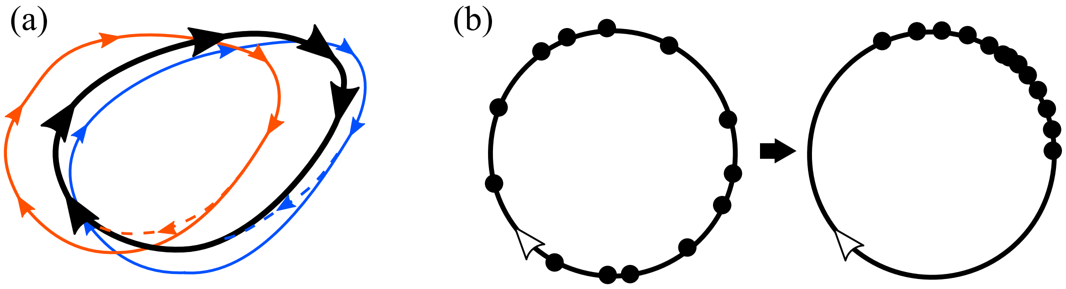


Figure 2.3: Examples of oscillator synchronization in phase space. (a) Two oscillators with initially different trajectories (thin lines) converge to a common behavior (thick line). (b) Several oscillators (black dots) describing the same trajectory but possessing different initial phases (left) start converging towards a synchronous oscillation.

### 2.2.2 Coupling

As mentioned earlier, some sort of interaction between oscillators is needed for them to synchronize. This interaction, or **coupling**, is any kind of force, exerted by one oscillator on another, that modifies the latter's trajectory in phase space (Pikovsky et al., 2001). In Huygens' experiment, the coupling between clocks occurs through the vibrations of the beam that supports them.

Coupling can be unidirectional or bidirectional (also called mutual) (Pikovsky et al., 2001; Boccaletti et al., 2002). In the former case, one oscillator drives the evolution of the other in a master-slave fashion, until the slave follows the trajectory of the master (Boccaletti et al., 2002). For instance, the motion of the haricot bean leaves synchronizes with the movement of the sun, but not the other way around. With bidirectional coupling, oscillators induce changes in each other's rhythms until they achieve a common behavior. Therefore, it is a self-organized behavior that is a result of their collective interactions (Pikovsky et al., 2001). This is the case with the pendulum clocks, where they mutually influence one another, or the rhythmic firing of cardiac pacemakers.

Coupling can be global (all-to-all) if every oscillator influences all other oscillators (Fig. 2.4(a)). A good example of this is the synchronized clapping of the audience at a theater, where each person hears the sound produced by all other people in the hall (Pikovsky et al., 2001). However, in the general case, oscillators form a *complex*

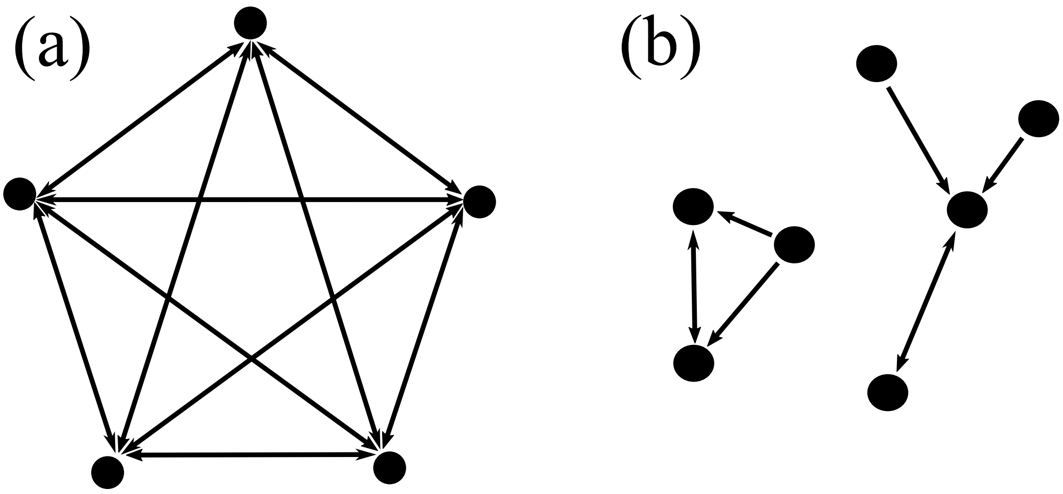


Figure 2.4: Examples of coupling between oscillators. Each oscillator is represented by a node of a graph (depicted as a circle). An arrow between two nodes denotes that the oscillator at the base of the arrow influences the oscillator at its tip. A two headed arrow means that the two linked oscillators influence each other. (a) Globally connected network. (b) A more complex network of interactions.

*network* of interactions, where each oscillator only influences, or is influenced by, a subset of all oscillators (Fig. 2.4(b)). Oscillator networks can be represented by a graph, in which each oscillator is a vertex and the edges correspond to couplings between oscillators (Arenas et al., 2008).

An oscillator can influence another oscillator continuously over time or discretely, in *pulses*. Tidal locking, or synchronous rotation, is a form of continuous coupling between the Moon's rotation along its own axis and its translation around the earth caused by its gravitational interaction with the Earth (Strogatz, 2003). The Kuramoto model (Kuramoto, 1975) is perhaps one of the most studied models where interactions occur continuously over time, and it has found numerous applications in biological and chemical oscillators (Arenas et al., 2008). On the other hand, in synchronized clapping, the coupling occurs discretely, in pulses, corresponding to individual claps. This is also the case for crickets chirping in harmony, or fireflies flashing in unison.

## 2.3 Pulse Coupled Oscillators

> *"Some twenty years ago I saw, or thought I saw, a synchronal or simultaneous flashing of fireflies. I could hardly believe my eyes, for such a thing to occur among insects is certainly contrary to all natural laws."*
>
> *Philip Laurent*, 1917

Rhythmic synchronous flashing by large groups of males of some firefly species, in particular, tropical fireflies in Southeast Asia, has been reported for hundreds of years. This phenomenon awoke the interest of biologists from the beginning of the 20th century, and a vast number of research articles appeared during that time, describing and modeling this behavior (see Buck, 1988, for a review). Each species of firefly may achieve this flash synchronization in a different manner, but the basic mechanism is as follows: each insect in the colony has an internal biological clock that indicates the timing of the flashes; by observing the flashes of other individuals, either the timing or frequency of its own flashes gets slightly shifted in an attempt to match this external stimuli (Ermentrout, 1991; Murray, 2002). In such a way, spontaneous order appears without a leader, in a process of self-organization.

The synchronization of populations of **pulse-coupled oscillators (PCOs)** is not unique to fireflies. It is observed in numerous natural systems, such as pacemaker cells in the heart, pacemaker neurons in the brain (for instance, those producing the circadian rhythm), the in-unison chirping of crickets, or even the synchronization of women's menstrual cycles (Winfree, 2001; Mirollo and Strogatz, 1990).

The most frequently studied PCOs model, and the one considered in this thesis, involves *integrate-and-fire oscillators* (IFOs) synchronizing their phases (Arenas et al., 2008). The phase of such an oscillator grows monotonically with time in an interval $I$ (for instance $\phi$ grows between 0 and $2\pi$ in Fig. 2.2(c)). That is

$$\frac{d\phi}{dt} = f(\phi)\,, \tag{2.2}$$

where $f(\phi) : I \to I$ and $\frac{df(\phi)}{d\phi} > 0$.

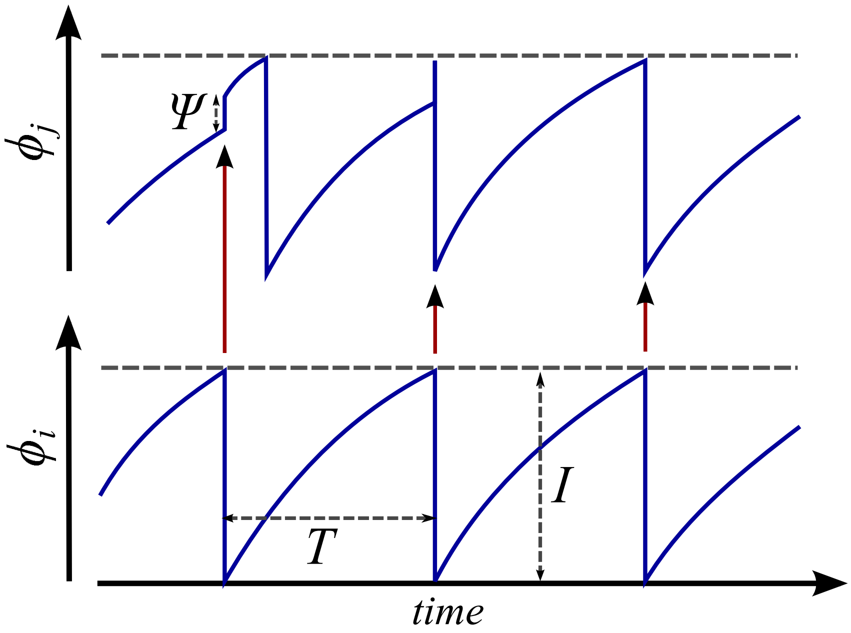


Figure 2.5: Two pulse-coupled IFOs. Oscillator $i$ (bottom) influences oscillator $j$ (top) until synchronization is achieved. $\phi_i$ displays a periodic oscillation in the interval $I$ with period $T$. When $\phi_i$ reaches the maximum of the interval it fires a pulse that induces a phase response $\psi$ in $\phi_j$.

When the phase reaches the maximum of the interval, it is immediately reset to its minimum value (see bottom plot in Fig. 2.5). At that moment, the oscillator signals the completion of a cycle with a pulse. Other oscillators, to which the firing oscillator is coupled, receive this pulse and, in turn, modify their individual phases (Peskin, 1975; Mirollo and Strogatz, 1990). That is, at the moment of firing of oscillator $i$ at time $t$,

$$\phi_i\left(t^-\right) = 1 \Rightarrow \begin{cases} \phi_i\left(t^+\right) = 0 \\ \phi_j\left(t^+\right) = \phi_j\left(t^-\right) + \Psi\left(\phi\left(t^-\right)\right), \end{cases} \tag{2.3}$$

where $j$ is an oscillator receiving the pulse of $i$, and $\phi_{i,j} \in I = [0,1]$ represent the corresponding oscillator' phases. The map $\Psi : I \to \mathbb{R}$ is called a *phase response curve* (PRC) and represents the amount of phase update induced after receiving a pulse (Buck, 1988). Figure 2.5 depicts the synchronization of two IFOs. The top oscillator receives pulses from the bottom oscillator. After two pulses and corresponding phase updates (PRCs), both oscillators are synchronized.

Mirollo and Strogatz (1990) pioneered the theoretical study of pulse-coupled oscillators by determining the sufficient characteristics of a PRC that guarantee that a system of globally coupled IFOs will always synchronize. Namely, $\Psi(\phi)$ should be some smooth, monotonically increasing and concave down function of $\phi$. Due to the ubiquity of synchronization in communities of pulse-coupled constituents, a vast body of research devoted to studying synchronization in PCOs subsequently emerged. These pieces of research dealt with different aspects or variations of the same problem as well as practical applications (Rosenblum and Pikovsky, 2003). For instance, synchronization in lattices, where the oscillators only form local connections, was studied (Corral et al., 1995a,b). Some works focused on the dynamics of the synchronization process (Pazó and Montbrió, 2014), observing, for instance, clusters of synchronized oscillators (Mauroy and Sepulchre, 2008), or partial synchronization (van Vreeswijk, 1996). In engineering, PCOs have been used to synchronize clocks in sensor networks (Scaglione and Hong, 2003; Hong and Scaglione, 2005; An et al., 2011; Carbone et al., 2013). Synchronization of certain chemical oscillators, such as pulse-coupled Belousov–Zhabotinsky oscillators, was also studied (Horvath et al., 2012, 2015). In biology, synchronization of glycolytic oscillations in a population of yeast cells was understood as a form of synchronization of PCOs (Richard et al., 1996). In neuroscience, synchronous activity of neurons mediated by synaptic pulses was studied (Izhikevich, 1999; Tateno and Robinson, 2007; Marella and Ermentrout, 2008; Stam and Van Straaten, 2012).

## 2.4 Mobile Pulse-Couple Oscillators

Most research in oscillator synchronization has focused on static networks. Recently, however, synchronization in temporal, or time-evolving, networks has garnered increasing interest. More specifically, in networks whose nodes represent physical agents that move in an environment and interact with other agents in their local vicinity (Sarkar and Parmananda, 2010; Fujiwara et al., 2011, 2016; Janagal and Parmananda, 2012; Prignano et al., 2012, 2013; Uriu et al., 2013; Gómez-Gardeñes et al., 2013; Shen et al., 2013, 2014; Wang et al., 2014, 2015; Levis et al., 2016). The study of such mobile oscillators aims at understanding the impact that motion-induced topological changes play on synchronization. When the oscillators at hand are PCOs, they are often referred in the literature as **mobile pulse-coupled oscillators (MPCOs)** (Wang et al., 2015).

Oscillator mobility leads to novel behavior if the interaction is restricted to a local neighborhood, forming a time-evolving network. When oscillators only influence others in their proximity, the connectivity of the resulting network may not guarantee that signals will propagate over all the units. Therefore, in general, agent mobility is necessary to achieve global synchronization.

Most of the existent work on MPCOs deals with *random geometric graphs* (Penrose, 2003), where two oscillators interact with each other if the distance between them is smaller than a certain range. It was found that, for a system of IFOs with this kind of connectivity, synchronization occurs faster with higher agent mobility (Prignano et al., 2012; Uriu et al., 2013; Wang et al., 2014, 2015). This is what one would intuitively expect: on the one hand, at high speeds every oscillator interacts with all others very frequently, and the dynamics of the system can be reduced to the mean-field (Uriu et al., 2013); on the other hand, at low speeds the neighborhood of any particular oscillator does not often change, leading to rapid local synchrony but requiring a longer time to achieve coherence at a global scale (see Fig.2.6, solid line). In addition, the larger the range of influence the faster synchronization will occur, as a greater range of influence allows for greater connectivity between oscillators (Prignano et al., 2012; Uriu et al., 2013; Wang et al., 2014, 2015).

The aforementioned results are qualitatively consistent for various phase response curves: Wang et al. (2015) used an *additive* PRC, where a constant is added to an oscillator's phase upon receiving a pulse; Prignano et al. (2012) used a *multiplicative* PRC, where an oscillator's phase increased proportionally to its value upon receiving a pulse; and Wang et al. (2012, 2014) used a PRC, known as *delay-advance*, in which an oscillator's phase is decreased after receiving a pulse early in its cycle, or increased if the pulse is received late in its cycle. Different motion dynamics have also been

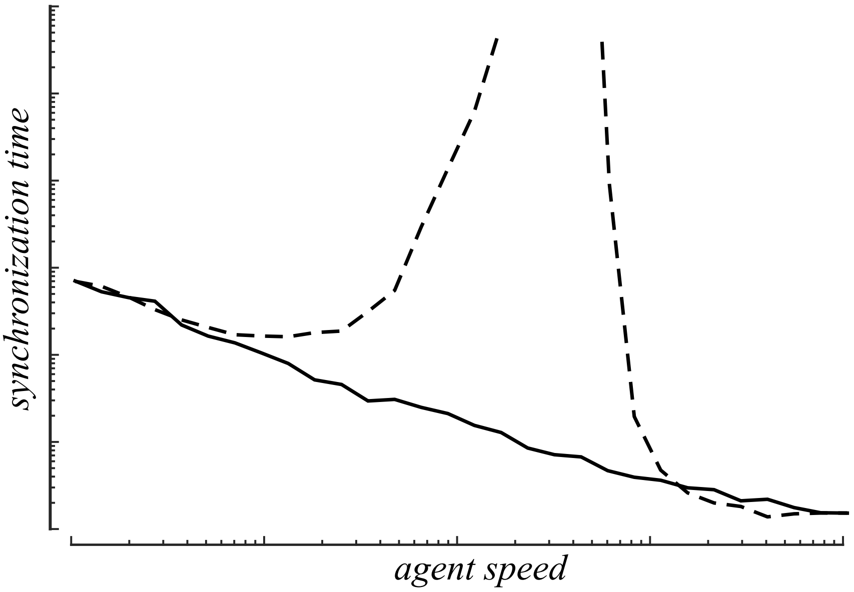


Figure 2.6: Synchronization time as a function of agent speed in MPCOs, represented in a log-log scale with arbitrary units. The solid line reproduces the monotonical dependence observed when oscillators influence others within a range (replication of Prignano et al., 2012). The dashed line reproduces the nonmonotonical behavior observed when oscillators influence their nearest neighbor (replication of Prignano et al., 2013).

studied yielding equivalent results: Prignano et al. (2012) considered oscillators moving in straight lines in a two-dimensional (2D) environment, and reorienting at random after receiving a pulse; and in Wang et al. (2012, 2014) oscillators performed a random walk in a 2D environment.

In the results described above, using an interaction range connectivity, the time until synchronization is reached, or **synchronization time**, decreases monotonically with oscillator mobility. In contrast, Prignano et al. (2013) reported the counterintuitive result that for nearest neighbor connectivity this is no longer the case. Instead, a nonmonotonic dependence is found (Fig. 2.6, dashed line). The behavior for slow and fast speeds remains the same as for range interaction. That is, it takes long time to synchronize with low mobility but synchronization occurs fast for high unit speeds. However, an intermediate regime emerges where synchronization is inhibited.

## 2.5 Applications of Mobile Pulse-Coupled Oscillators

The MPCO framework, of locally interacting moving oscillators, has received growing attention in technical domains, particularly in distributed multi-agent systems. As they are based on simple identical pulses, rather than packet messages, MPCOs make for suitable scalable protocols for decentralized systems (Deligeorges et al., 2015a). Specifically, MPCOs have been recently studied in the contexts of mobile sensor networks (Wang et al., 2012, 2014; Wang et al., 2015; Deligeorges et al., 2015a,b), mobile ad hoc networks (Tyrrell et al., 2006; Wang et al., 2015), and swarm robotics systems (Christensen et al., 2009; Trianni and Campo, 2015).

### 2.5.1 Sensor Networks

Wireless sensor networks, or simply **sensor networks**, are systems consisting of large numbers of low-cost, low-power, small (from the centimeter down to nanometer scale) autonomous sensor nodes (Akyildiz et al., 2002a,b; Akyildiz and Kasimoglu, 2004). These tiny devices are deployed in an environment to study phenomena occurring in it. Some of the main uses of sensor networks are in military applications, such as surveillance or reconnaissance; environmental or agricultural applications, such as monitoring soil, pollution, temperature, etc.; or health applications, for instance collecting physiological data to track a patient's condition (for comprehensive surveys see Akyildiz et al., 2002a,b; Yick et al., 2008). Sensor networks can comprise hundreds to thousands of sensors. Due to the large numbers of units, together with the low-power constraints and their small size, sensing is typically performed in a distributed manner (Akyildiz et al., 2002a). In addition, sensors often need to cooperate with each other to carry out the assigned sensing tasks (Akyildiz and Kasimoglu, 2004).

Time synchronization can be a prerequisite for the operation of sensor networks, as it provides a common time frame to different units and, thus, allows them to cooperate with each other (Wu et al., 2011; Carbone et al., 2013). It is, therefore, essential when fusing data from different sensors, particularly for the detection or monitoring of temporally correlated data, such as audio or video in order to detect a moving target (Hong and Scaglione, 2005; Wu et al., 2011). Temporal synchronization also plays a crucial role in power management in sensor networks. Units might be left unattended for their whole lifetimes. Energy efficiency and energy saving are, thus, critical factors in the design of sensor networks. Synchronization is key to coordinate sleep wake-up scheduling (Yu et al., 2010; Wu et al., 2011). For instance, synchronization was proposed for power saving in sensor networks, by keeping the sensors' communication channels idle most of the time and only using them at precise times (Yu et al., 2010). Likewise, for communication purposes, time synchronization is also essential for time-based channel sharing and transmission scheduling (Yu et al., 2010).

#### 2.5.1.1 PCOs Synchronization in Sensor Networks

The large numbers of units in a sensor network and the typical lack of central control make synchronization a challenging task (Akyildiz et al., 2002b). Numerous centralized synchronization protocols have been proposed. However, they can fail if the central station malfunctions, and could be, therefore, vulnerable to attacks (Hong and Scaglione, 2005). The ad-hoc and distributed nature of sensor networks requires synchronization protocols that are decentralized, simple and scalable (Hong and Scaglione, 2005). Synchronization protocols based on pulse-coupled oscillators are under active development and study (Scaglione and Hong, 2003; Hong and Scaglione, 2005; Werner-Allen et al., 2005; Tyrrell et al., 2007; An et al., 2011; Pagliari and Scaglione, 2011; Wang and Doyle III, 2012; Wang et al., 2013; Carbone et al., 2013; Núñez et al., 2015). The simplicity of PCOs, based on simple identical pulses, rather than packet messages, makes them ideal for sensor networks. These pulses operate exclusively at the physical layer, which, in turn, eliminates the inaccuracies due to Media Access Control (MAC) layer delays, protocol processing, or software implementation (Wang et al., 2015; Deligeorges et al., 2015a). Moreover, all received pulses are treated identically, independently of the origin, therefore eliminating or reducing the memory requirements. In conclusion, the PCO synchronization strategy is inherently scalable (Deligeorges et al., 2015a). Given that the inspiration for

PCO synchronization is drawn from the behavior of fireflies, the derived protocols are also often referred to as **firefly synchronization** (Werner-Allen et al., 2005; Tyrrell et al., 2006; Yick et al., 2008).

The aforementioned works deal with sensor networks in a static configuration. In recent years, however, a number of papers emerged studying PCO-based synchronization protocols with moving sensors (Wang et al., 2012, 2014; Wang et al., 2015; Deligeorges et al., 2015a,b). In other words, they were studying mobile pulse-coupled oscillators. In the context of mobile sensor networks, the decentralized nature of firefly synchronization presents yet another advantage compared to packet-based algorithms. The latter typically requires a hierarchical structure, such as a spanning tree, which is difficult to maintain due to the sensor movement (Wang et al., 2012).

Wang et al. (2012, 2014) and Wang et al. (2015) studied sensors performing a random walk in a square 2D environment, where each unit could send pulses to others within a certain range or radius around it. Using various phase response curves, the authors calculated analytically the expected time until synchronization as a function of the system parameters. They found that synchronization time decreases monotonically with the average unit speed, as mentioned earlier. This result was further verified in simulations with networks of nine sensors. Finally, Deligeorges et al. (2015a,b) presented field experiments where they successfully validated the use of PCO synchronization for data fusion applications with mobile sensors. The mobile sensors in question were acoustic gunfire detectors mounted in unmaned aerial vehicles (UAVs) (Deligeorges et al., 2015a) or carried by soldiers (Deligeorges et al., 2015a).

In addition to the applications of synchronization in sensor networks, desynchronization (or temporal incoherence) has also been exploited in the context of task scheduling (Simeone et al., 2008). By performing tasks at different times, monitoring activities or sleeping cycles can be organized and distributed (Degesys et al., 2007; Patel et al., 2007). Desynchronization can also benefit communication in this context, effectively implementing time division medium access (TDMA), a well know medium access control protocol. By sending messages at different times agents do not have to compete for the shared medium and can avoid message collisions (Degesys et al., 2007; Patel et al., 2007; Cornejo and Kuhn, 2010). PCO mechanisms have also been studied to achieve desynchronization with a view towards channel-sharing applications (Degesys et al., 2007; Patel et al., 2007; Cornejo and Kuhn, 2010; Buranapanichkit et al., 2015). In these scenarios, desynchronization is mostly used to refer to equidistant distribution in time of the signals, tasks, etc. Note that in other contexts this effect may not be called desynchronization, instead the terms phase locking or splay state may be used (Pikovsky et al., 2001).

## 2.5.2 Swarm Intelligence and Swarm Robotics

### 2.5.2.1 Swarm Intelligence

**Swarm intelligence** studies the collective behaviors emerging in a decentralized manner from local interactions between large numbers of (relatively simple) individuals (Bonabeau et al., 1999; Nouyan et al., 2009; Beni, 2005). This subfield of artificial intelligence draws inspiration from the collective behavior of social animals, specially insects. Some examples are: bee swarms, ant colonies, flocks of birds, and schools of fish (Blum and Groß, 2015). Many of these animals have quite limited individual abilities. However, they are able to carry out complex tasks as a group. These, group-level, behaviors emerge from simple interactions between simple individuals in a process of self-organization (Blum and Groß, 2015). Self-organization is, thus, one of the key focuses of swarm intelligence.

Synchronization of groups of individuals, in particular firefly synchronization, is a self-organizing behavior itself, where global order emerges from local interactions. Therefore, it is of interest in swarm intelligence (Zhang et al., 2014). The more abstract literature can be found in the fields of non-linear dynamics and complex systems in physics, and has been discussed in previous sections. Most of the work on firefly synchronization in the field of swarm intelligence itself can be found in the derived swarm robotics application domain (Trianni and Campo, 2015), as it will be discussed later. A related algorithm, known as the firefly algorithm, inspired by the manner fireflies flash to attract mates, has been extensively studied with applications to optimization (Fister et al., 2013).

#### 2.5.2.2 Swarm Robotics

**Swarm robotics** refers to the application of swarm intelligence techniques for the coordination of multi-robot systems (Şahin, 2004; Beni, 2005). Therefore, swarm robotics is concerned with the study of emergent self-organized behavior in groups of embodied agents (i.e. robots) (Brambilla et al., 2013). These robots typically: are autonomous; have limited sensing and communication capabilities, confined locally; do not have a centralized control or global knowledge; and cooperate to accomplish tasks (Dorigo et al., 2004; Brambilla et al., 2013). The emerging behaviors of the swarm are characterized by robustness to the loss of individuals, scalability to different group sizes and flexibility to cope with various environments.

Similar to sensor networks, timing synchronization can be necessary for the operation of robotic swarms, mainly to coordinate actions or achieve consensus in a decentralized manner (Parker, 2008; Ren et al., 2005). Certain modes of cooperative transport need synchronization to ensure that groups of robots push an object simultaneously (Parker, 2008; Vig and Adams, 2006). Temporal synchronization has also been used in distributed sensing and data gathering by teams of robots (Ranganathan et al., 2010; Winfield, 2000). For example, a swarm of robots is able to track a target that moves faster than any individual robot by synchronizing the timings of the robots' observations (Khaluf et al., 2011). In addition, synchronous robots are required in some solutions to the rendezvous problem (Cortés et al., 2006; Lin et al., 2007), which involves a group of agents meeting in a previously unspecified location without active communication.

#### 2.5.2.3 PCOs Synchronization in Swarm Robotics

Due to its simplicity and its decentralized nature firefly synchronization is well suited for robotic swarms. The most relevant approaches to swarm synchronization using PCOs are presented below, specially those where the mobility of the robots plays an important role.

Melhuish et al. (1999) used PCOs to regulate the size of a traveling robotic cluster. The oscillators used were not IFOs. Instead, they fired stochastically within a certain time interval. In addition, oscillators entered a refractory period after firing, during which firings from other robots are discarded. The robots are influenced by firings from other units only outside the refractory period . Upon receiving such a firing an oscillator reacted by immediately firing. The firing signal was an acoustic signal emitted radially. When a group of robots is synchronized, the number of robots forming that group can be estimated based on the strength of the signal. In addition, robots turn left or right in response to the strength of the sound at both their sides. In this a way, clusters of specific sizes can be formed.

Christensen et al. (2009) developed a swarm level fault detection algorithm based on the synchronization of PCOs, and implemented it on the swarm-bot robotic platform. In their work the agents interacted with others within a certain range, similar to Prignano et al. (2012); Wang et al. (2014) and Wang et al. (2015). Each of the robots possesses LEDs that can be detected by the onboard omnidirectional camera on the others robots. Through PCOs, robots attempt to flash in synchrony. Robots that failed can be identified as those that are not flashing in unison with the rest of the swarm. This algorithm was evaluated on swarms of up to 100 robots in simulation, and on 10 physical robots. This is a perfect example of mobile pulse-coupled oscillators, similar to those described in Sec. 2.4. It was found that the synchronization time decreased with the robot density. In addition, it was shown that moving robots synchronized faster than static ones. However, the effect of the speed of the robots and their radius of interaction was not investigated.

Perfect temporal synchronization could be, in some cases, unattainable, both in biological and technological scenarios. There might exist a delay between a leader and a follower who is trying to synchronize with it. Together with a limited sensor range, propagation waves might be generated as a by-product. Hartbauer and Römer (2007) exploited this effect, and proposed a control algorithm that allows robots to communicate the direction of a target by initiating and propagating such waves. The application of this technique was demonstrated in a cleaning task with I-Swarm micro-robots. Robots flashed their on-board LEDs to signal a firing, emitting light in four 60° cones. Upon perceiving this signal with a photo-receptor, a robot would modify its current phase with a delay-advance PRC as in Wang et al. (2012, 2014). Hartbauer and Römer (2007) analyzed the effect of speed on the completion of the task analyzed, but not its effect on the synchronization time.

Trianni and Nolfi (2009, 2010, 2011) utilized evolutionary approaches that would lead to swarm synchronization. In particular, these papers dealt with movement synchronization in an environment rather than temporal synchronization. Robots were rewarded if they alternated periodically between two locations in the environment and if they signaled to other robots in order to synchronize their motions. The evolved strategies resembled those found in fireflies (Buck, 1988). Earlier, Trianni and Nolfi (2007); Sperati et al. (2008) had performed similar studies on movement synchronization using artificial evolution. In this case, the desired behavior was for the robots to oscillate back and forth in proximity to a light source.

Similar to the above, Castillo-Cagigal et al. (2014) studied periodic oscillatory motion of a swarm between two areas of an environment. These two areas exhibit periodic temporal patterns, whereby a task is available for the robots in one or the other area. The authors designed an algorithm that allows the robots to maximize their time at the right location and, therefore, synchronize their movement with the changing environment. In addition, to speed up the process, pulse-coupled synchronization was also implemented among the robots. The utilized signal is the blink of a light that could be detected via a line-of-sight sensor on the robots. Interestingly, the PCO synchronization algorithm was modified to allow for near-synchronization instead of perfect synchronization, to avoid physical interference (e.g. path obstruction) among robots.

Wang et al. (2016), implemented an algorithm inspired by firefly synchronization to align the headings of robots in a swarm via exchange of short pulses. In this case, the orientation of the robot represented its phases, and synchronization equated to alignment. A similar approach had been presented earlier (Sepulchre et al., 2007), however it involved continuous sensing of all surrounding robots, instead of being pulse-based.

The works presented in the above three paragraphs focused on motion synchronization. Although PCOs were studied in moving oscillators, in those studies the position of the robot in an environment is coupled to its phase. Therefore, there is no effect of an independent spatiotemporal motion on a temporal-only dynamics as it is the typical case of MPCOs as presented in Sec. 2.4.

In an attempt to avoid interference among robots performing a foraging tasks, Wischmann et al. (2006) studied PCO synchronization using sound signals. In Wischmann and Pasemann (2006), firefly-like synchronization of periodic acoustic signals emerged spontaneously after evolving a swarm behavior that maximized foraging by communicating the presence of food between robots. In a similar foraging scenario, Chevallier et al. (2011) investigated synchronization among subpopulations of agents. It was observed that this reduced the chances of collisions or obstruction between robots.

Silva et al. (2015) studied PCO synchronization in a swarm of robots, analyzing also the effect of topological changes on the synchronization rate. Unlike MPCOs, oscillators (robots) here were static cellular automata and topological changes were modeled by randomly removing some of the connections. Therefore, this setup is not totally equivalent to MPCOs.

In contrast to the aforementioned works, Sutantyo and Levi (2013, 2015) used a variant of PCOs to achieve desynchronization in a swarm of underwater robots. As with sensor networks, desynchronization facilitates time division medium access (TDMA). This allows robots to communicate at different time slots without interfering with each other.

## 2.6 Summary

In this chapter, the basic principles of synchronization were presented. Special emphasis was given to phase oscillators, where only the phase (or current point along the oscillation cycle) needs to be synchronized. In particular, the chapter focused on oscillators interacting via pulses, or pulse-coupled oscillators. Subsequently, recent results in mobile pulse-coupled oscillators were described. In these, oscillators represent physical agents moving in an environment, in which agent mobility influences the network topology and, as a consequence, the synchronization rate. Previous works have reported that, depending on the topology and system parameters, different synchronization regimes appear. This thesis focuses on the study of the emerging dynamical regimes with a view towards applications in technical domains. In this respect, previous approaches to pulse-coupled oscillators networks and swarm robotic systems were recounted in this chapter.

# Chapter 3

# Towards Firefly Synchronization in Swarms of Mobile Agents[1]

The previous chapter showed the growing interest in synchronization of mobile pulse-coupled oscillators, both as an abstract non-linear dynamics physics problem and for its applications in technical domains, such as sensor networks and swarm robotics. In most works, agents interacting with the neighbors within a local range were considered. In this scenarios, synchronization occurs monotonically faster as agent speed increases, as shown in Figure 2.6 (solid line). However, Prignano et al. (2013) found a nonmonotonic dependence if agents interact only with their nearest neighbor, as shown in Figure 2.6 (dashed line). Depending on their speed, synchronization emerges as a slow process through spreading of the local coherence, as a fast process where global synchronization dominates, or it is inhibited for a range of intermediate speeds.

In this chapter we study a different kind of interaction, in which agents are influenced only by others in their cone of vision. This is a potentially more realistic scenario for technical applications than nearest neighbor interaction, given that, without knowledge of the position of every member of the population, a particular agent cannot know whether it is the nearest neighbor to another agent. Moreover, although omnidirectional range interactions are common in sensor networks or swarm

[1] This chapter has been adapted from Perez-Diaz et al. (2015).

robots, some sensing modes involve a detection cone, such as a camera or parabolic antennas. We build on the work by Prignano et al. (2013) and show that, with a cone of vision interaction, not only the speed of the agents, but also their angle and range of interaction can tune the appearance of this intermediate regime. Specifically, we show that by tuning the angle and range of vision, both the monotonic and the nonmonotonic behaviors described earlier can be obtained. These results are first demonstrated in abstract simulations, where agents are point-like particles. Subsequently, our findings are validated in a robotic simulator, where the effects of embodiment are taken into account.

This chapter is organized as follows. Section 3.1 introduces the methods used, including the oscillator and neighborhood models as well as the synchronization metric. Section 3.2 presents the experiments performed in a particle simulator and compares the results with those of Prignano et al. (2012, 2013). Section 3.3 shows the effects of embodiment and physical constraints in a robot simulator. Finally, Section 3.4 concludes the chapter.

## 3.1 Methods

We consider a population of $N$ agents moving with constant speed $V$ in a bounded square 2-D environment of side length $L$. Each agent possesses an associated internal oscillator. When an agent reaches the scene boundary it will randomly reorient its motion to a direction uniformly selected from the range $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ with respect to the wall's normal. This is in contrast to many previous approaches (Prignano et al., 2012, 2013; Wang et al., 2014, Wang et al., 2015), where periodical boundary conditions were considered (topologically equivalent to a torus). A bounded scenario was preferred here for realism and practical feasibility.

Two sets of simulations were realized. Firstly, we analyze an abstract model, where each oscillator behaves as a point-like particle. Secondly, we study the effects of physical constraints and embodiment on a robot simulator. The robot chosen for these simulations was the *e-puck* (Mondada et al., 2009).

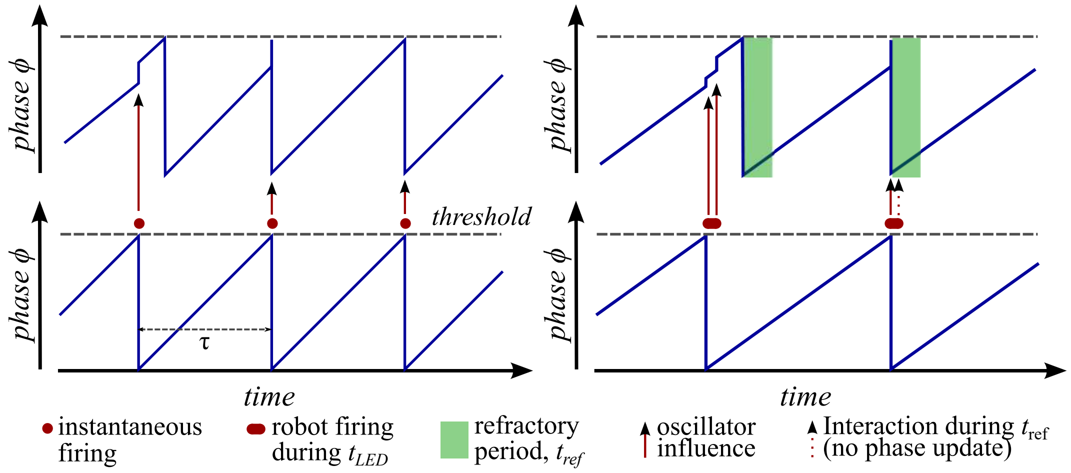


Figure 3.1: Dynamics of two oscillators, where the bottom one influences the top one. Left: Oscillator model and interaction as described in Sec. 3.1.1. Right: Oscillator as implemented in the robot simulations. During the refractory period (only shown for the upper oscillator) the oscillator is not influenced by any interaction. Outside the refractory period multiple consecutive interactions are allowed, as it will benefit synchronization.

### 3.1.1 Oscillator Model

The model of the internal oscillator of each agent, $i$, is a simple integrate and fire oscillator,

$$\frac{d\phi_i}{dt} = \frac{1}{\tau}, \tag{3.1}$$

where its phase, $\phi_i \in [0, 1]$, grows linearly in time with period $\tau$, until a threshold, $\phi_{thres} = 1$, is reached and a firing event occurs. Upon firing, the oscillator resets its phase to 0, and the phase of the neighbors, $\phi_n$, is updated multiplicatively by a factor $\epsilon$ (Christensen et al., 2009; Prignano et al., 2012, 2013), as follows (see Fig. 3.1 left):

$$\phi_i\left(t^-\right) = 1 \Rightarrow \begin{cases} \phi_i\left(t^+\right) = 0 \\ \phi_n\left(t^+\right) = (1+\epsilon)\,\phi_n\left(t^-\right). \end{cases} \tag{3.2}$$

### 3.1.2 Neighborhood Model

An agent, $A$, is considered neighbor of another, $B$, if and only if $B$ lies inside the **cone of vision** of $A$, that is, the circular sector centered in $A$, with radius $R$ and

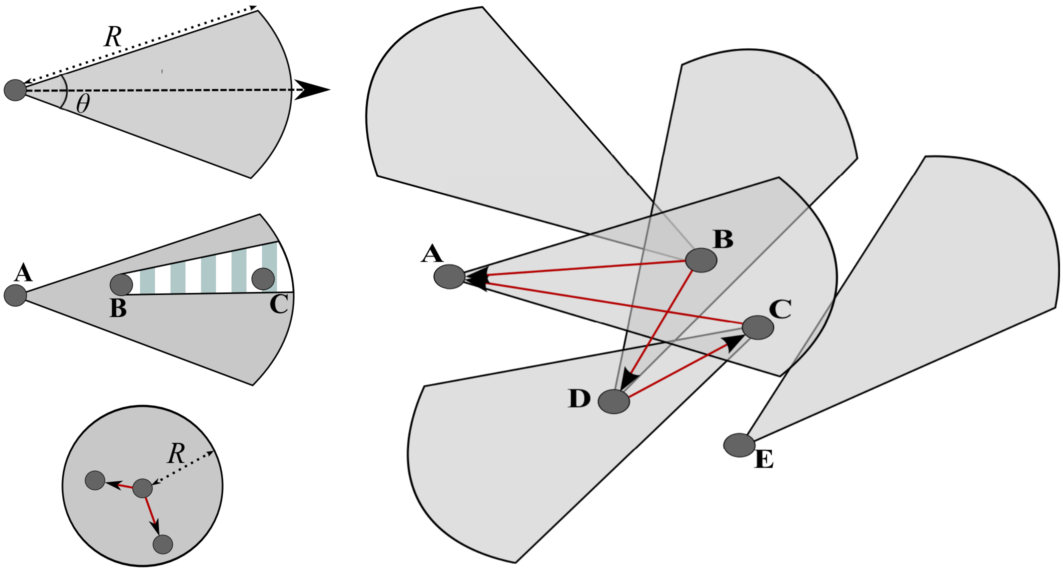


Figure 3.2: Top-left: Cone of interaction. Center-left: Occlusion in the robot simulation; $C$ cannot be seen from $A$. Bottom-left: Interaction range used in Prignano et al. (2012). The red arrows indicates the direction of the interaction. Right: Example of different interactions. $A$ is neighbor of $B$ and $C$; $B$ has two neighbors, $A$ and $D$; $A$ and $E$ have no neighbors and $E$ is not a neighbor of any other agent.

angle $\theta$, and oriented in the direction of motion of $A$ (see Fig. 3.2 top-left). If $A$ is a neighbor of $B$, then $B$ will influence $A$ when applying Eq. 3.2.

The bottom-left panel in Fig. 3.2 shows the frequently used *interaction range*. In that case, agents that lie within a certain distance, $R$, from another are considered its neighbors. By contrast, regardless of the shape of the interaction region, our *cone* assigns the neighborhood in the opposite direction. That is, if $B$ can be seen by $A$, then $A$ is $B$'s neighbor. This choice was made to emulate a natural scenario, where the flashing of a firefly would only be recognized by those insects that are currently seeing it. Note, however, that for $\theta = 360°$ this interaction becomes equivalent to the interaction range.

In contrast to the *nearest neighbor interaction* described in Prignano et al. (2013) where each oscillator has precisely 1 neighbor, in our model any given agent could have from 0 to potentially $N-1$ neighbors (see Fig. 3.2 right for examples of different interactions). In this aspect, this is similar to the *interaction range* scenario.

The dimensions of the cone of vision that were studied are shown in Table 3.1.

Note that any $R$ greater than $\sqrt{2}L$, the length of the diagonal, is equivalent to an infinite range.

Table 3.1: Dimensions of the Cone of Vision

| **Parameter** | **Value** |
|---|---|
| $R$ | $[0.05L, \infty)$ |
| $\theta$ | $[10°, 360°]$ |

### 3.1.3 Particle Simulations

The particle simulations were performed in an event-driven manner. The positions, phases and orientations of the oscillators were updated either when a wall was reached by a particle or when an oscillator reached the firing threshold. In this way we simulated pure continuous time and equation (3.1) is integrated exactly.

Given that some agents may fire upon receiving a phase update from a firing neighbor and, it turn, could elicit further firings, in the particle simulation this interaction occurs instantaneously in frozen time. This also implies that, if two firings are received at the exact same time, a single phase update will take place.

### 3.1.4 Robot Simulations

#### 3.1.4.1 Robot and Simulator

The robot simulations were performed using the open-source Enki simulation toolkit (Magnenat et al., 2007). Enki provides a faster than real time simulation of the physics and dynamics of colonies of robots, and it contains a built-in model of the e-puck (see Fig. 3.3). This robot weighs 152 g and its body is modeled as a differential-wheeled cylinder of diameter 7.4 cm. The distance between the wheels is 5.1 cm. Their speed can be set independently within the range $[-12.8, 12.8]$ cm/s. The length of the control cycle was set to $\Delta t = 0.1$ s and the physics was updated 100 times per second. The choice of simulator is supported by the work of Gauci et al. (2014a,b) where swarm controllers synthesized using Enki were successfully validated with up to 40 physical e-pucks.

The e-puck is equipped with eight short-range infra-red proximity sensors and

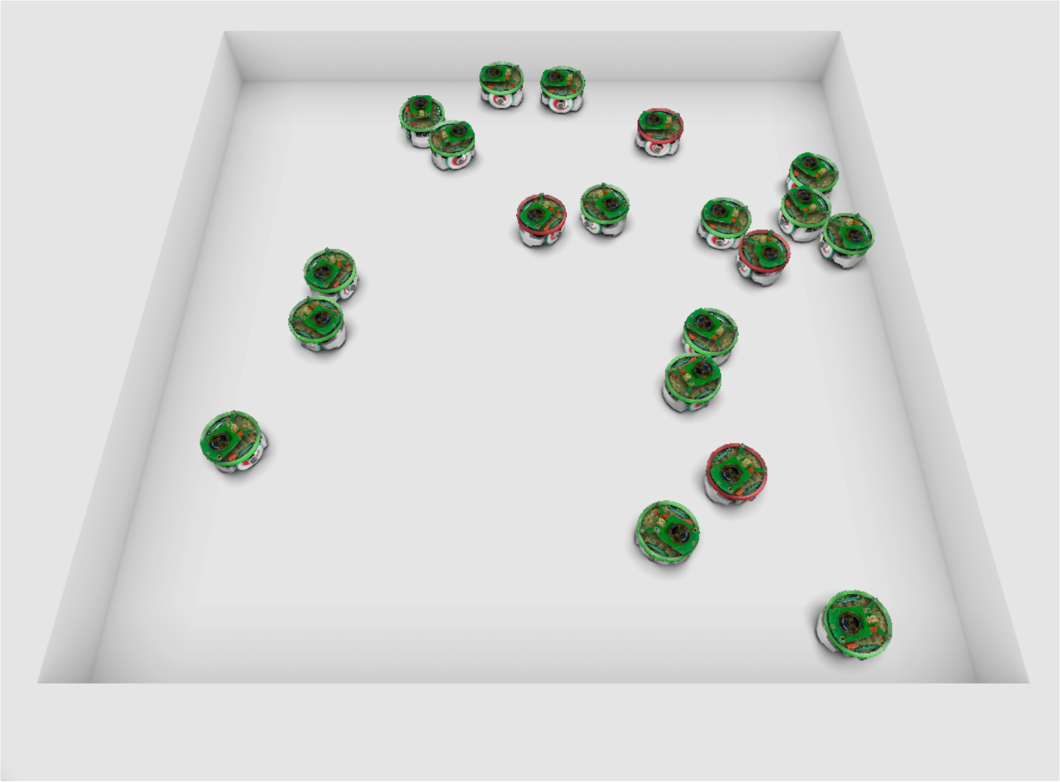


Figure 3.3: The robot environment in Enki showing 20 e-pucks in a 100 cm square arena. The robots with the red LED ring turned on are firing.

a proximity camera located at its front. The camera has a resolution of 640 (horizontal) by 480 (vertical) RGB pixels, with a corresponding 56° horizontal viewing angle. The full image cannot be processed or stored in the dsPIC of the robot. Nevertheless, a subsampled image can be acquired at 4 frames per second (Mondada et al., 2009).

The proximity sensors are used to implement a simple wall avoidance algorithm. When the distance to a wall is smaller than a certain value, the robot reorients itself as described at the beginning of Section 2. In order to keep the robot simulation as close as possible to the particle simulation, no explicit robot-robot collision avoidance mechanism was employed. We carried out preliminary tests with Enki and the physical e-pucks, and found that the circular shape of the robot allows two agents to come in contact and slide past each other.

#### 3.1.4.2 Oscillator

The phase of the robot's oscillator is updated according to the discrete version of (3.1),

$$\phi_i\left(t_n\right) = \frac{\Delta t}{\tau}\phi_i\left(t_{n-1}\right), \tag{3.3}$$

where $t_n$ represents the $n^{th}$ control step. Once the phase reaches the threshold and is reset to 0, the red LED ring on the contour of the robot is switched on and kept lit for a certain interval.

#### 3.1.4.3 Cone and Interaction

The agents take snapshots using their camera, which is pointed in the direction of motion. In Enki, the camera can be set to capture images of objects within a certain distance range and angle of view, which directly implements the neighborhood model. If a certain robot is firing, and thus having its LEDs on, it will be detected by the camera of its neighbors and lead to an update of their phases according to (3.2). However, in contrast to the particle simulation, in this case a direct line of sight between the two agents is also required. Therefore, the effects of occlusion by other robots is taken into account (see center-left panel in Fig. 3.2).

In the particle simulation the interaction between oscillators occurs instantaneously. In a real robot scenario that would be impossible. For a robot to observe a flash it must record and process the snapshot of its camera (Christensen et al., 2009). This process inevitably gives rise to delays. For this reason, the LEDs are kept on for a certain time, $t_{LED}$. Because of the necessary latency of the flashing signal and the delays in processing it, an oscillator could get displaced from synchrony if it detects more than one firing. To compensate for this effect, a refractory period, $t_{ref}$ was added immediately after each agent has fired. During this interval the oscillator is not influenced by any interaction (Kuramoto, 1991; Nymoen et al., 2013) (see Fig. 3.1 right). We have found experimentally that keeping the LEDs on for four control cycles, $t_{LED} = 4\Delta t$, while staying in the refractory period for eight cycles, $t_{ref} = 8\Delta t$ allowed the robots to achieve and sustain synchronization.

Even though the actual camera of the e-puck is directional and has a limited field of view (approx. 56°), we assumed that it can detect images of up to 360 degrees for the sake of completeness.

### 3.1.5 Synchronization Metric

In order to measure the level of synchrony of the system, a certain oscillator, $\phi_1$, is selected as reference, and upon its $k$th firing, at time $T_k$, we calculate the order parameter (Prignano et al., 2012, 2013), defined as follows,

$$\eta\left(T_k\right) = \frac{1}{N}\sum_{i=1}^{N} \cos\left(2\pi\phi_i\left(T_k\right)\right). \tag{3.4}$$

As it is calculated at the moment of firing of the reference oscillator, this function measures the average phase difference between the other oscillators and the reference. It increases monotonically with the degree of synchronization, from $\eta\left(T_k\right) = 0$ for a totally uniform phase distribution, to $\eta\left(T_k\right) = 1$ for complete synchronization.

The simulation is stopped once the order parameter arrives to a certain threshold, $\eta_{sync} \simeq 1$; in this case we consider the system to be synchronized. Moreover, we count the number of cycles, $k$, of the reference oscillator elapsed until synchronization. From this point, we will refer to this value as $T_{sync}$. This **synchronization time** is a good measure of how long it takes for a system to achieve coherence independently of the oscillation period.

For practical reasons, we halted any simulation where $T$ exceeded a certain censoring threshold, $T_{cens}$, before synchronization is achieved. This censoring is taken into account when calculating the mean of $T_{sync}$ over several repetitions (see Appendix A for further details).

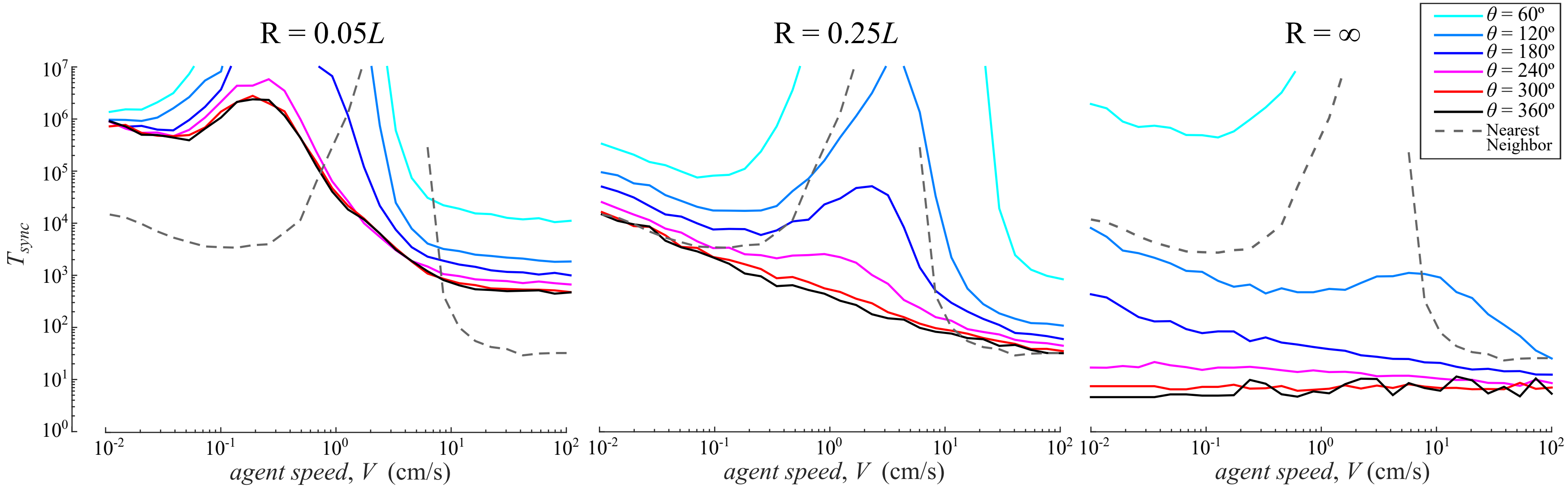


Figure 3.4: Log-log plot of $T_{sync}$ as a function of the speed of the oscillators for the particle simulation. The results for three ranges of interaction, $R$, are shown. Different color lines represent different angles of interaction. The dashed line represents the nearest neighbor interaction of Prignano et al. (2013) for the parameters of Table 3.2.

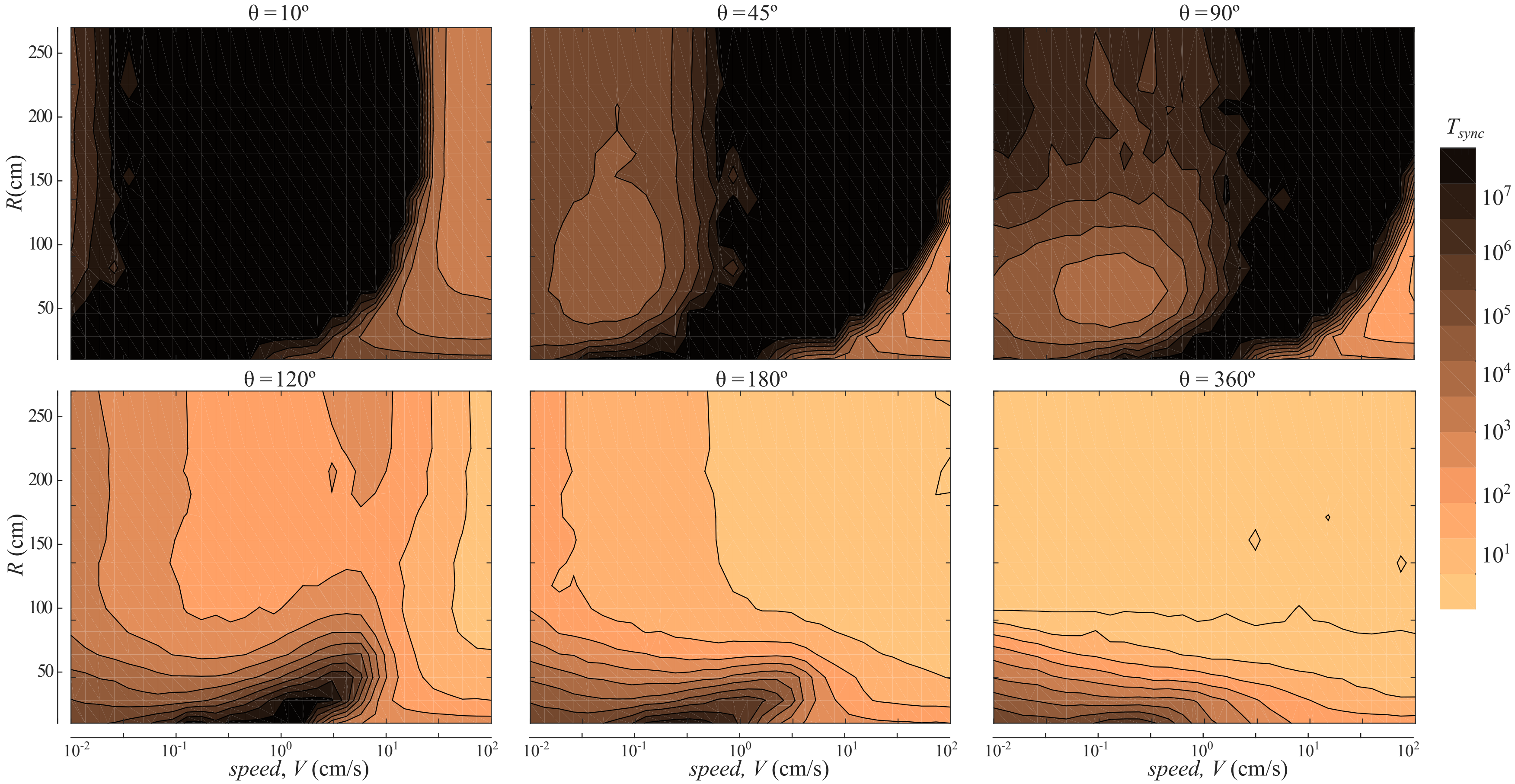


Figure 3.5: Contour plot showing the change in $T_{sync}$ as a function of the speed of the oscillators, $V$, and their ranges of interaction, $R$, for the particle simulation. The results for six different angles of vision are shown. $\theta = 360$ is equivalent to a range of interaction.

## 3.2 Particle Simulation Results

We performed the particle simulations with the parameters shown in Table 3.2. We studied the effect of varying the cone dimensions for 30 different speeds, $V$. We explored a set of 18 angles of view, $\theta$, and 14 radii, $R$, from the ranges in Table 3.1. For each combination of $V$, $\theta$, $R$ the simulations were repeated 100 times, where the initial phases, positions and orientations of the oscillators were randomly chosen.

Figure 3.4 shows the average synchronization time, $T_{sync}$, as a function of the speed of the oscillators, for several angles, $\theta$, and radii, $R$, of the cone of interaction. We obtained both the monotonic dependence characteristic of the range of interaction and the nonmonotonic relationship of the nearest neighbor interaction. For large angles and large cone radii $T_{sync}$ is a strictly decreasing function of $V$. However, as the size of the cone decreases (either $\theta$ or $R$ decrease), the monotonicity is broken, and an intermediate region appears where the synchronization is totally impeded. This intermediate regime varies from a small bump in the curve, to a drastic inhibition of synchronization, for relatively narrow and short interaction cones. Furthermore, we observe that, when present, the onset of the intermediate regime varies as a function of $R$. For comparison, the behavior for a nearest neighbor interaction, for the same setup is depicted with a dashed-line. The resemblance to our case becomes apparent in the second plot in Fig. 3.4 ($R = 0.25L$), where the intermediate regime coincides for the same range of speeds.

For further insight, Figure 3.5 displays the dependence of $T_{sync}$ as a function of the ranges of interaction for several angles of vision. Here we perceive three clearly distinct sectors corresponding to the three dynamical regimes. Narrow angles exhibit

Table 3.2: Parameters of the Particle Simulation

| **Parameter** | **Value** |
|---|---|
| $N$ | 20 |
| $L$ | 200 cm |
| $\tau$ | 1 s |
| $\epsilon$ | 0.1 |
| $V$ | $\left[10^{-2}, 10^{2}\right]$ cm/s |
| $\eta_{sync}$ | $1 - 10^{-6}$ |
| $T_{cens}$ | $10^{7}$ |

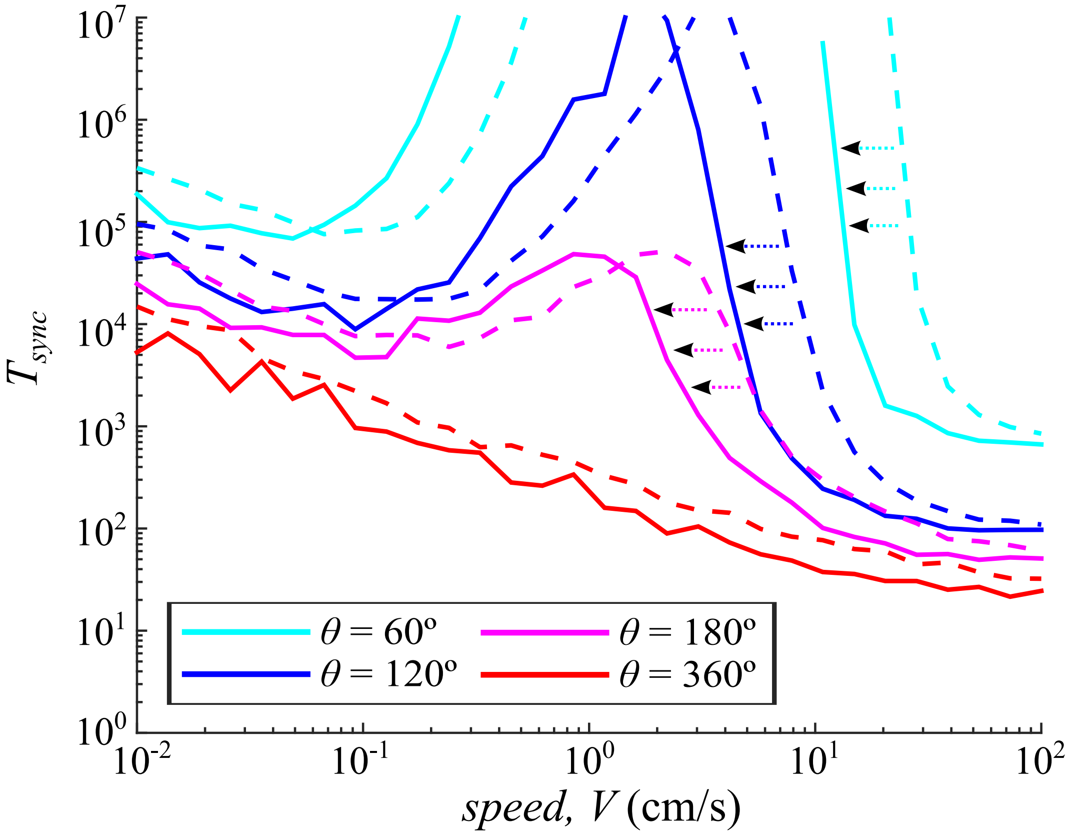


Figure 3.6: Effects of decreasing the environment size for $R = 0.25L$ in the particle simulation. The dashed lines represent the log-log plot of $T_{sync}$ as a function of $V$ for $L = 200\,\text{cm}$ whereas the solid lines depict the corresponding curves for $L = 100\,\text{cm}$.

a low degree of synchronization and the three areas are visibly separated. As the cone widens, the intermediate regime gradually disappears and gets relegated to the small radii region of the graph. Concurrently, a monotonic gradient of $T_{sync}$ as a function of $V$ appears and gradually spreads over all values of $R$. In the limit when $\theta \to 360°$, which is equivalent to a range of interaction, the synchronization time decreases monotonically with $V$ and $R$ as in Prignano et al. (2012).

The experiments corresponding to $R = 0.25$ were repeated for four additional environment sizes ($L = 50$, 100, 300 and 400 cm). In agreement with the results of Prignano et al. (2013), the location of the intermediate regime varies proportionally with the environment size. This relationship is not trivial, given that for a change of $L$ not only the relative speed, $V$, changes but also the density of oscillators. Figure 3.6 shows the shift in speeds of $T_{sync}$ from $L = 200\,\text{cm}$ to $L = 100\,\text{cm}$.

Table 3.3: Parameters of the Robot Simulation

| **Parameter** | **Value** |
|---|---|
| $N$ | 20 |
| $L$ | 200 cm |
| $\tau$ | 8 s |
| $\epsilon$ | 0.1 |
| $V^2$ | $\left[10^{-2}, 12.8\right]$ cm/s |
| $\eta_{sync}$ | 0.95 |
| $T_{cens}$ | $10^4$ |
| $\Delta t$ | 0.1 s |
| $t_{LED}$ | 0.4 s |
| $t_{ref}$ | 0.8 s |

## 3.3 Robot Simulation Results

We performed the robot simulations with the parameters shown in Table 3.3. For the initial experiments we used a scenario of the same size as in the particle simulation ($L = 200\,\text{cm}$). Nevertheless, the maximum speed of the e-pucks is 12.8 cm/s, which is significantly lower than the maximum investigated speed in the particle simulations (100 cm/s). We set $\tau = 8\,\text{s}$ in order to keep the number of oscillation cycles it takes for an agent to cover $L$ approximately equal in both simulations.

Conditioned by the fact that embodied simulations are several orders of magnitude more costly to perform than particle simulations, we set lower synchronization and censoring thresholds compared to the particle simulation. By applying Eq. 3.4 to a system with 20 agents, $\eta_{sync} = 0.95$ implies that the firing time of any individual is not shifted further than $0.01\tau$ from all other firings, which is equivalent to the threshold suggested in Christensen et al. (2009). $T_{cens} = 10^4$ with $\tau = 8\,\text{s}$ corresponds to 22 hours, much longer than what an e-puck can run continuously.

The effect of varying the cone dimensions was studied for 30 different speeds. We selected five different angles of interaction, $\theta$, and six interaction radii, $R$, from the ranges in Table 3.1. For every combination of $V$, $\theta$ and $R$ we performed 25 trials, where the initial phases, positions and orientations of the robots were randomly chosen. Additionally, we ensured that the random initial positions of the e-pucks were physically possible, i.e. no two individuals can occupy simultaneously the same space.

[2] Although the e-puck cannot move at 0.01 cm/s, this lower limit was set for completeness.

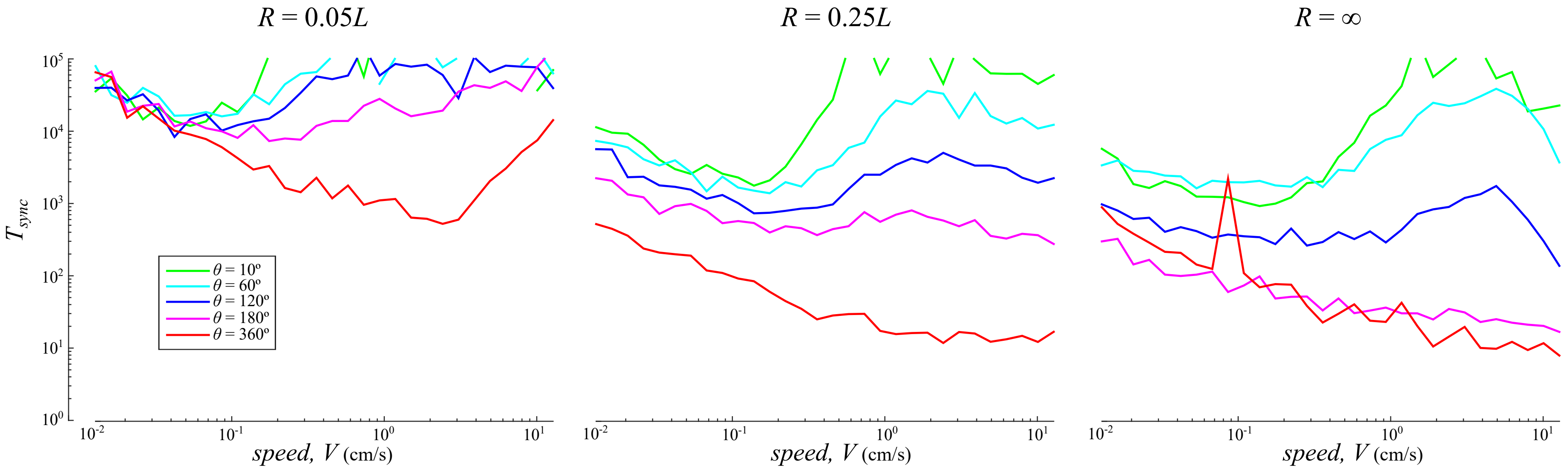


Figure 3.7: Log-log plot of $T_{sync}$ as a function of the speed of the oscillators for the robot simulation. The results for three ranges of interaction, $R$, are shown. Different color lines represent different angles of interaction.

The results of these experiments are shown in Figure 3.7. Despite the changes in implementation with respect to the particle simulation, the same essential behavior is found (cf. Fig. 3.4). The nonmonotonic relationship of $T_{sync}$ with respect to $V$ is observed. For cones of vision of increasing dimensions, a single monotonic regime dominates. However, in the case of embodied robot simulations, we observe a stronger prevalence of the intermediate regime than in the particle simulation. For instance, for $R = \infty$ and $\theta = 60°$ we still observe an inhibition of the synchronization in the robot simulation, whereas in the particle simulation this is almost completely smoothed out.

The simulations corresponding to $R = 0.25L$ were repeated for two smaller environment sizes ($L = 100$ and 150 cm). In contrast to the particle simulation, we observe that the inhibitory regime disappears for $L = 100\,\text{cm}$ for some of the studied angles (Fig. 3.8), instead of the shift in speeds previously obtained. We ascribe this difference to the effect of occlusion. For large environments, i.e. low robot densities, the effect of the physical size of the agents can be neglected. However, for an environment densely populated with robots, occlusion will occur more frequently. This will, inevitably, influence the rate at which agents change neighbors.

## 3.4 Discussion

This chapter presented a new case study on the emergence and tuning of synchronization in mobile pusle-coupled oscillators and it has showed a possible implementation on a swarm of robots.

Prignano et al. (2013) hypothesized that the occurrence of the intermediate regime when agents influence only their nearest neighbors might be related to the rate of change of neighbors. Our results support this hypothesis, given that a smaller cone of interaction (both in angle and range) implies a higher frequency of neighborhood change. Moreover, for very wide angles, our results approach those of Prignano et al. (2012), where all agents within a certain distance from a firing agent are influenced. The dimensions of the cone of vision allow to connect both cases,

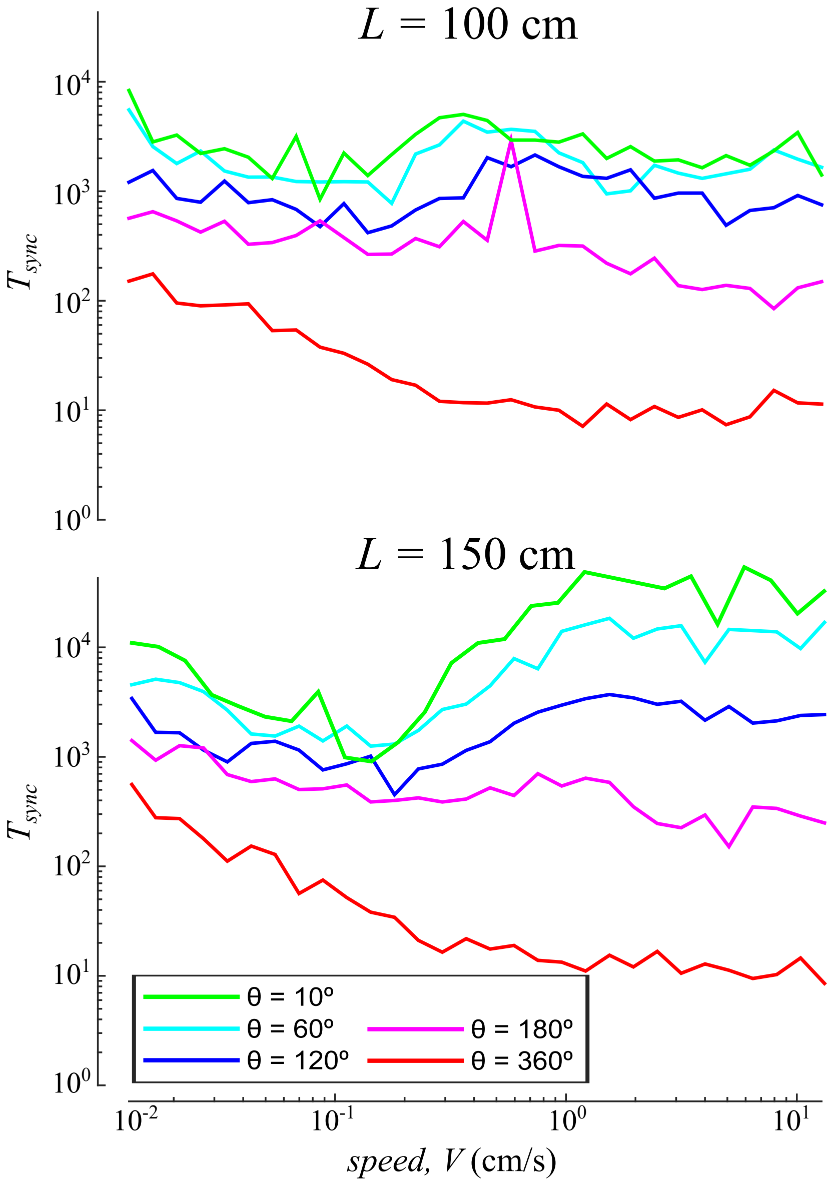


Figure 3.8: Effects of decreasing the environment size for $R = 0.25L$ in the robot simulation. The solid lines represent the log-log plot of $T_{sync}$ as a function of $V$ for $L = 100\,\text{cm}$ (top) and $L = 150\,\text{cm}$ (bottom).

thus yielding a natural extension of the aforementioned work. We have shown that global synchrony can be tuned both with the speed of the agents and the characteristics of their interaction. Further analysis is needed to understand the underlying mechanism that governs the dynamical regimes of the system.

The work of Prignano et al. (2013) suggested a possible discontinuity in the intermediate regime, where the synchronization is completely inhibited (i.e. $T_{sync} = \infty$) with a nearest neighbor interaction. In our case, the synchronization time is computationally incalculable for small cones of vision. Nevertheless, we observe a gradual change of the behavior for the intermediate range of speeds, from a slight impediment to a large inhibition of the synchronization. This could indicate that such a discontinuity might not exist in the present case until the size of the cone becomes infinitely small. However, this hypothesis remains unproven. In practice, the intermediate regime acts as a true boundary to synchronization.

It is worth noting that, to the best of our knowledge, this nonmonotonic dependence has only been observed elsewhere with a phase update such as the one described in Eq. 3.2 and with agents influencing their nearest neighbor (Prignano et al., 2013). Chapter 5 discusses how the type of neighborhood and the phase update rule can affect this emergent behavior.

In addition, an implementation on a swarm of e-pucks is simulated. For a sufficiently large environment, this yielded similar results to the particle simulation despite the physical constraints of the system, such as occlusion, collisions between agents and a non-instantaneous interaction. However, for a small environment, where the robots would be closer to each other, the effects of embodiment (i.e. occlusion, collision) become apparent. The intermediate regime is no longer present for some configurations.

The cone of vision was chosen due to its applicability in a robotics scenario. We demonstrated that the synchronization of the system can be inhibited under certain conditions. This becomes relevant when applying algorithms that rely on synchronization to robots that interact with each other using an interaction akin to the cone of vision. An example would be to apply the fault-detection method proposed by Christensen et al. (2009) on robots with a directional camera, whereas it was originally implemented on robots with an omnidirectional camera. Our findings suggest that depending on the dimensions of their field of view, the speed of the agents would affect the performance of this algorithm.

# Chapter 4

# Emergence and Inhibition of Synchronization in Robot Swarms[1]

The previous chapter described how different dynamical regimes can emerge in firefly synchronization of a swarm of mobile agents interacting with each other through a cone of vision. The presented results are indicative of the effects it can cause on practical applications in a real swarm of robots or a sensor network. As explained in Sec. 2.5.1 synchronization or the lack thereof could be beneficial depending on the desired use. Most of the work presented in Chapter 3 refers to simulation of dimensionless agents. An initial validation with a robot simulation was also performed. However, realism was lacking as robots performed an instantaneous reorientation when approaching the scenario's walls. In addition, collision avoidance was not executed. Instead, robots slided past each other when they enter in contact. Although this is possible with the used e-puck robots due to their circular shape, this type of contact between robots may be detrimental in other cases.

This chapter extends our previous work and focuses on embodied agents that signal their pulses visually using LED lights and are influenced only by others in their cone of vision. We study the parameters that influence synchronization in a

[1] This chapter has been adapted from Perez-Diaz et al. (2016a).

simulated robotic environment. In this occasion realism was a priority in our experiments. Firstly, number of pixels instead of camera angle was used to manipulate the dimensions of the cone of vision. And secondly, obstacle avoidance and realistic reorientation at walls were implemented using the robots' infrared sensors. Finally, we present a validation of the simulation results with physical robot experiments. In the robotic implementation the processing delays are also taken into consideration. The presence of the three dynamical regimes as a function of robot speed is observed, confirming experimentally our previous results.

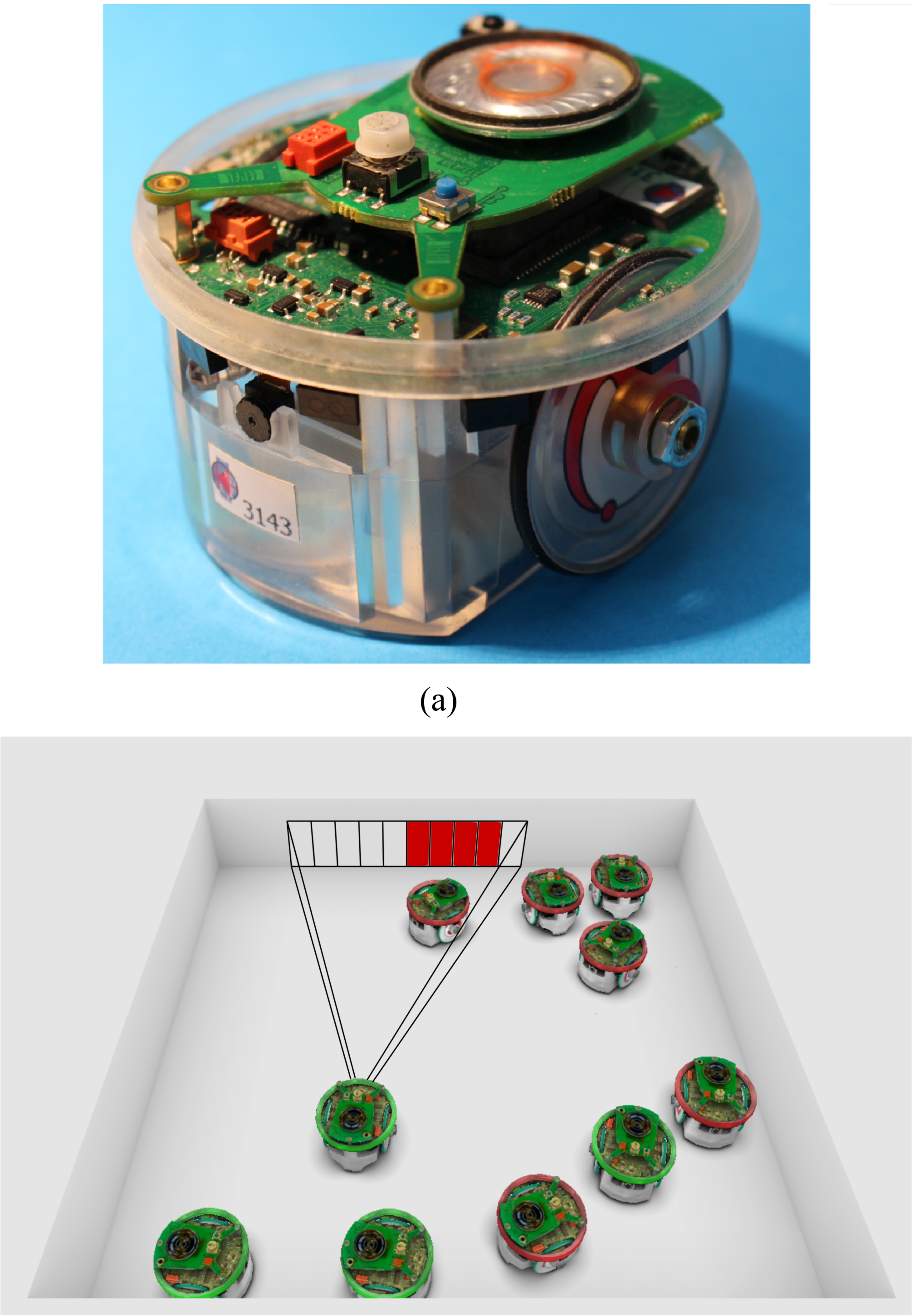
This chapter is structured as follows. Section 4.1 introduces the methodology, with an overview of the system as well as a description of the agents' temporal and spatial dynamics. Section 4.2 presents the simulation setup and results, with an analysis of the effect of the parameters involved. Section 4.3 describes the robotic trials and confirms experimentally the simulated results. Finally, Section 4.4 concludes the chapter.

## 4.1 Methods

We consider a group of $N$ robots moving in a walled square arena of length $L$. Every robot possesses an associated internal oscillator. The completion of a robot's oscillation cycle is signalled visually, and can be detected by neighboring robots (see details below). Thereby, the robots can influence each other in an attempt to synchronize their oscillators.

### 4.1.1 Robotic and Simulation Platforms

The robot used is the e-puck (Mondada et al., 2009), which is a differential-wheeled cylindrical robot of 7.4 cm in diameter (Fig. 4.1(a)). The distance between the wheels is 5.1 cm, and their speed can be set independently within the range [-12.8, 12.8] cm/s. The e-puck is equipped with a ring of red LED lights, eight short-range infra-red proximity sensors and a camera located at its front. The camera has a 56° horizontal viewing angle, with a resolution of 640 × 480 RGB pixels. Each robot



(a)

(b)

Figure 4.1: (a) An e-puck robot. (b) Simulation setup in the Enki simulator. The overlaid drawing illustrates an e-puck's field of view, where a few pixels have detected a flashing robot (note that the actual amount of pixels in Enki is 60).

**Algorithm 1** Oscillator dynamics

```
Require: $\phi \leftarrow \text{random}\,(0,\tau)$                    ▷ Initialize $\phi$ randomly within $[0,\tau)$
Require: $\phi_{\text{ref}} \in [0,\tau) \wedge \phi_{LED} \in [0,\tau) \wedge \phi_{ref} \geq \phi_{LED}$
Require: $\epsilon > 0$
 1: procedure OSCILLATOR
 2:     $\phi \leftarrow \phi + \Delta t$                                ▷ Linear increase of phase
 3:     if $\phi > \phi_{ref}$ then                      ▷ If phase is outside the refractory period
 4:         if wasFlashDetected then
 5:             $\phi \leftarrow \text{PHASEUPDATE}(\phi) = (1+\epsilon)\phi$
 6:         end if
 7:     end if
 8:     if $\phi \geq \tau$ then                          ▷ If a full oscillation cycle is complete
 9:         $\phi \leftarrow 0$
10:         turnLEDsOn();
11:     end if
12:     if $\phi > \phi_{LED}$ then                      ▷ If the cycle is outside the flashing period
13:         turnLEDsOff();
14:     end if
15: end procedure
```

signals the completion of a cycle of its internal oscillator by activating its red LED ring for a short duration. This, in turn, can be detected by another robot with its camera if the flash is in its field of view (cone of vision).

The robot simulation was performed using the open-source Enki library (Magnenat et al., 2007), see Fig. 4.1(b). Enki provides a faster than real time 2-D simulation of the physics and dynamics of groups of robots, and it contains a built-in model of the e-puck. In Enki the camera capture is a single row of 60 pixels spanning the field of view of the e-puck.

### 4.1.2 Oscillator Dynamics

Algorithm 1 presents the dynamics of each agent's internal oscillator. The dynamics is described by the value of its phase $\phi \in [0,\tau] \cap \mathbb{N}$, which is initialized at random[2]. The procedure OSCILLATOR is executed every control cycle ($\Delta t \ll 1\,\text{s}$). At the beginning of each cycle $\phi$ is advanced by $\Delta t$ (Line 2). When the threshold $\phi = \tau$ is reached a firing event occurs (Lines 8-11). At that moment the oscillator starts a new cycle by resetting its phase to 0, and turns on its red LEDs for a certain period

[2] All random numbers used in this chapter are generated using uniform distributions.

**Algorithm 2** Modified phase update function to compensate for camera delay

```
Require: $\mathit{\Delta} \geq 0$, $\mathit{\Delta} \ll \phi_{LED}$                ▷ Average camera delay
Require: $\phi_{\text{ref}} \in [0, \tau)$
Require: $\epsilon > 0$
 1: function PHASEUPDATE′($\phi$)
 2:     $\phi \leftarrow (\phi - \mathit{\Delta}) \bmod \tau$                   ▷ Phase at estimated time of flash
 3:     if $\phi > \phi_{ref}$ then  ▷ If phase at time of flash is outside the refractory period
 4:         $\phi \leftarrow (1 + \epsilon)\phi$
 5:     end if
 6:     if $\phi > \tau$ then           ▷ If this would have triggered a new oscillation cycle
 7:         $\phi \leftarrow 0$
 8:         turnLEDsOn();
 9:     end if
10:     $\phi \leftarrow \phi + \mathit{\Delta}$                                ▷ Calculate current phase
11:     return $\phi$
12: end function
```

of time to signal the firing to neighboring agents. The active LEDs of an agent can be perceived by the camera of another agent. In that case, the later would update its phase multiplicatively by a factor $\epsilon$ (Prignano et al., 2012, 2013; Christensen et al., 2009; Perez-Diaz et al., 2015) (Lines 4-6),

$$\phi \leftarrow (1 + \epsilon)\phi \,. \tag{4.1}$$

Note that, contrary to the previous chapter, here $\phi$ is restricted to the naturals instead of the interval [0, 1] $\in \mathbb{R}$. The e-puck lacks a floating-point unit, therefore all stored and calculated values must be integers. This not only has an effect on the precision of $\phi$ but also results in rounding in Eq. 4.1.

The LEDs of a robot need to be active for a sufficient amount of time, $\phi \in [0, \phi_{LED}]$ (Lines 12-14) to ensure that the cameras of other robots can detect it. In addition, because of the necessary duration of the flashing signal and the delays in processing it, an oscillator could get displaced from synchrony if it detected a firing shortly after starting a new cycle. To compensate for this effect, a refractory period, $\phi \in [0, \phi_{ref}]$ was added during which an oscillator is not influenced by any other (Kuramoto, 1991) (Line 3).

We found empirically that accounting for the average delay due to the frame rate ($\mathit{\Delta} = \frac{1}{2} framerate^{-1} \ll \phi_{LED}$) yielded a more stable synchronization in the real e-puck implementation. The function PHASEUPDATE in Alg. 1 is replaced by PHASEUPDATE′, defined in Alg. 2. The basic principle is to apply the phase update due to an LED flash detected at time $t$ to the phase at time $t - \varDelta$, as if it had been perceived instantaneously (Line 2). Subsequently the phase is advanced by $\varDelta$ to obtain the current value (Line 10). Note that setting $\varDelta = 0$ in Alg. 2 yields the original behavior described in Alg. 1.

### 4.1.3 Motion Controller

Each robot moves in a straight line with speed $V$ while there are no obstacles blocking its way. A collision avoidance algorithm is used to avoid running into walls or other robots. This algorithm implements a small turn if a robot is detected and a random reorientation upon encounter with a wall. The distinction between other robots and walls is determined heuristically by the time spent near the obstacle (see Appendix B for code and detailed description).

## 4.2 Simulation

### 4.2.1 Setup

Simulations were performed for a range of values of the system parameters: size of the environment $L$, number of robots $N$, oscillator period $\tau$ and phase update factor $\epsilon$. The LED and refractory periods were fixed to $\phi_{LED} = 0.075\,\tau$ and $\phi_{ref} = 0.15\,\tau$ respectively for all simulations. In addition, the effects of the dimensions of the cone of vision were studied by considering only certain fractions of the total camera pixels: from 4 to 60 pixels centered at the middle of the row. All the combinations of parameters were simulated in 200 trials for a range of 20 agent speeds $V$, from near stop to the maximum e-puck speed. In all trials the initial positions, orientations and phases of each robot were set at random.

In order to measure the level of synchrony, a certain robot is selected as reference. At the time of its $k$th firing, $T_k$, the complex order parameter is calculated as follows

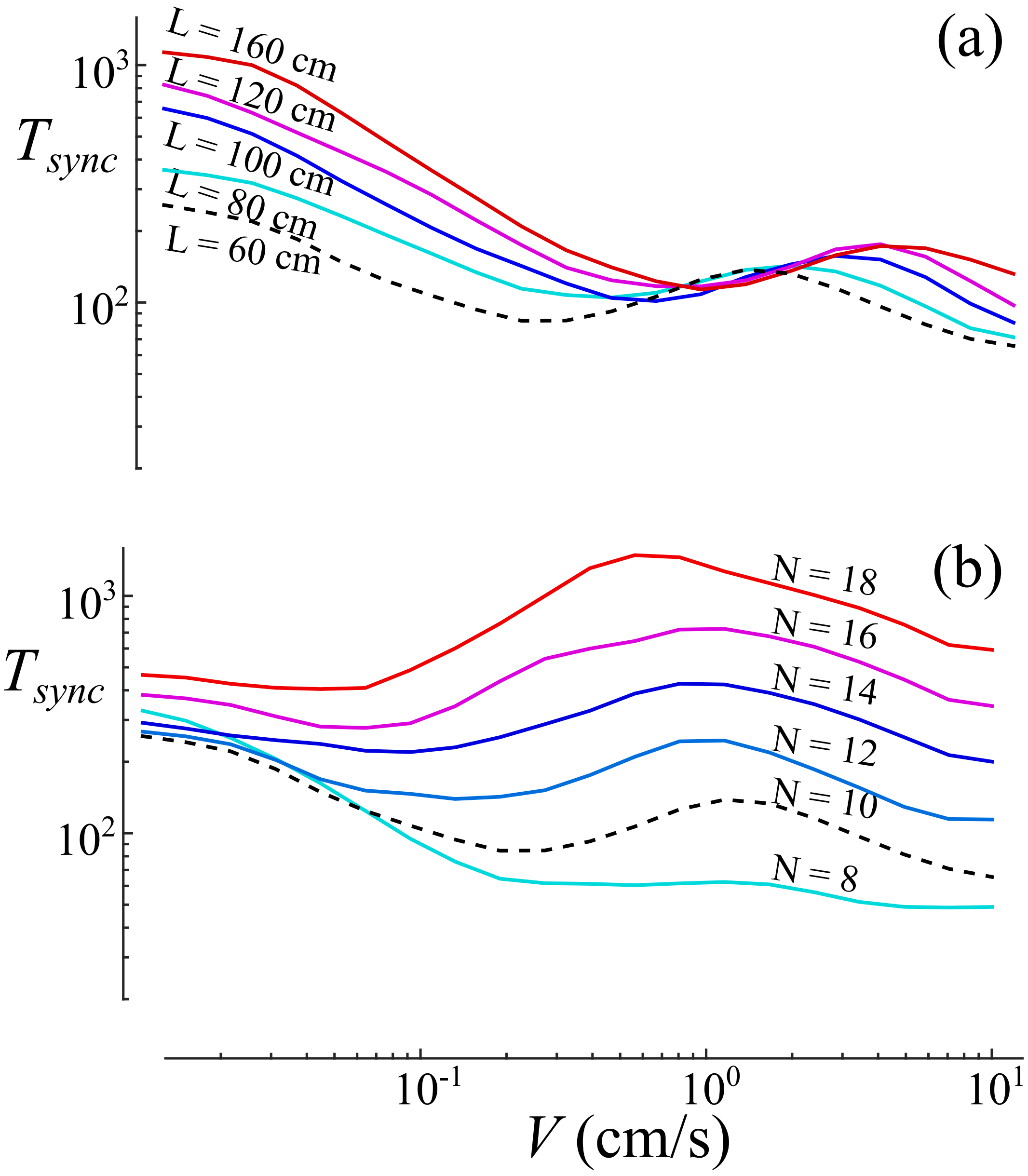


Figure 4.2: Synchronization time $T_{sync}$ (in number of cycles) as a function of agent speed $V$ for a variety of parameters. (a) Varying the arena size $L$ while fixing $N = 10$, $\tau = 5$ s and $\epsilon = 0.1$. (b) Varying the number of agents $N$ while fixing $L = 60$ cm, $\tau = 5$ s and $\epsilon = 0.1$. In all cases $\phi_{LED} = 0.075\,\tau$, $\phi_{ref} = 0.15\,\tau$ and the full image was considered (i.e. 60 pixels). The dashed lines in this figure and Fig. 4.3 correspond to the exact same parameter configuration. Points $(V_m, T_m)$ and $(V_p, T_p)$ denote the synchronization speed and time for the local minimum and the peak in the intermediate regime.

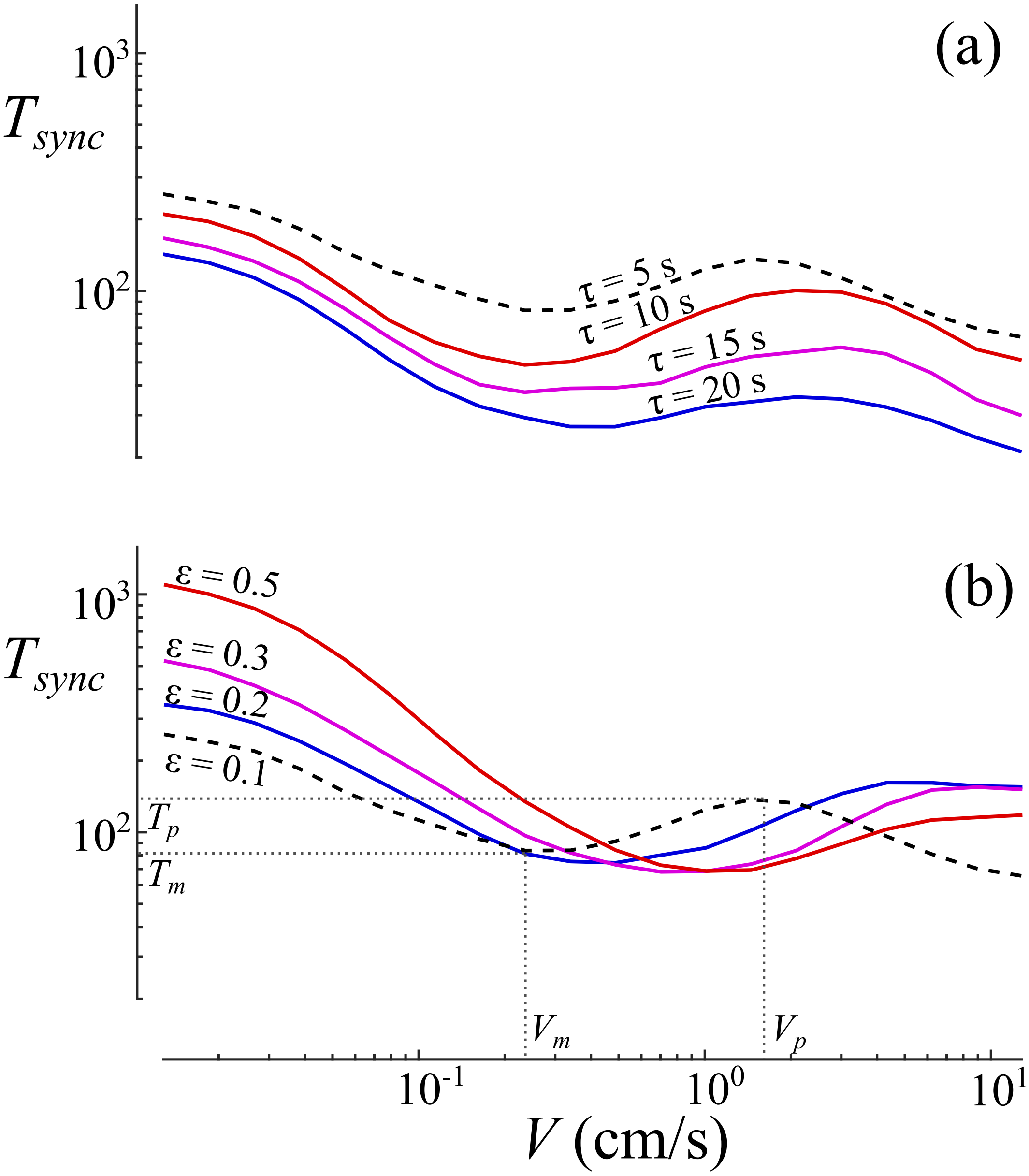


Figure 4.3: Synchronization time $T_{sync}$ (in number of cycles) as a function of agent speed $V$ for a variety of parameters. (a) Varying the oscillation period $\tau$ while fixing $L = 60$ cm, $N = 10$ and $\epsilon = 0.1$. (b) Varying the update factor $\epsilon$ while fixing $L = 60$ cm, $N = 10$ and $\tau = 5$ s. In all cases $\phi_{LED} = 0.075\,\tau$, $\phi_{ref} = 0.15\,\tau$ and the full image was considered (i.e. 60 pixels). The dashed lines in this figure and Fig. 4.2 correspond to the exact same parameter configuration. Points $(V_m, T_m)$ and $(V_p, T_p)$ denote the synchronization speed and time for the local minimum and the peak in the intermediate regime.

(Arenas et al., 2008),

$$r(T_k)e^{\mathrm{i}\frac{2\pi\phi(T_k)}{\tau}} = \frac{1}{N}\sum_{j=1}^{N} e^{\mathrm{i}\frac{2\pi\phi_j(T_k)}{\tau}}, \tag{4.2}$$

where $\phi(T_k)$ is the mean phase and modulus $r(T_k)$ measures the level of synchrony, from $r(T_k) = 0$ for a totally incoherent system, to $r(T_k) = 1$ for complete synchronization. This metric is related, although not exactly equal to Eq. 3.4, used in the previous chapter. The simulations were stopped once $r(T_k)$ reaches a threshold $r_{sync}$, set here to $r_{sync} = 0.95$ as in the previous chapter. We have observed that from this point synchronization becomes stable[3]. The number of cycles $k$ to synchronization is recorded. We refer to this value as the synchronization time $T_{sync}$. This measure represents how long it takes the system to synchronize independently of the oscillation period.

### 4.2.2 Results

Figures 4.2 and 4.3 show the synchronization time $T_{sync}$ for a variety of parameters. We observe three clearly distinct regimes. For slow agent speed the system takes long to synchronize and $T_{sync}$ decreases with $V$. For high enough speeds synchronization occurs significantly faster and $T_{sync}$ also decreases with $V$. For an intermediate range of speeds the monotonically decreasing dependence observed in the other two regimes is broken, observing a local maximum. The precise position and strength of this peak, as well as the local minimum that precedes it, depend on the oscillator parameters.

We performed a least-square fitting of $V_p$ and $V_m$, the speeds at which the intermediate regime peak and the preceding minimum occur respectively (Fig. 4.3 (a)), with respect to each parameter and found that,

$$V_m \propto \frac{\epsilon L}{N^{3/2}} \quad \text{and} \quad V_p \propto \frac{\epsilon L}{N^{1/2}},$$

[3] We consider the synchronization to be stable if the system continues in coherence ($r(T_k) \gtrsim r_{sync}$) from that point in time onwards.

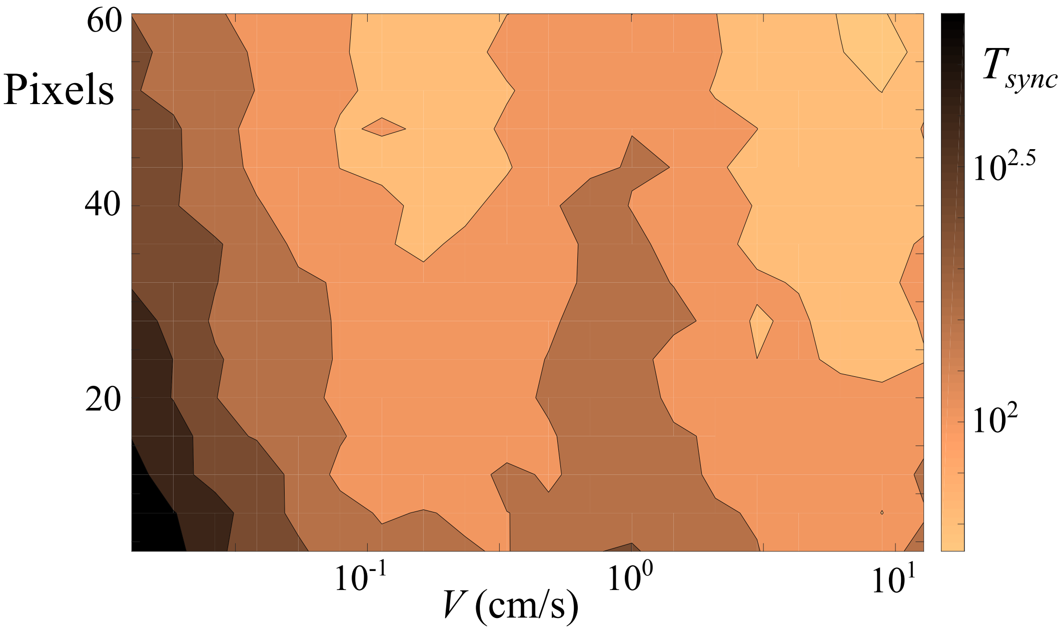


Figure 4.4: Time $T_{sync}$ (in number of cycles) required to synchronize 10 agents moving at speed $V$ in a square arena of side 60 cm while observing different fractions of the total camera image. The oscillator parameters are the same as for the dashed lines in Fig. 4.2 and 4.3: $\tau = 5\,\mathrm{s}$, $\epsilon = 0.1$, $\phi_{LED} = 0.075\,\tau$ and $\phi_{ref} = 0.15\,\tau$.

which agrees with the analysis by Prignano et al. (2013) on point-like agents influencing their nearest neighbor. No clear dependence of either point with $\tau$ was found. In addition, we found the following relationships for the corresponding synchronization times, $T_p$ and $T_m$, for both points,

$$T_{p,m} \propto \frac{Le^N}{\tau^2} + K_{p,m} \,,$$

where $K_{p,m}$ are some constant offsets. No evident dependence with $\epsilon$ was found.

Figure 4.4 shows the effect of changing the dimensions of the cone of vision by considering different amounts of pixels as described in the previous section. The three synchronization regimes can be clearly identified over the whole considered range of pixels. In addition, we observe that the synchronization time increases for all speeds as the number of pixels is reduced (narrower cone of vision).

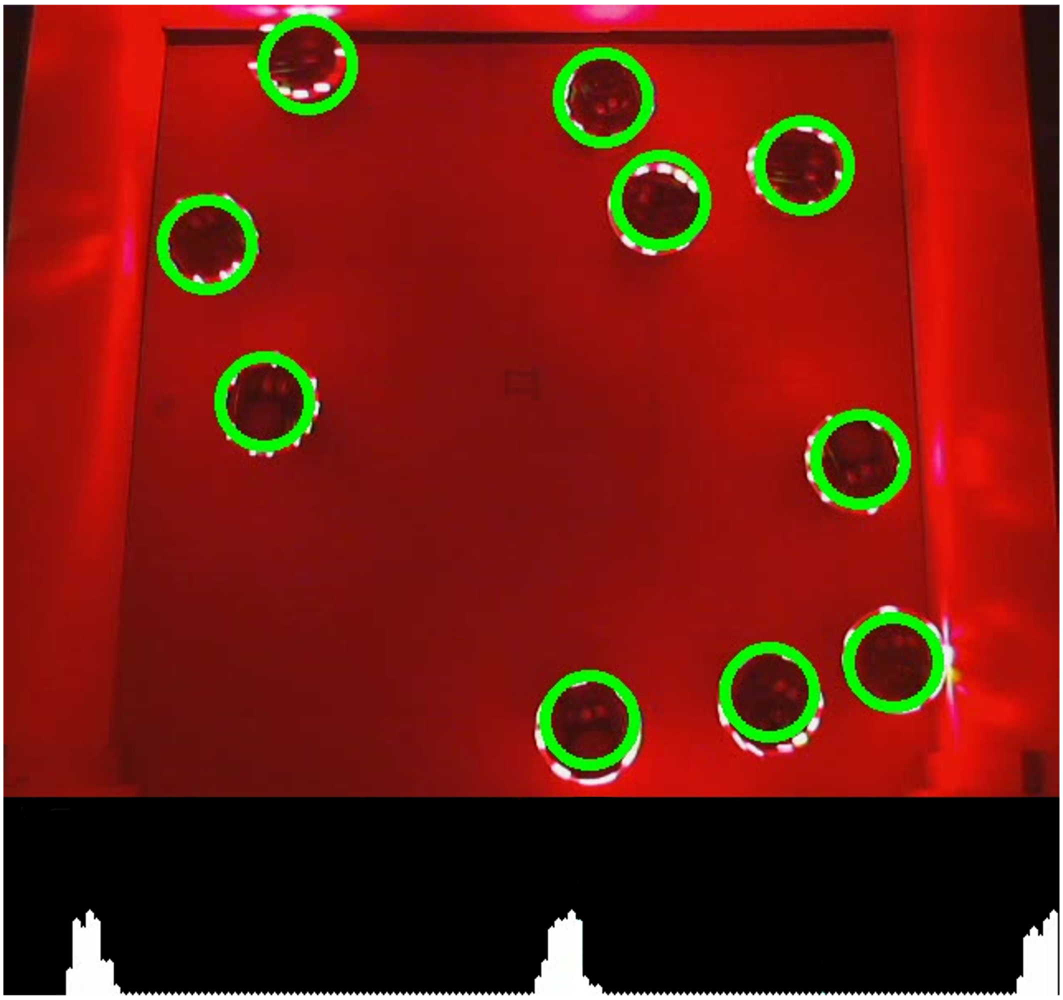


Figure 4.5: Snapshot of the experimental setup with 10 physical e-puck robots in a square arena of side length $L = 60\,\mathrm{cm}$. The active LED rings are detected and superimposed as green circles. The lower strip shows an evolving histogram of the number of LED rings detected over time. The system is considered to be synchronized if the distribution of the histogram around the current peak has a standard deviation smaller than two frames.

## 4.3 Physical Implementation and Experiments

### 4.3.1 Setup

Physical experiments were run in parallel with two groups of 10 e-pucks[4] in one of two adjacent $60 \times 60$ cm$^2$ white-walled arenas. Five trials for 10 different robot speeds were performed, starting with random initial phases until the system achieved global synchrony (see below for details). The robots' initial positions were randomly

[4] Hardware revision HWRev 1.2. http://www.gctronic.com/doc/index.php/E-Puck

selected from a 5×5 grid with 10 cm spacing between points and equal padding to the walls. The initial orientations were randomly selected at 45° degree intervals. The experiments were performed in total darkness, and the robots only started moving once the room lights were switched off. In a few trials, a robot stopped moving and was unable to recover, or the battery ran out and accidentally restarted (resetting its phase). In such cases, the trial was discarded and repeated.

The software was implemented utilizing functions of OpenSwarm, an embedded operating system running directly on each e-puck (Trenkwalder et al., 2016). The LED detection was performed at approximately 8 frames per second (FPS) by analyzing a single row of the acquired camera image (20 pixels with a 32× digital zoom). If two or more pixels were identified as red, a firing was considered to be detected. In addition, the LED indicating a low battery voltage was covered with tape to prevent false-positive flash detections.

A web camera was positioned at 110 cm above each arena and was used to record the trials and to measure the level of synchrony in real time. For this purpose, tracking software was developed using OpenCV that counted the number of detected red LED rings at each frame. The system was considered to be synchronized when the standard deviation of the ring counts over time was smaller than two frames, which corresponds to approximately 130 ms with our setup and ensured that synchronization was stable thenceforth (see Fig. 4.6). This threshold approximately corresponds to $r_{sync} = 0.95$ in accordance with the simulation setup (Sec. 4.2.1) (see Appendix C). The time required to achieve synchrony, $T_{sync}$, measured in number of cycles, is recorded. Figure 4.5 shows a snapshot of one experiment with the detected LED rings and a histogram of the ring count over time.

### 4.3.2 Results

Figure 4.7 shows the synchronization time $T_{sync}$ (in number of cycles) required to synchronize the group of 10 e-pucks as a function of speed. The obtained curve qualitatively displays the same behavior as in simulation. The slow, intermediate and fast regimes can be clearly identified.

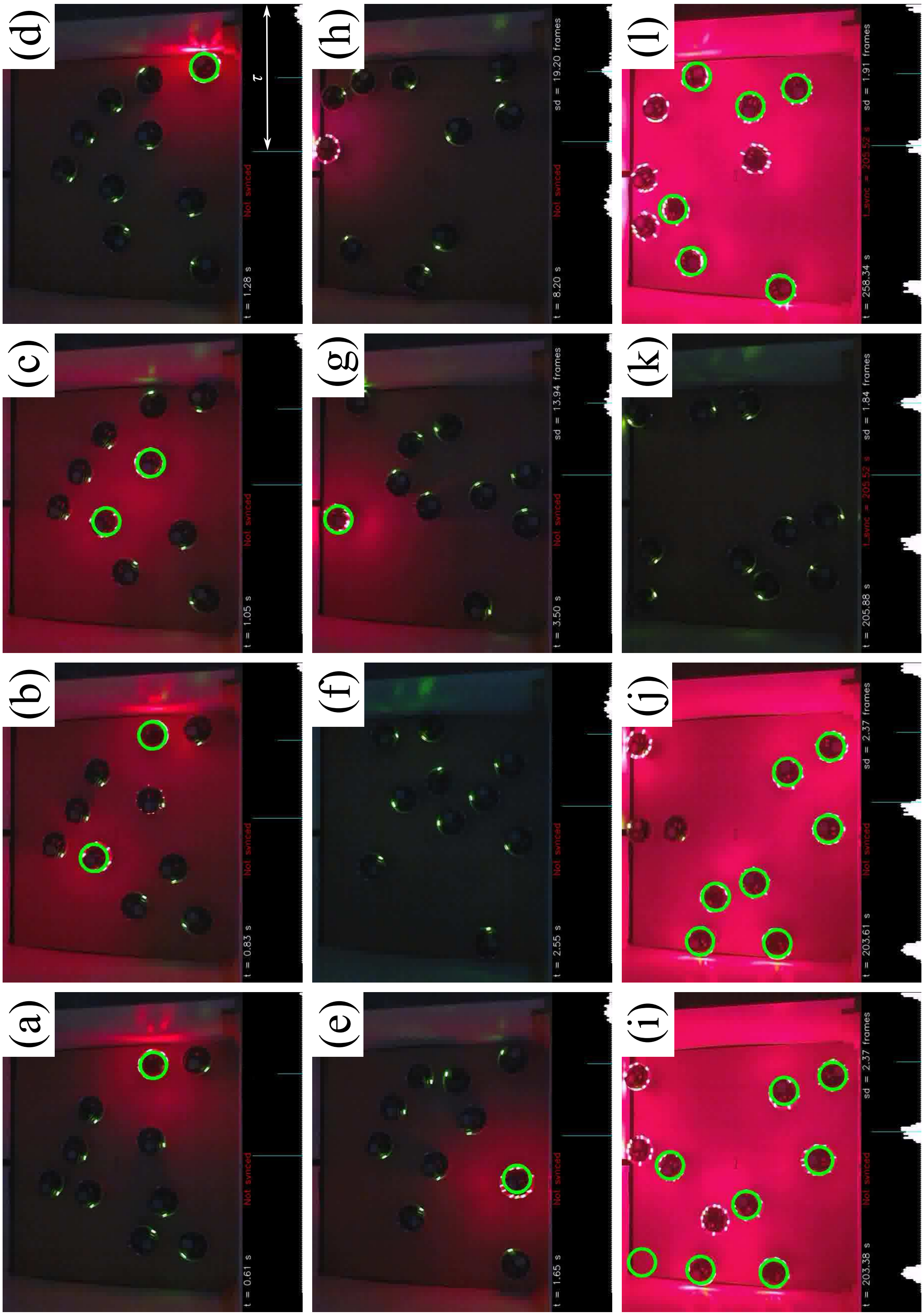


Figure 4.6: Time evolution of a robotic experiment trial with $V = 12.8$ cm/s. The histogram, counting the number of observed flashes (green circles) per frame evolves from right to left. Long blue lines mark an interval equal to the oscillator period, $\tau$. The standard deviation of the histogram during the $\tau$ interval is only calculated when the mode of the distribution passes through the midpoint of the interval (short blue line). The first row displays the initial moments of the trial. The standard deviation is first calculated in frame (g). The third row shows the system reaching and maintaining synchrony. A standard deviation smaller than 2 frames is first detected in frame (j), and it persists over time.

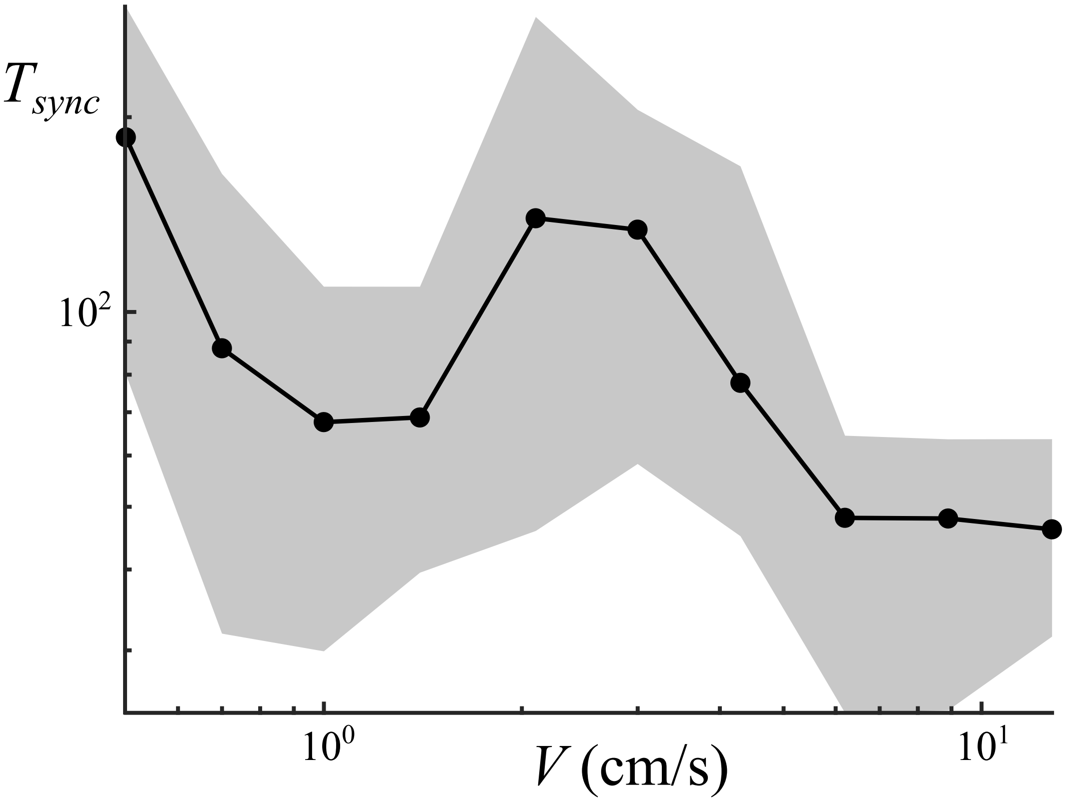


Figure 4.7: Time $T_{sync}$ (in number of cycles) needed to synchronize 10 physical e-pucks moving at speed $V$ in a $60 \times 60\,\mathrm{cm}^2$ arena. Black markers: average values over 5 trials. Grey contour: minimum and maximum values over the trials. The oscillator parameters are the same as for the dashed lines in Figs. 4.2 and 4.3: $\tau = 5\,\mathrm{s}$, $\epsilon = 0.1$, $\phi_{LED} = 0.075\,\tau$ and $\phi_{ref} = 0.15\,\tau$.

Observation of the trials yields visual confirmation of previous hypothesis regarding the underlying mechanisms governing each regime (Prignano et al., 2013; Perez-Diaz et al., 2015) (see trial videos in the online supplementary material (Perez-Diaz et al., 2016b)). In the slow regime an agent spends many cycles observing the same agent(s), or not seeing any other. This helps to form locally synchronized clusters. However, the whole system does not synchronize globally until a sufficient amount of neighborhood changes occur. In the fast regime, the opposite effect takes place. Each agent frequently changes the agents it sees. Therefore, global synchrony becomes much easier to attain. In the intermediate regime, the two mechanisms compete with each other. Local clusters are synchronized as in the slow regime, but they are constantly displaced by the frequent neighborhood changes.

## 4.4 Discussion

This chapter studied the presence of three synchronization regimes on a system of moving embodied pulse-coupled oscillators where the influence between agents is dictated by their cone of vision. The time $T_{sync}$ required to synchronize the system decreases as a function of agent speed in the slow and fast speed regimes. However for an intermediate range of speeds this dependence does not hold and a local maximum of $T_{sync}$ is observed. Building on previous work (Prignano et al., 2013; Perez-Diaz et al., 2015) we extended the understanding of the system by finding correlations between key features (namely the local minimum and maximum) and the involved parameters. In addition, the simulation results were experimentally validated in 50 trials with a group of 10 physical e-puck robots. Five trials were performed for 10 different robot speeds. The low number of trials is attributed to the time consuming set up process, which includes recharging the robot batteries for a few hours after every couple of trials. The lack of more data contributes to the large variance of the results.

The quantitative differences between the simulation and the physical system may be attributed to several factors. Firstly, real robots present differences from one another: the system clocks may be slightly different or experience jitter; the onboard cameras are not equally sensitive; and the proximity sensors vary greatly, both within the same robot and between robots, which may results in robots not avoiding obstacles efficiently and getting stuck for periods of time. Secondly, real robots present other imperfections. For instance, due to the low frame rate of the robot's camera, it can miss certain flashings when turning, or when other robots are too close or far away. Lastly, the simulation does not take into account reflections of the LED flashes on the arena walls, which can affect the effective field of view.

The biggest shortcoming of our experiments resides on the motion dynamics. As described in Sec. 4.1.3, robots move in a straight line except when avoiding other robots or reorienting at random when a wall is reached. The choice was made in order to build on the work of Prignano et al. (2013) and Perez-Diaz et al. (2015) (the previous chapter). One could argue that this is not a realistic motion for robots in a swarm, which may move independently or coordinately to perform some task. In lieu of our controller, different random walks could be explored. For instance, Dimidov et al. (2016) presented an in-depth study of random walks (from Brownian motion to Lévi walks) in the context of target searching in swarms of robots. It would be worth investigating the effect of the described motion models on firefly synchronization in the future.

Further future work could study whether the effect of speed on synchronization observed here is translatable to other related consensus problems. For instance, Trianni et al. (2016) studied the time required for a group of robots to converge to consensus in a similar setup to the one presented here. The authors studied a system of moving robots interacting with local neighbors within a certain range. As described in Sec. 4.1, a system of MPCOs influencing neighbors within a certain range yields a monotonically decreasing dependence of $T_{sync}$ with agent speed (Prignano et al., 2012), whereas a cone of vision leads to a nonmonotonic behavior. It would be worth investigating whether neighborhood type has similar effect on the convergence time in the described consensus problem.

# Chapter 5

# Robustness of Synchronization Regimes in Networks of Mobile Pulse-Coupled Oscillators[1]

Chapter 2 explained how agent synchronization and desynchronization could be utilized in decentralized multi-agent systems, such as robotic swarms and mobile sensor networks. It also described how pulse-coupled oscillators were garnering attention for these systems, due to their simplicity and intrinsically distributed nature. In these practical applications one is confronted with several challenges. Examples are: restricted communication range, latency, finite amplitude perturbations due to noise or jitter, etc. However, only the range interaction has been studied in these practical contexts, and the effect of perturbations has not been considered.

This chapter presents an in-depth analysis of the conditions and parameters that control the time required for a system of MPCOs to synchronize. We study the factors that affect the relationship between synchronization time and agent speed—from the monotonic dependence observed for range interactions (Prignano et al., 2012) to the nonmonotonic dependence found with nearest neighbor interactions (Prignano et al., 2013). Understanding the emergence of the aforementioned intermediate regime is of particular importance because it allows us to enable or impede

[1] This chapter has been adapted from Perez-Diaz et al. (unpublished).

synchronization depending on the application. In addition, we study the response of the system to perturbations such as those encountered in real systems. Our findings reveal the conditions for the control of clock or activity synchronization of agents with intermediate mobility.

We start by analyzing the effect of the oscillator's phase response curve on the nearest neighbor scenario and find that the inhibitory regime can be eliminated by adding an appropriate refractory period—an interval of time after emitting a pulse in which pulses from other agents are not received or taken into account. This is indeed a necessity in real-world applications. Contrary to idealized systems, propagation of a signal through physical media incurs latency. The subsequent processing of this signal also involves delays. Moreover, wireless antennas and other communication devices may not simultaneously transmit and receive signals (Wang et al., 2014). In a synchronized system, an oscillator could be pushed out of synchrony if it reacted to a slightly delayed signal from a neighbor. A refractory period can overcome problems that arise from these limitations.

Secondly, we analyze the effect of the neighborhood model, which determines the connectivity between agents. In practical applications a range connectivity may not always be achievable. Furthermore, an agent may be incapable of determining whether it is the closest to another agent, rendering the nearest neighbor interaction impractical. In this work we study three types of connectivity: cone of vision, cone of emission and k-nearest neighbors connectivities. As discussed in the preceding chapters, the cone of vision, in which an oscillator is only influenced by others in its cone of vision, is perhaps the more realistic connectivity in real-world scenarios. Its counterpart is the cone of emission, where an oscillator is only able to influence others in its cone of emission. In both cone of vision and emission the dimensions of such a cone (namely its angle and radius), influence the synchronization time. For narrow or short cones of vision, the aforementioned nonmonotonic behavior is found. However, as the size of the cone increases, the intermediate regime disappears, and synchronization time monotonically decreases with agent speed. Equivalently, for a system with K-nearest neighbor connectivity, we observe that as $K$ increases the intermediate regime becomes increasingly less inhibitory until the monotonic behavior is found.

Finally, the effect of perturbations on a synchronized system is investigated. We show that the propagation of this perturbation in the intermediate regime displays characteristics akin to stable chaos (Politi and Torcini, 2010). In addition, we study a system with small clock differences (i.e. oscillators with slightly different oscillation periods), in order to model jitter or manufacturing variations.

This chapter is organized as follows: In Sec. 5.1 we introduce the utilized models and metrics; Section 5.2 describes the effect of the response curve; Section 5.3 examines the effect of two neighborhood models on the synchronization curves; Section 5.4 investigates the effect of perturbations on the system; and Sec. 5.5 summarizes the chapter.

## 5.1 Methods

We consider a system of $N$ identical mobile agents moving in a two-dimensional (2D) bounded square environment of side length $L$. Unless otherwise stated, we set $N = 20$ and $L = 100$.

The adjacency matrix of the network, $A(t) = \{a_{ij}(t)\}$, defines the neighborhood of each agent based on geometrical criteria. Due to the mobility of the agents, the neighborhood evolves over time; $a_{ij}(t) = 1$ if unit $j$ is in the neighborhood of unit $i$ at time $t$, and $a_{ij}(t) = 0$ otherwise.

### 5.1.1 Oscillator Dynamics

Each unit, $i$, is a simple integrate-and-fire oscillator with free-running dynamics,

$$\frac{d\phi_i}{dt} = \frac{1}{\tau}, \tag{5.1}$$

where the phase $\phi_i \in [0, 1]$ grows linearly in time with period $\tau$ (set here to 1 without loss of generality), until the threshold, 1, is reached and a firing event occurs. Upon firing, the oscillator resets its phase to 0, and sends an instantaneous pulse to its neighbors. The phase of neighboring oscillators is shifted by a certain amount, which depends on the timing of the incident signal.

Equation (5.2) represents the dynamics of the system at the moment of a firing of oscillator $i$ at time $t$,

$$\phi_i\left(t^-\right) = 1 \Rightarrow \begin{cases} \phi_i\left(t^+\right) = 0 \\ \phi_j\left(t^+\right) = \phi_j\left(t^-\right) + a_{ij}(t)\Psi\left(\phi_j\left(t^-\right)\right), \end{cases} \tag{5.2}$$

where the map $\Psi : [0,1] \to \mathbb{R}$ is called a *phase response curve* (PRC). The factor $a_{ij}(t)$ guarantees that the phase shift is only applied to the current neighbors of the firing oscillator.

Several PRCs are considered: a *multiplicative* PRC, which produces an increase in phase proportional to the current phase (Prignano et al., 2012, 2013; Christensen et al., 2009); and two PRCs belonging to a class of response curves called *delay-advance* in the literature (Wang et al., 2012, 2014; Proskurnikov and Cao, 2015), which cause a decrease (inhibition) of the phase for $\phi \leq 0.5$ and an increase (excitation) for $\phi > 0.5$.

Equation (5.3) describes the multiplicative ($\Psi_{\text{mult}}$) PRC, whereas Eqs. (5.4) and (5.5) describe the delay-advance PRCs, that we will call *sawtooth* ($\Psi_{\text{saw}}$) and *sine* ($\Psi_{\text{sine}}$), respectively.

$$\Psi_{\text{mult}}(\phi) = \epsilon\phi \qquad \text{for}\, \phi \in (D,1] \tag{5.3}$$

$$\Psi_{\text{saw}}(\phi) = \begin{cases} -\kappa\phi, & \text{for}\, \phi \in (D,0.5] \\ \kappa(1-\phi), & \text{for}\, \phi \in (0.5,1] \end{cases} \tag{5.4}$$

$$\Psi_{\text{sine}}(\phi) = -\kappa\sin(2\pi\phi) \quad \text{for}\, \phi \in (D,1] \tag{5.5}$$

Constants $\epsilon \in \mathbb{R}_{\geq 0}$ and $\kappa \in [0,1]$ characterize the strength of the interaction. The interval $[0,D]$, with $0 \leq D < 0.5$, is the refractory period, during which no phase update is produced ($\Psi = 0$). As mentioned in the introduction, the refractory period is necessary in practical applications to counter physical delays as well as other technical constraints. In all cases, the resulting phase after the update, $\phi(t^+)$, is capped to 1, in order to prevent overshooting. In other words, for any PRC, $\Psi$, the effective phase response curve is $\Psi_{eff}(\phi) = \min(1-\phi,\, \Psi(\phi))$. Figure 5.1(a) shows examples of the three PRCs, where the aforesaid restriction has been taken into account.

Given that some units may fire upon receiving a phase update from a firing neighbor and, in turn, could elicit further firings, the phase shift is performed at frozen time until the phases of all oscillators have been updated. In addition, if more than one firing is simultaneously received by the same oscillator, a single phase update takes place.

### 5.1.2 Motion Dynamics

The units are initially placed at randomly chosen positions within the bounded environment. Their initial orientations are chosen at random from $[0, 2\pi)$. Each unit moves in a straight line at constant speed $V$ until it reaches the environment boundary. At that point, the unit reorients to a direction randomly chosen from $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ with respect to the boundary's normal and proceed with its straight line motion in the new direction. All random numbers are generated using uniform distributions.

The chosen motion of the agents is decoupled from the dynamics of the associated oscillators. This is because synchronization is not the main objective of a robotic swarm or a sensor network, but is instead a prerequisite for the agents to be able to coordinate their actions. Nevertheless, reorientation upon emitting or receiving a firing was also explored yielding qualitatively similar results (see Appendix D.1). In addition, whereas most previous work uses environments with cyclic boundary conditions (Fujiwara et al., 2011; Prignano et al., 2012, 2013), we opted for a bounded environment for practical realism. Cyclic boundary conditions yielded analogous results (see Appendix D.2).

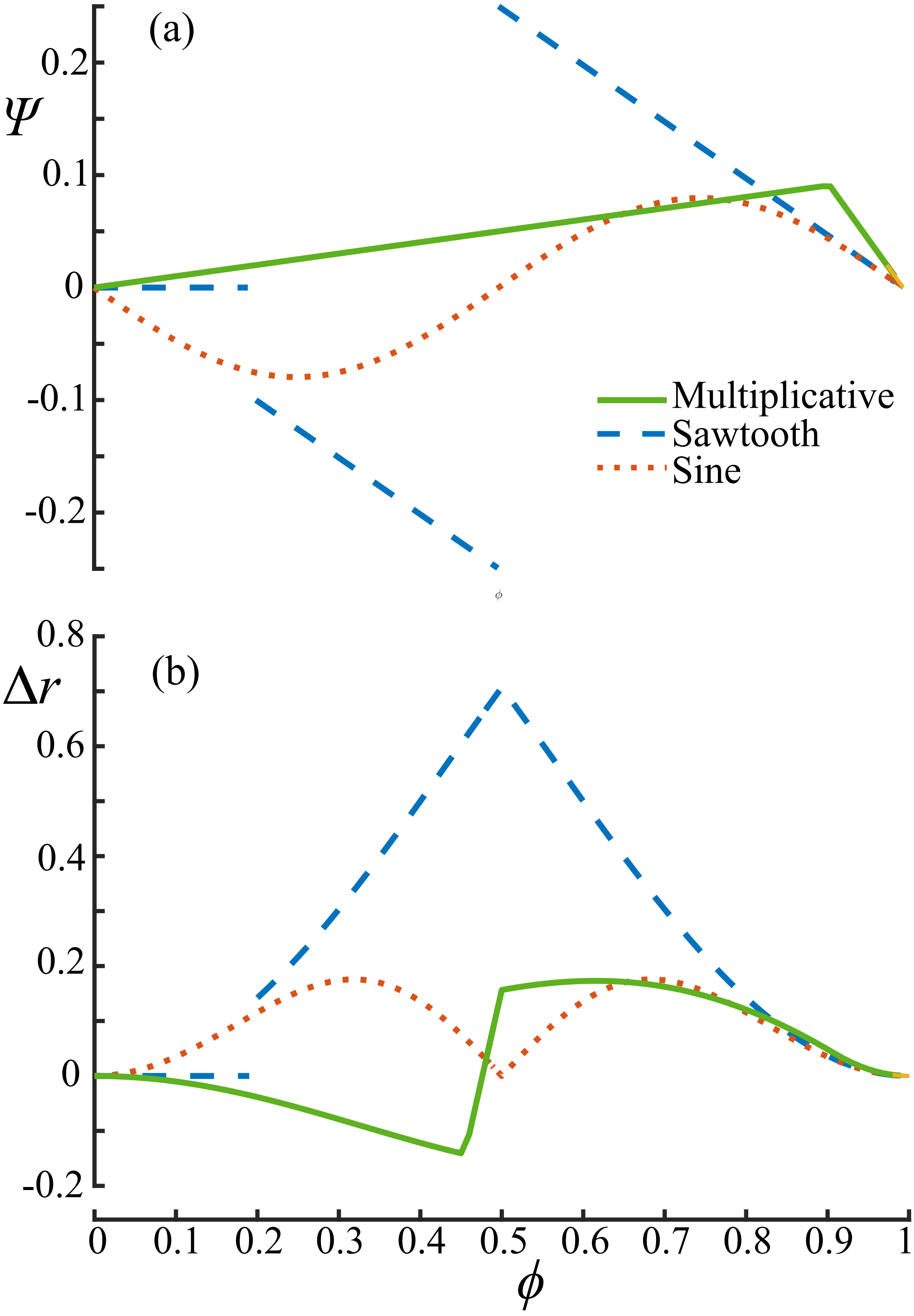


Figure 5.1: (a) Multiplicative PRC with $\epsilon = 0.1$ and $D = 0$, Sawtooth PRC with $\kappa = 0.5$ and $D = 0.2$, and Sine PRC with $\kappa = 0.08$ and $D = 0$. (b) Corresponding order parameter change, $\Delta r(\phi)$, after a firing of one oscillator in a two-unit fully-connected system.

### 5.1.3 Neighborhood Models

When an oscillator fires it induces a phase shift on its neighbors. We consider three neighborhood models based on geometrical criteria: *K-nearest neighbors* ($\mathrm{K_{nn}}$) connectivity, *cone of vision* connectivity, and *cone of emission* connectivity.

Given the positions of the oscillators at time $t$, $\{\mathbf{X_i}(t)\}$, the adjacency matrix of the network at that time, $A(t)$, is, in general, nonsymmetric (i.e. the graph is directed). The neighborhood of a given unit, $i$, at time $t$, is a set $\mathfrak{N}_i(t)$, which determines the non-zero elements of the adjacency matrix. Formally,

$$a_{ij}(t) = \begin{cases} 1 & \text{if } j \in \mathfrak{N}_i(t) \\ 0 & \text{otherwise} \end{cases}, \tag{5.6}$$

where $i, j \in \{1, 2, \dots, N\}$ and $i \neq j$. A unit cannot be its own neighbor, therefore the adjacency matrix has an all zeros diagonal, $A_{ii} = 0$.

In systems with K-nearest neighbors connectivity, oscillator $i$ influences the $K$ other units that are spatially closest to itself, $K \in \{1, 2, \dots, N-1\}$. We have that the cardinality of the neighborhood,$|\mathfrak{N}_i^{\mathrm{Knn}}(t)|$ is $K$ and that

$$j \in \mathfrak{N}_i^{\mathrm{Knn}}(t) \text{ if } \|\mathbf{X_j} - \mathbf{X_i}\| \;\leq\; \|\mathbf{X_l} - \mathbf{X_i}\| \;\forall l \notin \mathfrak{N}_i^{\mathrm{Knn}}(t),$$

where $\|\mathbf{X}\|$ represents the Euclidean norm. Note that the case where $K = N - 1$ is the *all-to-all* connected network. Figure 5.2(a) shows an example of $\mathrm{K_{nn}}$ connectivity for $K = 1$ in a system of three oscillators.

In systems with cone of vision connectivity, unit $j$ is considered a neighbor of another unit, $i$, if and only if $i$ lies inside the circular sector centered on $j$, with radius $R$, angle $\theta$, and oriented in the direction of motion of $j$. That is,

$$j \in \mathfrak{N}_i^{\overset{\text{cone}}{\text{vision}}}(t) \text{ if } \begin{cases} \|\mathbf{X_i} - \mathbf{X_j}\| \leq R \\ \frac{\mathbf{X_i} - \mathbf{X_j}}{\|\mathbf{X_i} - \mathbf{X_j}\|} \; \frac{\mathbf{V_j}}{\|\mathbf{V_j}\|} \;\leq \cos(\theta/2) \end{cases},$$

where $\mathbf{V_j}$ is the velocity vector of unit $j$, with $\|\mathbf{V_j}\| = V$. Figure 5.2(c) shows an

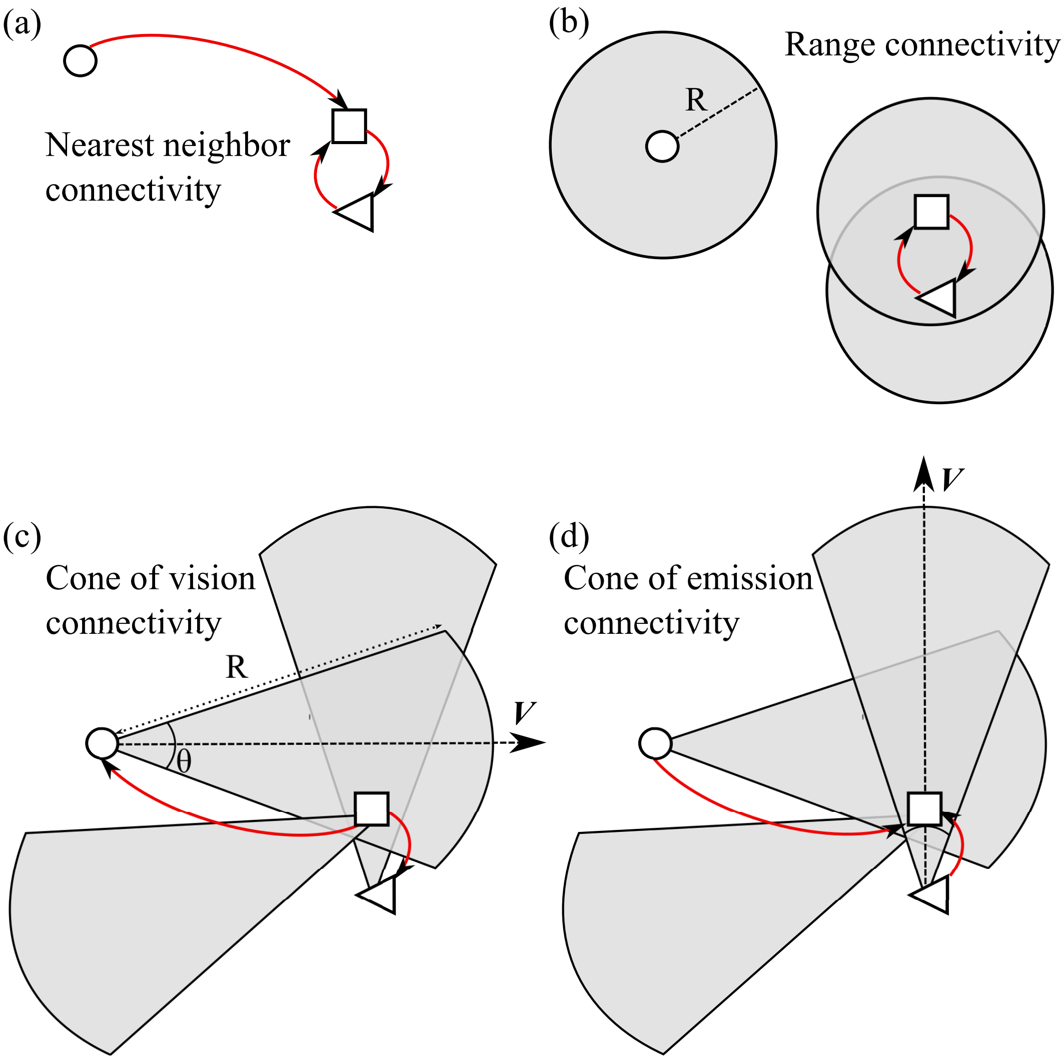

(a)
Nearest neighbor connectivity
(b)
Range connectivity
R
(c)
Cone of vision connectivity
R
θ
V
(d)
Cone of emission connectivity
V

Figure 5.2: Different neighborhood models and the corresponding interactions (solid arrows) for a system of three units (◯, □, △). Note that the unit at the end of an arrow denotes the neighbor of the unit at its origin. (a) *Nearest neighbor connectivity*—each unit influences the spatially nearest unit. □ and △ influence each other, ◯ influences □. (b) *Range connectivity*—two nodes influence each other if they lie within range $R$. □ and △ influence each other, ◯ does not influence other units as they are out of range. (c) *Cone of vision connectivity*—if a unit can *see* another unit (see text for details), the latter will influence the former. □ lies in the cone of vision of the other two units, thus it influences both. ◯ and △ do not influence other units for they do not lie in any other unit's cone of vision. (d) *Cone of emission connectivity*—if a unit can *reach* another unit (see text for details), the former will influence the latter. □ lies in the cone of emission of the other two units, thus is influenced by both △ and ◯. These two units, in turn, are not influenced by other units as they do not lie in any other unit's cone of emission.

example of cone of vision connectivity for a system of three oscillators.

Conversely, in systems with cone of emission connectivity the direction of the interaction is the opposite of the preceding case. Unit $j$ is considered a neighbor of another unit, $i$, if and only if $j$ lies inside the circular sector centered on $i$, with radius $R$, angle $\theta$, and oriented in the direction of motion of $i$. That is,

$$j \in \mathfrak{N}_i^{\overset{\text{cone}}{\text{emission}}}(t) \text{ if } \begin{cases} \|\mathbf{X_i} - \mathbf{X_j}\| \leq R \\ \frac{\mathbf{X_j}-\mathbf{X_i}}{\|\mathbf{X_j}-\mathbf{X_i}\|} \frac{\mathbf{V_i}}{\|\mathbf{V_i}\|} \leq \cos(\theta/2) \end{cases},$$

where $\mathbf{V_i}$ is the velocity vector of unit $i$, with $\|\mathbf{V_i}\| = V$. Figure 5.2(d) shows an example of cone of vision connectivity for a system of three oscillators. Compared with Fig. 5.2(c) the direction of the influence is reversed.

Note that, for both cone of vision and cone of emission, in the limit where $\theta \to 2\pi$ the network becomes a random geometric graph, and the limit where $R \to \sqrt{2}L$ and $\theta \to 2\pi$ corresponds to an all-to-all connected network.

It is important to remark on the *directionality* of the cone of vision connectivity. Contrary to K-nearest neighbors, cone of emission and the much used range of interaction, with a cone of vision a firing oscillator, $i$, does not directly elicit a phase update on other oscillators. Instead, the recipient unit must be able to physically see $i$ to react to that firing (see Fig. 5.2(c)). This choice was made to emulate a practical scenario. For instance, a robot may only have a directional camera, antenna or sensor to detect the pulse. Note that the 360° cone of emission is the previously studied range of interaction and it is equivalent to the 360° cone of vision (cf. Fig. 5.2(b) and (c)). That is, if $i$ is in the 360° cone of vision of $j$, $j$ will also be in the 360° cone of emission of $i$.

### 5.1.4 Synchronization Metric

The collective dynamics of the system are measured by the complex order parameter (Arenas et al., 2008),

$$r(t)e^{\mathrm{i}2\pi\phi(t)} = \frac{1}{N}\sum_{j=1}^{N} e^{\mathrm{i}2\pi\phi_j(t)}, \tag{5.7}$$

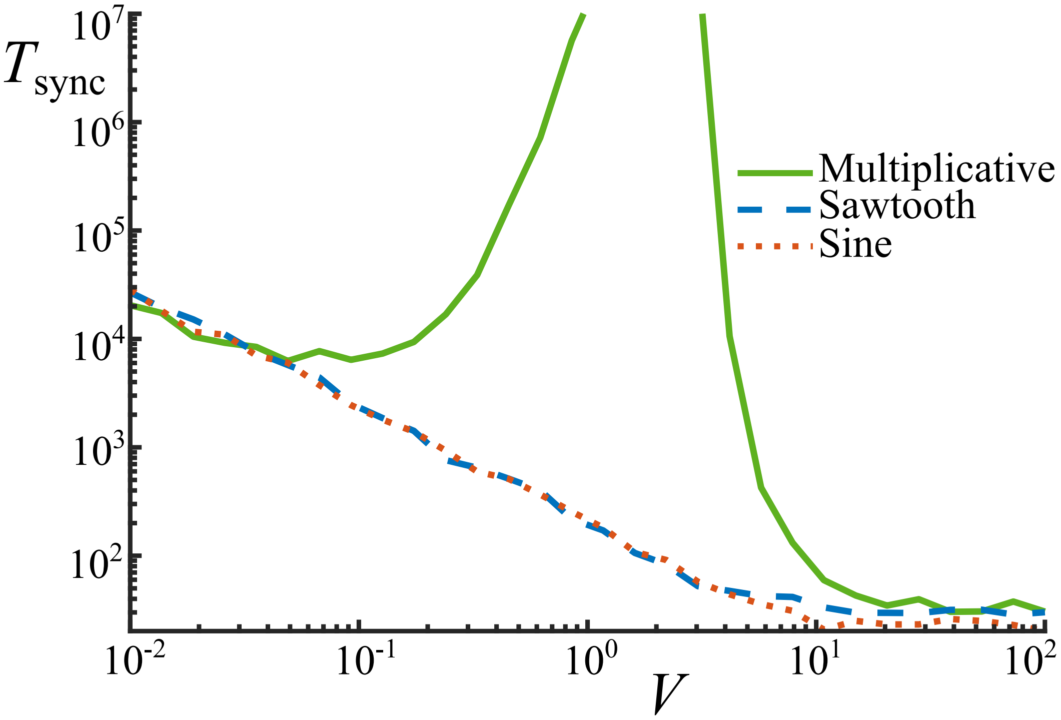


Figure 5.3: Time $T_{sync}$ (in number of cycles) required to synchronize 20 units with nearest neighbor connectivity and moving with speed $V$ in a 2D square environment of side 100 for the multiplicative ($\epsilon = 0.1$), sawtooth ($\kappa = 0.1$), and sine ($\kappa = 0.1$) PRCs with no refractory period ($D = 0$).

where $\phi(t)$ is the mean phase and modulus $r(t)$ measures the level of coherence, from $r(t) = 0$ for a totally incoherent system, to $r(t) = 1$ for complete synchronization. Henceforth we will refer to the real part $r(t)$ as the order parameter without ambiguity.

A certain oscillator, $\phi_1$, is selected as reference, and the order parameter is calculated upon its $k$th firing at time $T_k$, $r(T_k)$. The simulation is stopped once the system is synchronized (i.e. $r(T_k) = 1$) and the number of firings, $k$, emitted by the reference oscillator until synchronization is counted. This value, that we will call *synchronization time* $T_{sync}$, is a good measure of the time needed to achieve coherence independently of the oscillation period.

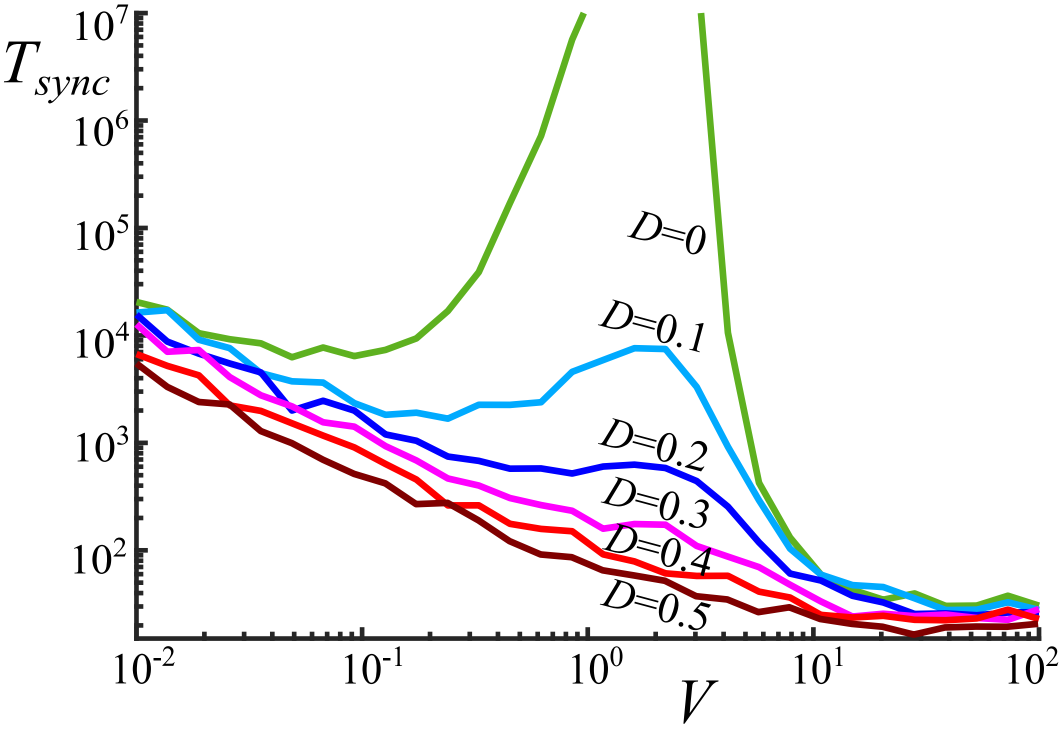


Figure 5.4: Time $T_{sync}$ (in number of cycles) required to synchronize 20 units with nearest neighbor connectivity and moving with speed $V$ in a 2D square environment of side 100 using the multiplicative PRC for different refractory periods.

## 5.2 Effect of the Response Curve

We start by extending the work by Prignano et al. (2013) on oscillators interacting with their nearest neighbor to different PRCs. In their work, they observe a non-monotonic dependence of the synchronization time, $T_{sync}$, on the speed of agents, $V$, for the multiplicative PRC, with no refractory period ($D = 0$). We observe that both delay-advance response curves, sawtooth and sine, display a monotonically decreasing dependence (see Fig. 5.3).

We attribute the observed difference of behavior to the sign of $\Psi(\phi)$. As can be seen in Fig. 5.1(a) both delay-advance response curves have a negative sign for $\phi < 0.5$ and positive otherwise. This means that the phase of an oscillator receiving a pulse will always get closer to the emitting pulse (note that the two phase limits $\phi = 0$ and $\phi = 1$ are equivalent). However, for the multiplicative PRC, the receiving oscillator will distance its phase from the pulse emitter for any $0 < \phi < \frac{1}{2+\epsilon}$.

We can measure the effect on synchronization of a PRC by considering a system of only two units, one emitting a pulse (i.e. phase equal to one) and the other, with phase $\phi$, receiving it, and calculating the difference of the system's order parameter before and after the firing,

$$\Delta r(\phi) = r(\phi(t^{+})) - r(\phi(t^{-}))\,. \tag{5.8}$$

Figure 5.1(b) shows that $\Delta r(\phi)$ is always positive for the delay-advance PRCs but it is negative when $0 < \phi < \frac{1}{2+\epsilon}$ for the multiplicative PRC. Adding a refractory period as in Eq. 5.3 confirms that this negative interval hinders synchronization (see Fig. 5.4). When suppressing the effects of this interval, the nonmonotonicity disappears as $D$ increases. Moreover, sufficiently large refractory periods ($D \geq 0.4$) display a faster synchronization for slow speeds than any of the delay-advance PRCs in Fig. 5.3.

## 5.3 Effect of the Neighborhood Model

In the previous section we concluded that the multiplicative PRC with no refractory period displays the strongest nonmonotonic behavior in a system with nearest neighbor connectivity. In this section we study the effect of three other neighborhood models for the same phase response curve. Henceforth, we set the multiplicative factor in Eq. 5.3 to $\epsilon = 0.1$.

### 5.3.1 K-Nearest Neighbors

Firstly, we extend the nearest neighbor interaction to K-nearest neighbors. We observe that the intermediate, synchronization-inhibiting, regime becomes weaker as $K$ increases, until it finally disappears (see Fig. 5.5). Large enough $K$ ($K > 5$) yield a monotonically decreasing dependence of $T_{sync}$ on $V$, similar to the one observed in previous work for a 360° range interaction (Prignano et al., 2012).

It is worth noting that, in general, mobility is necessary for the system to synchronize. For a static population of oscillators, it is necessary (although typically

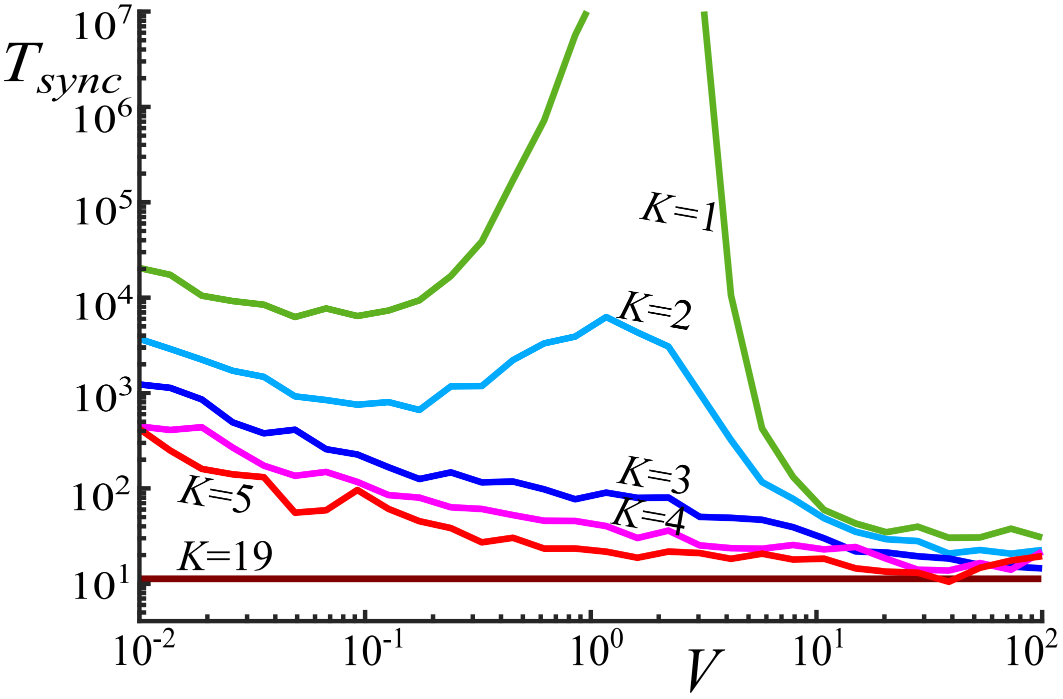


Figure 5.5: Time $T_{sync}$ (in number of cycles) required to synchronize 20 units with K-nearest neighbors connectivity and moving with speed $V$ in a 2D square environment of side 100.

not sufficient) that the undirected graph forms a single connected component for global synchronization to be achieved. Otherwise, two or more clusters could remain isolated from each other, reaching only local synchrony. The smallest cluster size with $\mathrm{K_{nn}}$ connectivity is the complete (fully connected) $K+1$ directed graph. We have that for $K+1 \leq N/2$ there could exist two or more non-interacting clusters and, therefore, synchronization is not guaranteed if the units are static. The case where $K = 19$ in Fig. 5.5 corresponds to the trivial all-to-all connected network.

### 5.3.2 Cone of Vision and Cone of Emission

In the preceding subsection we observed how the synchronization time of the system can be tuned, not only by changing the speed of agents, but also by changing the number of neighbors with which each unit interacts. This tuning is also applicable to cone of vision and cone of emission neighborhoods. A cone (of vision or emission), being a circular sector, can be considered an extension of the widely studied range of interaction. However, in contrast to the latter, where $T_{sync}$ always exhibits a monotonic dependence with $V$, the dimensions of the cone of vision can alter the synchronization behavior.

Figures 5.6 and 5.7 show the resulting synchronization behavior for various cone angles and radii using a cone of vision and a cone of emission, respectively. Large ($R \to \sqrt{2}L$) and wide ($\theta \to 2\pi$) cones lead to a decreasing dependence of $T_{sync}$ on $V$. Nevertheless, as the size of the cone decreases (either $\theta$ or $R$ decrease), the monotonicity is broken, and synchronization is slowed for an intermediate range of speeds. In fact, in the top-left panel in Figs. 5.6 and 5.7 three clearly distinct sectors, corresponding to three dynamical regimes, can be observed. Comparing Figures 5.6 and 5.7 it can be seen that the effect of the cone of vision and cone of emission neighborhoods is qualitatively similar. However, the nonmonotonicity is more prominent with a cone of vision.

Note that the interplay between the radius and the angle of the cone is not trivial. For a given speed, a wider angle and larger radius produce, in general, faster synchronization. However, a greater radius can slow synchronization for particular speeds and angles (see Fig. 5.6 for $\theta = 10°$, $45°$, $90°$, and $120°$), or a wider angle can slow synchronization for particular speeds and radii (cf. Fig. 5.6 for $\theta = 10°$ and $120°$ at $V = 10^0$ with $R < 20$).

As in the previous section, in the general case mobility is necessary for the system to synchronize. For a cone of vision only $R \geq \sqrt{2}L$ and $\theta = 360°$ (i.e. the all-to-all connected network) guarantee that the units will always form a single connected cluster.

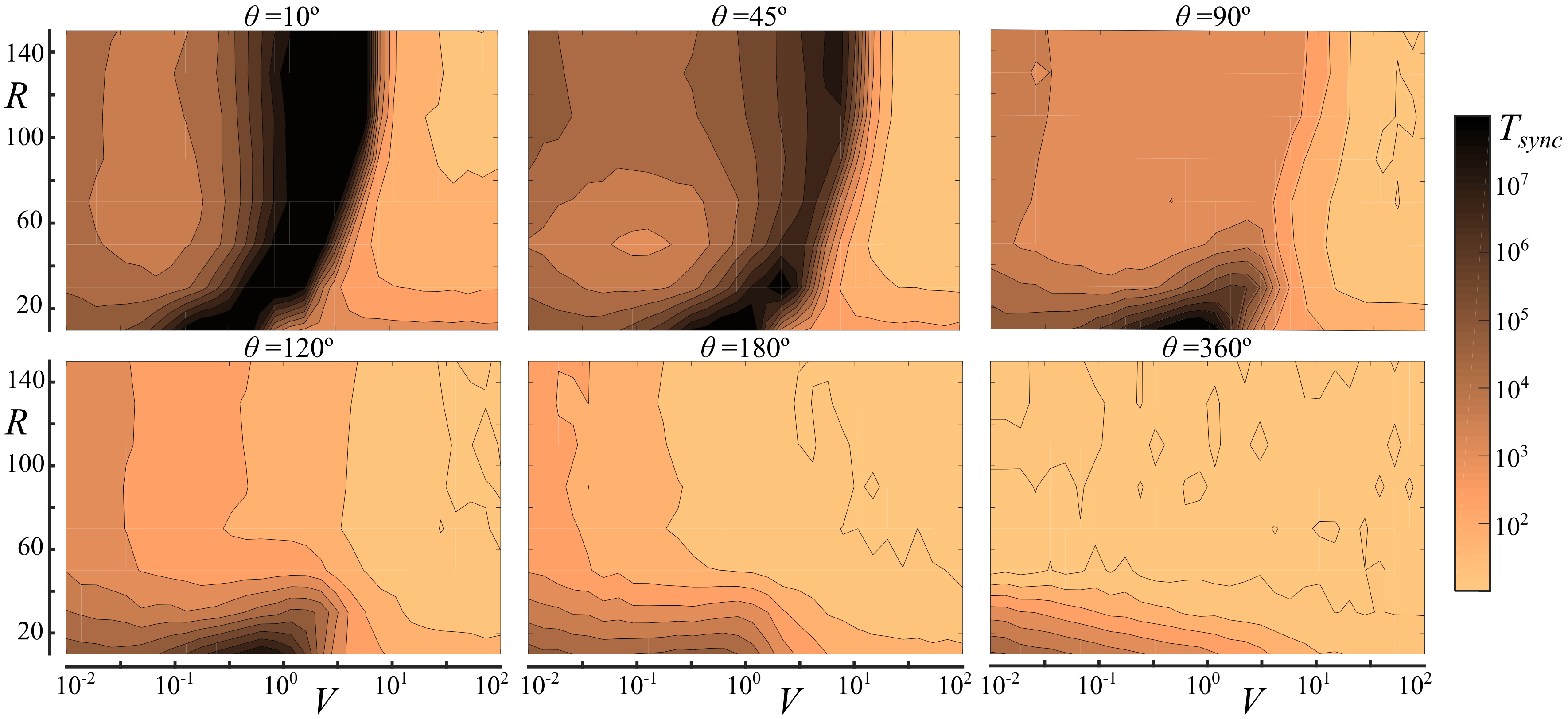


Figure 5.6: Time $T_{sync}$ (in number of cycles) required to synchronize 20 units with cone of vision connectivity (with radius $R$ and angle $\theta$) and moving with speed $V$ in a 2D square environment of side 100 using the multiplicative PRC.

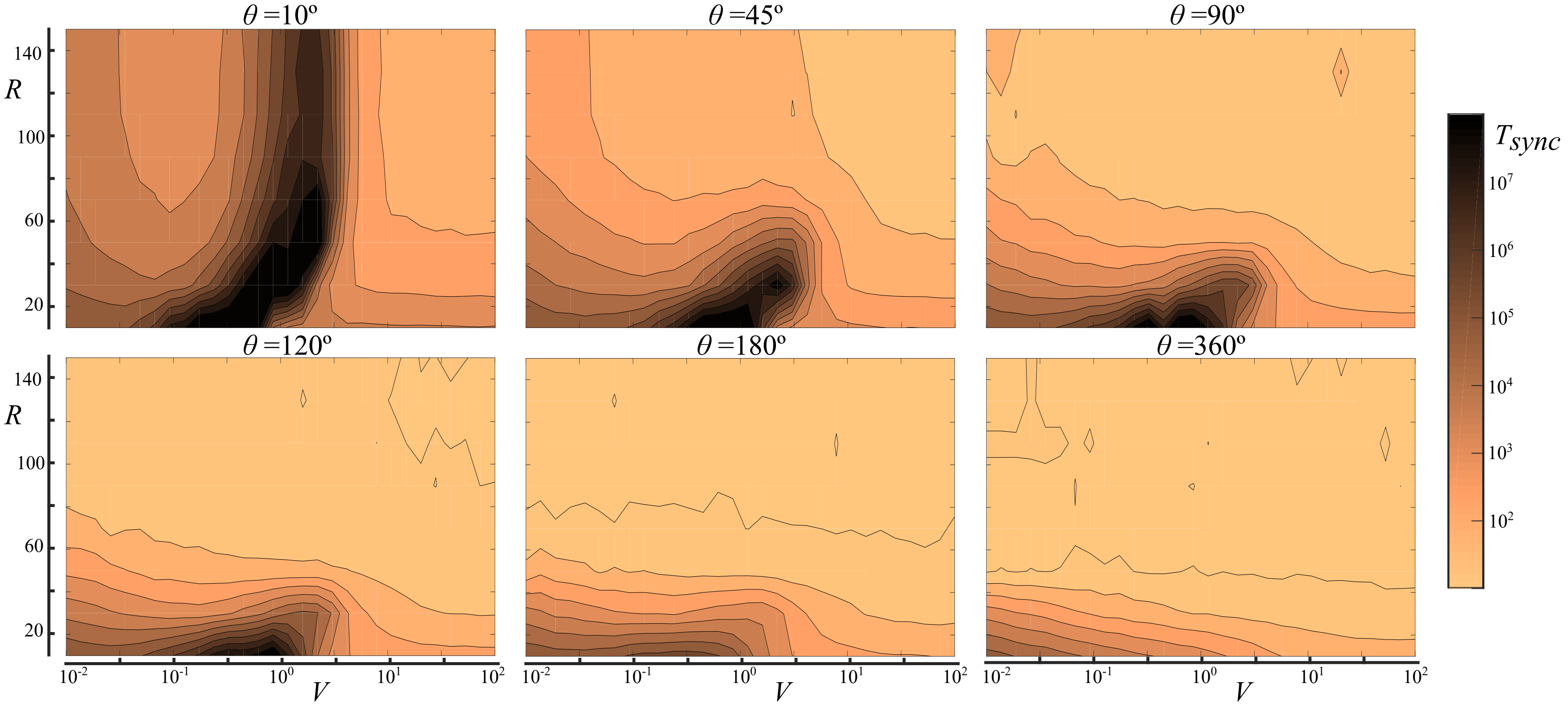


Figure 5.7: Time $T_{sync}$ (in number of cycles) required to synchronize 20 units with cone of emission connectivity (with radius $R$ and angle $\theta$) and moving with speed $V$ in a 2D square environment of side 100 using the multiplicative PRC.

## 5.4 Effect of Perturbations

Real-world systems are subject to a number of imperfections, such as noise or variations within manufacturing tolerances. In this section we study the effect of perturbations in two ways: firstly, we measure the propagation of finite-amplitude perturbations through the system; secondly, we study a system with slightly misaligned clocks to model possible physical faults or jitter.

### 5.4.1 Finite-Amplitude Perturbations

Departing from a fully synchronized state, one of the oscillators is initially displaced by a small but finite amount [$\phi_1(0) = 0.001$, $\phi_{j\neq 1}(0) = 0$]. Figure 5.8 provides a typical example of the effect of this perturbation over the system for a relatively small cone of vision ($R = 40$ and $\theta = 20°$). The figure shows the phase difference $|\Delta\phi|$, between the perturbed and unperturbed [$\phi_1(0) = 0$] systems over time.

In the slow regime (top panel), the perturbation spreads among the connected units, but the system settles before the topology changes. In the fast regime (bottom panel), the perturbation is quickly transferred to all other units. However its magnitude remains small until it completely vanishes due to the rapid changes in connectivity. The connection between two given oscillators is too short-lived for long-lasting changes to be produced. In addition, due to the high speeds, the oscillators effectively experience a mean field of the interactions which causes the displaced system to rapidly return to coherence. Naturally, the resulting synchronous state for both slow and fast regimes could be displaced by a constant amount with respect to the unperturbed scenario (see Fig. 5.8 bottom panel).

In the intermediate regime (central panel) characteristics from both the slow and fast regimes are present. On the one hand, the movement is sufficiently fast for the perturbation to spread over all units. On the other hand, the changes of connectivity occur at a rate that still permits the magnitude of this perturbation to be amplified.

The obtained results are equivalent to those found with oscillators interacting with their nearest neighbor as can be seen from Figure 5.9. The plot shows that $|\Delta\phi|$ (order parameter, $r$) increases (decreases) with time for intermediate speeds,

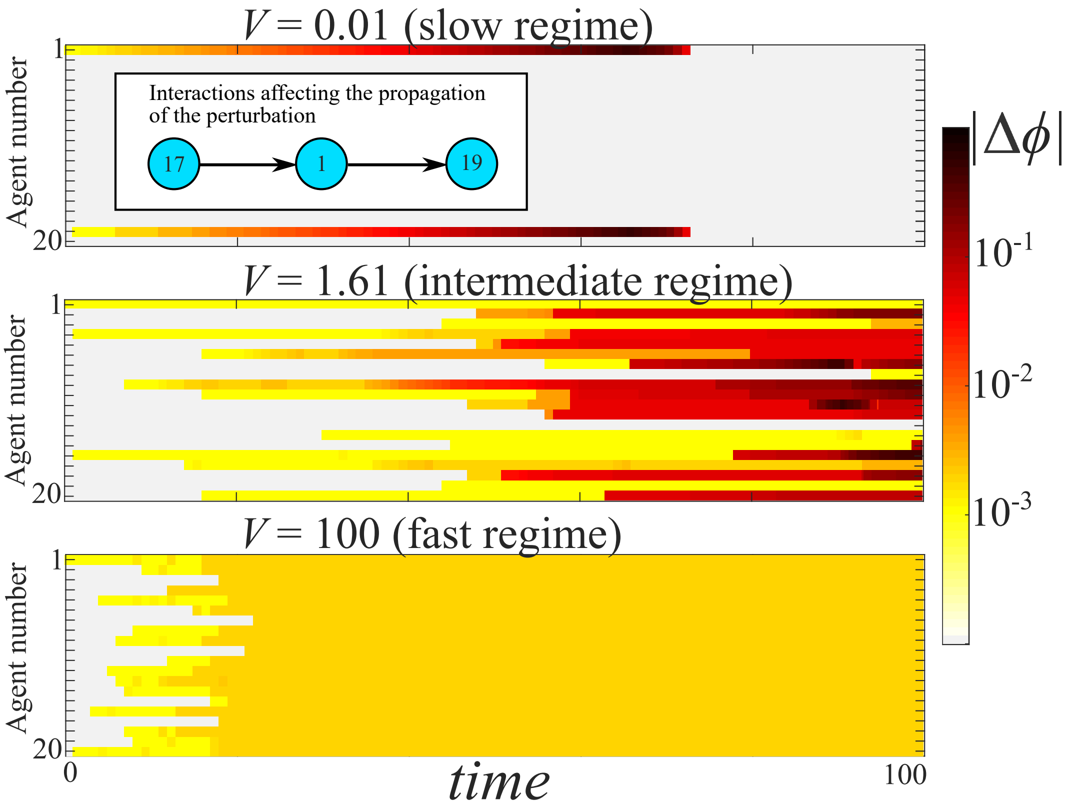


Figure 5.8: Propagation of small perturbation over time in a system of 20 units with cone of vision connectivity ($R = 40$ and $\theta = 20°$) using the multiplicative PRC. The color represents the magnitude, in logarithmic scale, of the phase difference, $|\Delta\phi|$, with respect to the unperturbed system. Top panel: slow regime. Central panel: intermediate regime. Bottom panel: fast regime. Inset: oscillators that play a role in the propagation of the perturbation in the slow regime. The initially perturbed unit 1 displaces unit 19. In parallel, undisturbed unit 17 brings 1 (and therefore 19) back to synchrony with the rest of the system.

whereas the perturbation vanishes after some time in the slow regime, and the system is barely affected in the fast regime.

In the intermediate regime all memory of the initially synchronized state is lost in the span of a few firing events, which is typical of chaotic and stochastic regimes. Synchrony will eventually be reached, however, after a long number of cycles as if it were a randomly initialized system.

The sensitivity to small but finite-amplitude perturbations in the intermediate regime resembles the behavior of stable chaos, which has been observed in networks of integrate-and-fire neurons (Politi and Torcini, 2010; Zillmer et al., 2009). In

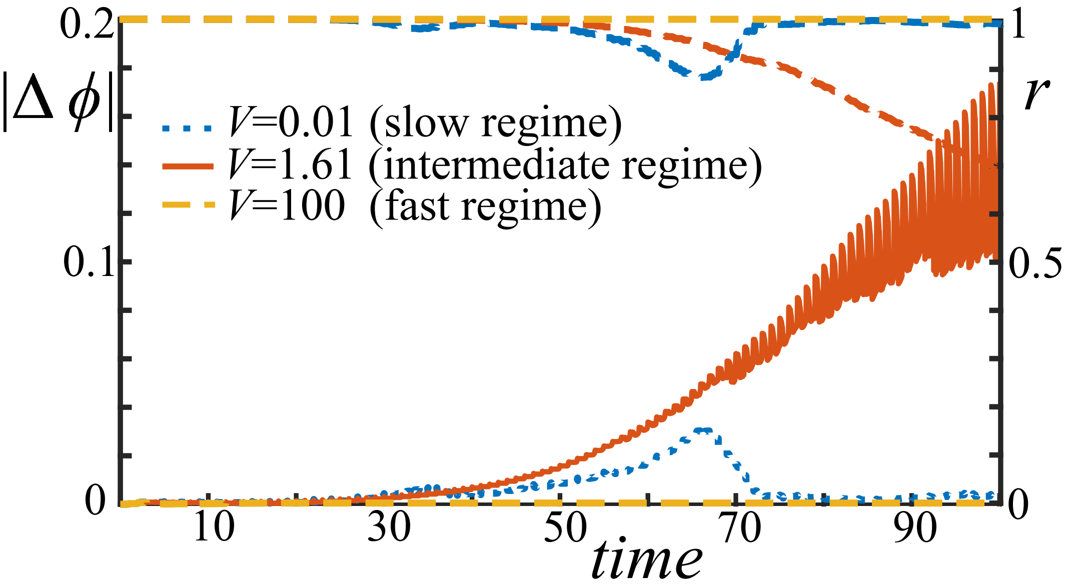


Figure 5.9: Average evolution of the order parameter, $r$, (top curves) and $|\Delta\phi|$ (bottom curves) after applying a perturbation once to a single oscillator in an initially synchronized system of 20 units with nearest neighbor connectivity.

contrast to *standard* chaos, stable chaos is a transient phenomenon restricted to finite time-scales. However, the transient duration diverges exponentially with system size. Indeed, as shown in Fig. 5.10 for a system with nearest neighbor interaction, the time to achieve synchronization after applying a perturbation grows exponentially with population size for intermediate speeds. In contrast, $T_{sync}$ barely depends on $N$ in the slow and fast regimes. Prignano et al. (2013) found that the speed at which the system enters the fast regime for nearest neighbor interaction is $V_f \propto L/\sqrt{N}$. Therefore in Fig. 5.10 the density of the system, $N/L^2$, is kept constant while increasing the number of units in order to ensure that the fast regime limit remains fixed.

### 5.4.2 Clock Jitter

In order to model jitter, we consider a system where the oscillation period is different for each oscillator. Therefore, Eq. 5.1 transforms into

$$\frac{d\phi_i}{dt} = \frac{1}{\tau_i}, \tag{5.9}$$

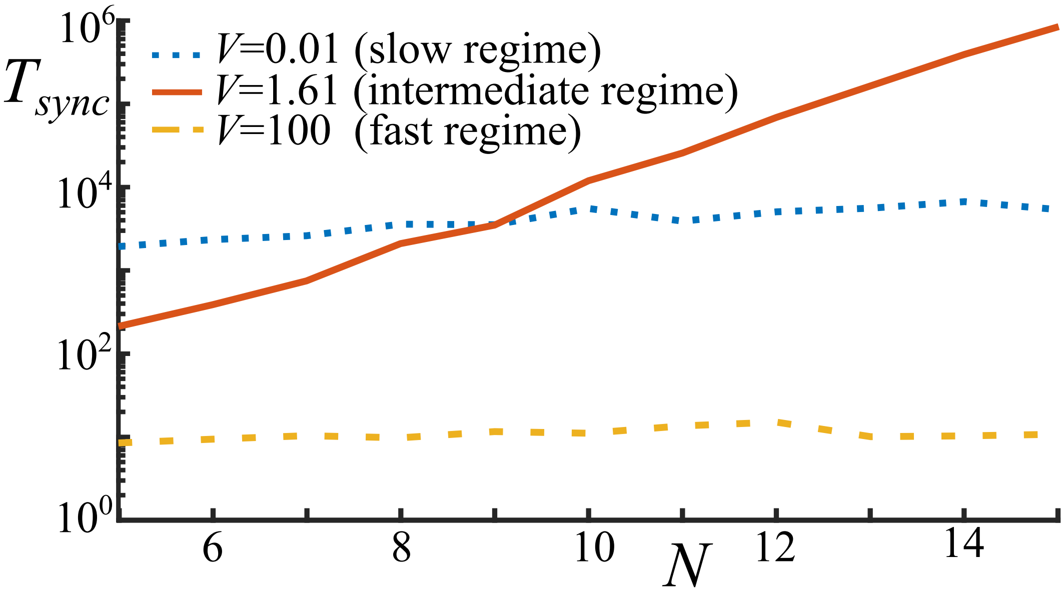


Figure 5.10: Average transient time until synchronization, $T_{sync}$, as a function of system size after a small perturbation is applied once to a single oscillator in an initially synchronized system with nearest neighbor connectivity. Average performed over 20 random initializations (positions and orientations) with fixed initial perturbation [$\phi_1(0) = 0.001$, $\phi_{j\neq1}(0) = 0$].

where $\tau_i$ is the oscillator period of unit $i$. In the following $\tau_i$ is randomly selected from the Gaussian distribution centered in $\tau = 1$ and with standard deviation $\sigma = 5 \times 10^{-5}$. The choice of $\sigma$ was made to better illustrate the effect of clock misalignment on synchronization.

Departing from a fully synchronized state, the evolution of the system is recorded. Figure 5.11 shows the order parameter over time for speeds in the three dynamical regimes in a system with nearest neighbor connectivity. Similarly to the results reported in Section V.A, the fast regime is negligibly affected by the perturbations. However, for both low and intermediate speeds the oscillators are permanently displaced from synchrony. In the intermediate regime, the system gets totally desynchronized, almost approaching a random system ($r = 0$). In contrast, for slow speeds, connected oscillators can achieve local synchronization. Therefore, the effect on global synchronization is not as severe as in the intermediate regime.

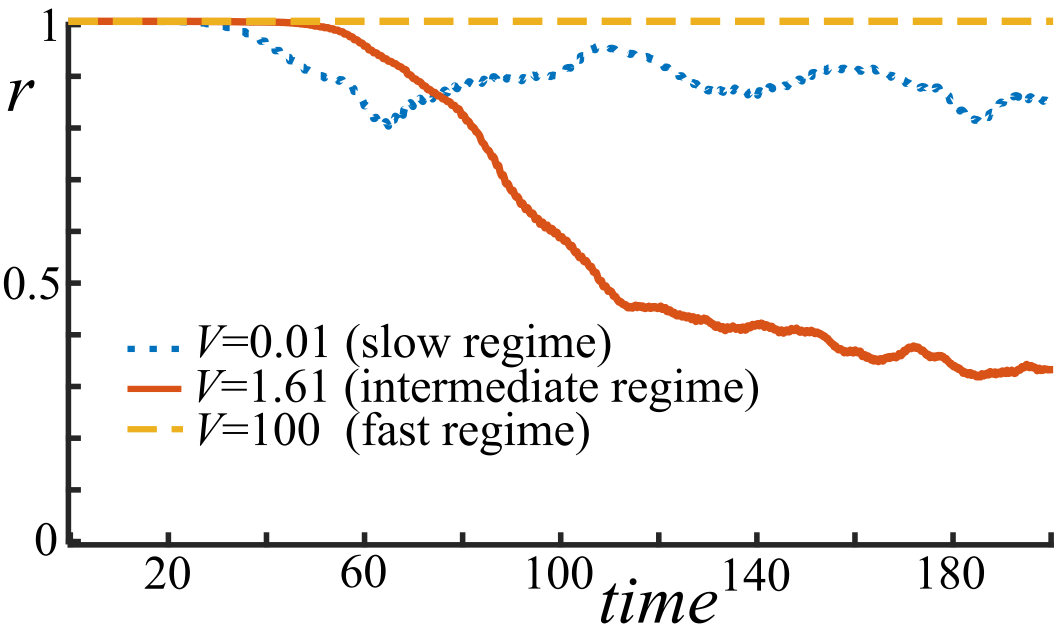


Figure 5.11: Average evolution of the order parameter $r$, in a system with clock jitter, in which the oscillators are initially synchronized. Average over 100 populations of 20 units, with Gaussian distributed oscillator periods, $\tau_i$ (see text for details).

## 5.5 Discussion

In this chapter we analyzed the dependence of synchronization time on the speed of agents in a population of mobile pulse-coupled oscillators. Previous research showed that synchronization is facilitated by oscillator mobility when the agents interact with others within a certain range. In contrast, if the agents interact with their nearest neighbors, a regime of intermediate speeds is observed, where synchronization is hindered. Taking this as a point of departure, we studied various factors that lead to this pernicious effect with a view towards controlling synchronization in real-world applications.

Firstly, the conditions for the previously reported nonmonotonic behavior with nearest neighbor interaction were studied. We found that the synchronization-hindering regime only emerges with a multiplicative phase response curve whereas it is not present in the other studied phase response curves. We devised a metric for the phase response curve, $\Delta r(\phi)$, that correlates with the emergence of the pernicious interval. Indeed, a partial or total blocking of this interval, by means of a refractory period, recovers the monotonic dependence.

We extended the previously studied work, by studying K-nearest neighbors, cone of vision and cone of emission connectivities with the multiplicative phase response curve. We linked the previously found monotonic and nonmonotonic synchronization curves by tuning the average size of the agents' neighborhood. For K-nearest neighbor connectivity, as $K$ increases the effect of the intermediate regime is decreased until a monotonically decreasing dependence is observed. Similarly, for both the cone of vision and the cone of emission connectivities, a synchronization impeding regime appears for small or narrow cones. This nonmonotonic dependence of the synchronization time with agents' speed can be gradually transformed into a monotonic one by widening or extending the range of the cone of vision. The interplay between the cone parameters, $\theta$ and $R$, is not trivial; increasing the angle or range of the cone can result in an increased synchronization time. This aspect deserves further analysis in future work.

Finally, we analyzed the effect and propagation of small perturbations on initially synchronized systems. It was found that the perturbations were contained within the locally connected cluster and did not spread over the system for the slow regime. In the fast regime, the perturbation propagates to all agents but its magnitude remains small due to the brevity of each interaction. For both slow and fast speeds a coherent state is achieved in relatively short time. Interestingly, in the intermediate regime, perturbations get amplified as they propagate through the system. Nevertheless, the system will eventually return to synchrony after a certain transient period. We found that the length of this transient increases exponentially with system size. Given the characteristics of this irregular behavior, we draw an analogy with stable chaos. The latter was found in models characterized by discontinuities in the evolution rule (Politi and Torcini, 2010). In our case, the firing events and the elicited phase responses constitute such discontinuities.

Similarly to the effect of small perturbations, a system with clock jitter exhibited a strong departure from synchrony in the intermediate regime. Jitter is, to a lesser degree, also deleterious for synchronization if the movement is slow. This implies that, in real applications using the multiplicative phase response curve, synchronization is stable only for sufficiently high agent speeds.

An important difference between our model and the previous approach on nearest neighbor interaction (Prignano et al., 2013) is that, in our case, the motion of the units is totally decoupled from the oscillator dynamics. Other motion dynamics were explored with similar results (see Appendix D). Namely, randomly reorienting units upon firing or upon being influenced by another unit (as in Prignano et al., 2013). Therefore, we observe that the emergence of synchronization regimes depends on the relationship between the temporal dynamics of the oscillators, the spatio-temporal dynamics of the units' motion, and the spatial influence of the neighborhood models.

In practical applications in multi-agent systems, such as sensor networks or robotic swarms, both synchronization and desynchronization can serve their purpose to coordinate the agents' activities. The former allows the agents to perform their actions at the same time whereas the latter allows the distribution of individual activities over time. We have observed that the intermediate regime not only slows synchronization down but also promotes desynchronization after small perturbations are applied to the system. Depending on what is most desirable at the time, the agents could easily switch between a synchronized state or a desynchronized one conveniently changing their speed, modifying their neighborhood scope, increasing or decreasing the refractory period, and/or perturbing the phase of just a single oscillator.

# Chapter 6

# Conclusions

This thesis presented an in-depth investigation of firefly synchronization (or pulse-coupled oscillator synchronization) in groups of mobile agents (that is, mobile pulse-coupled oscillators) with a view towards implementations in technological applications, mainly swarm robotics and sensor networks.

We built on the findings of Prignano et al. (2012, 2013), which show how mobile oscillators describe a monotonically descending relationship between the synchronization time (the time required for the system to synchronize) if oscillators influence others within a certain range of interaction (Prignano et al., 2012), but describe a nonmonotonic relationship when oscillators only influence their nearest neighbor (Prignano et al., 2013). In the first case, with a range of influence, oscillators take a long time to synchronize if they move slowly, but this time is reduced as the speed of the agents increases. This is intuitively expected as for slow speeds interactions occur only at a local level and global coherence takes time to spread, whereas at high speeds all oscillators interact with each other frequently, fostering global synchrony. On the other hand, with a nearest neighbor interaction, while the slow and fast regimes are maintained, a new regime emerges for an intermediate range of speeds, where synchronization is hindered or limited.

Chapter 3 studied the effect of a more realistic mode of interaction for practical applications: a cone of vision. Firstly, we showed in abstract simulations, where oscillators are point-like particles, that both the monotonic and nonmonotonic behaviors could be retrieved with appropriate dimensions of the cone of vision, namely its radius and angle. For small and narrow angles we found a strong nonmonotonicity. As the angle and radius of the cone of vision were increased, the intermediate regime gradually disappeared until the monotonic behavior was retrieved. This successfully linked both behaviors previously described in the literature. Secondly, we verified these results in a simplified robotic simulation, where the effects of embodiment (i.e. occlusion and collisions) are taken into account. We considered that a robot could signal a pulse to other robots by briefly flashing its onboard LEDs, which in turn could be detected by other robots in their camera field of view. We demonstrate that, despite the finite size of the robots, both behaviors could be retrieved with a cone of vision.

Chapter 4 focused on the validation of the aforementioned results on a swarm of real robots. To this purpose, the robotic simulations of the previous chapter were extended to maximize realism. Contrary to the preceding chapter, here collision avoidance was implemented, as well as a realistic reorientation algorithm. In addition, a more faithful cone of vision, taking pixels into consideration, was modeled. These simulations also served to characterize the dependence of key features of the synchronization curve on the system parameters. Good fits were found for the dependence on the number of agents, the size of the environment, the period of oscillation and the coupling strength between oscillators. This chapter also successfully validated our findings in a swarm of real e-puck robots. The nonmonotonic behavior was found for the robots' field of view, overcoming the reality gap and distinctly displaying the three dynamical regimes.

Chapter 5 further analyzed the factors that influence the time required for a system of mobile pulse-coupled oscillators to synchronize. Firstly, the effect of different phase response curves was investigated. We found that only a multiplicative response curve produced the described nonmonotonicity. In addition, we provided a suitable metric that predicts the appearance of this behavior. Secondly, we explored the effect of three neighborhood models. The impact of a cone of emission neighborhood is qualitatively equivalent to that of a cone of vision. The nonmonotonic behavior is present for short or narrow cones, and it gradually vanishes with increasing cone size. Similarly, with K-nearest neighbors connectivity we found that the nonmonotonicity disappears by increasing K. This led us to conclude that it is the average rate of neighbor change which leads to the nonmonotonic behavior. This chapter also studied the effect and propagation of perturbations on the different dynamical regimes. We observed that, for slow and fast-moving agents synchronization is resilient to perturbations. However, in the intermediate, synchronization hindering, regime, an initially completely synchronized system can be totally disrupted by a small perturbation in a single oscillator. We drew an analogy between the behavior in this intermediate regime and stable chaos. Furthermore, we found that the presence of clock jitter is catastrophic for synchronization in the intermediate regime, whereas the slow regime is moderately affected and the fast regime is resilient to it.

## 6.1 Discussion

The aim of this thesis was to further understand the synchronization dynamics of mobile pulse-coupled oscillators with a view towards applications in sensor networks and swarm robotics. As previously discussed, both synchronization and desynchronization can be beneficial for these systems. Our findings can be applied to achieve either as follows:

- In general, agents will synchronize in a short time when moving sufficiently fast, independently of the phase response curve or the neighborhood model used.

- If we are only interested in synchronization, a response curve that only increases the order parameter (see Eq. 5.8) should be selected. These include delay-advance response curves, or a multiplicative response curve with a sufficiently large refractory period (see Eq. 5.3).

- A narrow or short cone of vision (or also an interaction with a small number of nearest neighbors) can allow us to switch between synchronization and desynchronization regimes when using a multiplicative phase response curve. For instance, one can tune the system parameters so that the typical agent speed corresponds to a point in the intermediate regime that is relatively close to the fast regime. Agent speed can be momentarily increased to foster synchronization. Once this is achieved, the speed can be returned to its normal value, and synchronization will persist. If desynchronization is then desired, it would suffice to add a small perturbation to a single agent for the entire system to be desynchronized.

A shortcoming of our experiments and simulations resides in the motion dynamics. Robots move in straight lines except when avoiding other robots or reorienting at random when a wall is reached. It can be argued that this is not a realistic motion for robots in a swarm, which may move independently or coordinately to perform some task. In addition, in our simulations we assume that all agents move at the same speed. In a real scenario a distribution of speeds would be typically observed. In the experimental validation with real e-puck robots of Chapter 4, we observed that the speeds of the robots were not exactly equal among robots or even, in a few cases, between the two wheels of a single robot (producing subtle turns). However, the effect of such distribution of speeds was not explicitly analyzed.

## 6.2 Future Work

The previous section proposed a possible application of our results to a swarm of robots that allows switching between synchronization and desynchronization. Future developments could formalize such an algorithm and study its validity in a swarm of real robots.

In light of the shortcomings discussed above, future work could also study other types of motion. Firstly, different types of random walks could be explored. For instance, the study of Dimidov et al. (2016) could be followed, as it presents an in-depth study of various families of random walks in the context of target searching in swarms of robots. Secondly, an explicit study of the effect of speed distributions could be performed. It would be of interest to find whether a broad distribution fosters or hinders synchronization and if the intermediate regime still exists in this case. Finally, synchronization could be explored in the context of a practical task, such as distributed sensing, object pushing, etc.

In this thesis we found an intermediate, synchronization-hindering, regime when using a cone of vision. Previous research in swarm robotics and sensor networks has focused on a range of interaction. In the future, one could study the effects of using a cone of vision in previously reported applications. For instance, one could explore whether the fault detection algorithm proposed by Christensen et al. (2009) is still applicable in the intermediate regime.

Further research could study whether the effect of speed on synchronization observed here is translatable to other related consensus problems. Trianni et al. (2016) studied the time required for a group of robots to converge to consensus in a system of moving robots interacting with local neighbors within a certain range. This is similar to the results of synchronization in mobile pulse-coupled oscillators interaction influencing neighbors within a certain range. Whereas with an interaction range a monotonically decreasing dependence of the synchronization time on agent speed was found, a nonmonotonic behavior was reported in this thesis when a cone of vision is used. It would be worth investigating whether neighborhood type has a similar effect on the convergence time in the described consensus problem.

This work could also be extended to other robot platforms. The current method depends on LED flashing, cameras and darkness (to maximize visibility), which may not be the dominant communication method used by other robot systems. Our approach could be generalized to systems where wireless communication is the utilized signaling method.

# Appendices

# Appendix A

# Censored Data

As described in Sec. 3.1.5, any simulation exceeding $T_{cens}$ is terminated. As a consequence, we will have a record of the repetitions where $T_{sync} < T_{cens}$ but no value of $T_{sync}$ for the censored results. Despite the incompleteness of the data, we can estimate the cumulative distribution function (CDF) of the synchronization times by using the Kaplan-Meier, K-M, estimator (Kaplan and Meier, 1958).

Assuming that out of $M$ trials, there are $K$ where $T^i_{sync} < T_{cens}$, the procedure for calculating K-M estimator is as follows (Natrella, 2010):

1. Sort $T^i_{sync}$ in increasing order, from $i = 1$ to $i = K$.
2. Associate a number $n_i$ for each $T^i_{sync}$. $n_i$ is the number of trials that take longer than $T^i_{sync}$ to synchronize.
3. Calculate $R(T^1_{sync}) = (n_1 - 1)/n_1$.
4. Calculate $R(T^i_{sync}) = R(T^{i-1}_{sync})\,(n_i - 1)/n_i$.
5. The CDF is estimated as $F(T^i_{sync}) = 1 - R(T^i_{sync})$

In this way, the censored values are counted up to the latest recorded synchronization time, $T^K_{sync}$. Note that the CDF is calculated without making any assumption about the form of the probability distribution function (PDF) of the data.

By examining our results we concluded that the data best fits a Weibull distribution for all the performed simulations. By adjusting the K-M estimator to the

CDF of a Weibull distribution,

$$F(T_{sync}) = 1 - e^{-\left(\frac{T_{sync}}{\alpha}\right)^{\gamma}},$$

we obtain its two parameters, $\alpha$ and $\gamma$. Lastly, we can calculate the mean of $T_{sync}$ as

$$\langle T_{sync} \rangle = \alpha \Gamma \left(1 + \frac{1}{\gamma}\right),$$

where $\Gamma$ is the Gamma function.

The accuracy of this method decreases as the number of censored values increases. Therefore, we required a minimum of 10% of the measures to be exactly obtained in order to calculate their average. If this condition is not met then we consider the mean value as undetermined.

Figure A.1 shows how the estimation of censored data fits the uncensored points with high precision for synchronization using nearest neighbor interaction. Each violin plot shows the distribution of $T_{sync}$ for a given agent speed, $V$. The solid line shows the estimated $T_{sync}$ when setting $T_{cens} = 10^7$ as in the main text. Note that the estimated mean value can exceed the censoring threshold if enough point were censored.

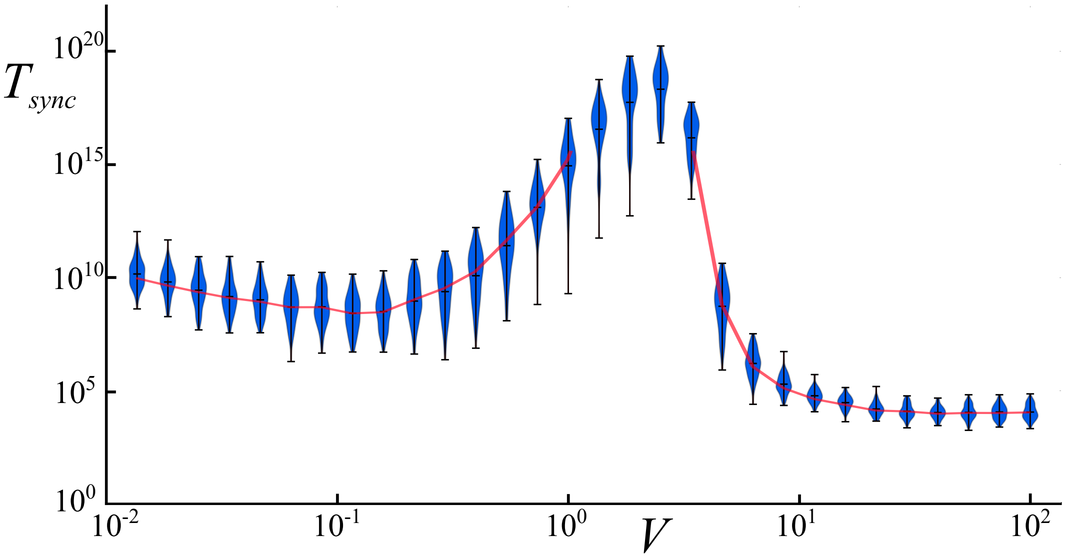


Figure A.1: Comparison of censored and uncensored data for synchronization using nearest neighbor interaction. Violin plots shows the whole uncensored distribution of synchronization times. The solid red line shows the estimation of $T_{sync}$ with censoring at $T_{cens} = 10^7$.

# Appendix B

# Motion Algorithm

The motion of the robots in Chapter 4 is calculated, in simulation and experiments, independently to the oscillator and synchronization dynamics. The following implementation is an illustration of the motion algorithm. For better readability, the code was simplified and functions have been renamed. The function `motionAlgorithm` describes the entire implementation of the algorithm and is called periodically every 25 ms.

```
int motionAlgorithm(void){
    getProximityValues();//get the values from the sensors
    calculateProxPointer();//calculate the overall vector pointing to the centeroid od the obstacle
    calculateMotorSpeed(&robot_speed);//calculate the motor speed from the vector
    //Set the motor speeds of the robot
    Sys_Set_LeftWheelSpeed(robot_speed.left);
    Sys_Set_RightWheelSpeed(robot_speed.right);
}
```

First, the sensor values of 6 sensors located on the front of the robot are obtained and stored in global buffers (`proximity_values[]`) as follows:

```
void getProximityValues(){
    int new_value;

    for(int i = 0; i < PROXIMITIES; i++){// for each proximity sensor
        if(Sys_Get_Prox(i) > 100){//measure if it is out of sensor range
            proximity_values[i] = 0;
        }else{//was something measured?
            proximity_values[i] = 100-Sys_Get_Prox(i);
        }
    }
}
```

Subsequently, the values of the directed proximity sensors are transformed into vectors where its magnitude is an indicator how close the measured obstacle is. These vectors are then summed to a vector that points towards the centroid of the measured obstacles.

```
void calculateProxPointer(){
    int prox_x = 0, prox_y = 0;
    int i,j;
    for(i = 5; i < 11; i++){ // sums all sensors from the front half
        j = i % 8;
        if(proximity_values[j] > 70){// if obstacle is close
            prox_x += proximity_values[j] * angle_component_x[j];//calculate
                the X component
            prox_y += proximity_values[j] *
                angle_component_y[j]);//calculate the Y component
        }
    }
    //store in global buffer
    proximity_pointer.x = prox_x;
    proximity_pointer.y = prox_y;
}
```

The resulting vector is then used to calculate the wheel speeds and therefore the movement of the robot. The robot follows the following rules: Move forward when no obstacle is detected. If an obstacle was detected on the left side, then rotate to the right side and vice versa. If the robot detected for a long time an obstacle in its surroundings, the robot rotates on the spot for a random amount of time to avoid potential life-locks. This effectively implements obstacle avoidance and random reorientation at walls.

```
void calculateMotorSpeed(motor_speeds *speeds){
    static int rotTime = 0;  // counts number of consecutive cycles where
        robot was rotating
    static int proxCount = 0; // counts the cycles that the robot is close
        to an object

    int vel = MAX_SPEED;     // robots velocity
    int max = MAX_SPEED;     // 128 mm/s (maximum possible velocity)
    int rotDir = 0;          // -1 for right and +1 for left

    if(rotTime > 0 || proxCount > 0){ // if the robot is rotating or
        detected something
        // is the X or Y component above threshold?
        if( abs(proximity_pointer.x) > proxThres ||
            abs(proximity_pointer.y) > proxThres ){
            proxCount++;//Something was measured for an additional timestep
        } else {
            proxCount = 0;
        }
    }

    // If obstacles obstacle was right or left
    if(proximity_pointer.y > 0){
        rotDir = 1; //left
    } else {
        rotDir = -1; //right
    }

    // If the robot was close to an obstacle for too long (>
        proxCountThres) then turn away (for a random amount of time)
    if(proxCount > proxCountThres){
        speeds->left = - max*rotDir;
        speeds->right = max*rotDir;
        rotTime += 1 + (rand() % rotMax);
        proxCount = 0;
    }

     // If there is an obstacle ahead turn slightly away
    if(proximity_pointer.x > proxThres){
        if(proxCount == 0){//if you haven't rotated before
            speeds->left = -max*rotDir;
            speeds->right = max*rotDir;
        } else { // use lowpass to avoid life-lock by toggling direction
            due to noise
            speeds->left = speeds->left/2 - max*rotDir/2;
            speeds->right = speeds->right/2 + max*rotDir/2;
        }
        rotTime = 1;
    } else {// if there is no obstacle
        if(rotTime == 0) { // go forward
            speeds->right = speeds->left = vel;
        } else {// or continue last action until the robot finished rotating
            rotTime -= 1;
        }
    }
}
```

# Appendix C

# Synchronization Threshold

In Section 4.3 we claim that a standard deviation of 2 frames (approx. 130 ms) in the robotic experiments is approximately equivalent to $r_{sync} = 0.95$ as set for the simulation experiments. In this appendix we present a proof.

We start by considering the distribution of instantaneous firings ($\phi_{LED} \to 0$) corresponding to the lowest order parameter, $r$, for a system of $N$ oscillators firing within a certain time interval, $[-\sigma_A, \sigma_A]$ ($\sigma_A \leq \tau/2$). This is displayed in Fig. C.1(a): half of the firings occur at $t = -\sigma_A$ and the other half occurs at $t = +\sigma_A$. The standard deviation of this distribution is, thus, exactly $\sigma_A$.

Remember that the complex order parameter is calculated as follows,

$$r(T_k)e^{\mathrm{i}\frac{2\pi\phi(T_k)}{\tau}} = \frac{1}{N}\sum_{j=1}^{N} e^{\mathrm{i}\frac{2\pi\phi_j(T_k)}{\tau}} . \tag{C.1}$$

Solving (C.1) for the standard deviation, $\sigma_A$, corresponding to $r = 0.95$ in the described scenario,

$$0.95 = \frac{1}{N}\frac{N}{2}\left(e^{-\mathrm{i}\frac{2\pi\sigma_A}{\tau}} + e^{\mathrm{i}\frac{2\pi\sigma_A}{\tau}}\right) = \cos\left(\frac{\tau}{2\pi\sigma_A}\right) , \tag{C.2}$$

yields $\sigma_A = 0.25\,\mathrm{s}$ for $\tau = 5\,\mathrm{s}$ as used in the robot experiments. At the frame rate used for tracking (approximately 15-16 FPS) it corresponds to $\sigma_A \approx 3.8\,$frames.

In reality, however, firings are not instantaneous, but the LEDs are turned on for a period of time. In our experiments, $\phi_{LED} = 0.075\tau = 0.375\,\mathrm{s} \approx 5.7$frames.

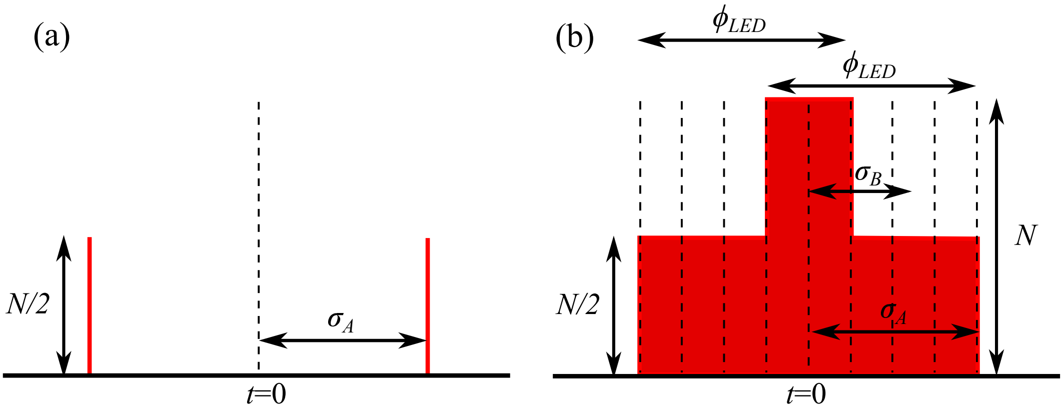


Figure C.1: Distribution of firings yielding the lowest synchronization for (a) instantaneous firings, and (b) non-instantaneous firings.

Now we consider the distribution of non-instantaneous firings that corresponds to the worst synchronization of $N$ oscillators firing within time interval $[-\sigma_A, \sigma_A]$. If we count the number of agents that are flashing at any particular frame, the worst synchronization corresponds to the greatest standard deviation, $\sigma_B$ (see Fig. C.1(b)). Naturally, $\sigma_B \leq \sigma_A$.

Approximating $\sigma_A = 4$ frames and $\phi_{LED} = 5$ frames to minimize skewness (as in Fig. C.1(b)), we can obtain the upper limit of $\sigma_B$:

$$\sigma_B \leq \sqrt{\frac{2}{5N}\left(\frac{N}{2}(4^2+3^2+2^2)+1^1\right)} \approx 2.49\,\text{frames}, \qquad \text{(C.3)}$$

which is consistent with the choice of synchronization threshold (2 frames) set in our experiments.

# Appendix D

# Additional Results

This appendix presents additional results that complement those of the main text.

## D.1 Motion Dynamics Coupled to Interactions

Chapters 3, 4 and 5 considered agents moving in straight lines that reorient at random when approaching the environment boundaries. Additionally, embodied agents (robots) avoided collisions with other agents in Chapter 4. In general, the motion dynamics considered in the main text are independent of the oscillator dynamics or interactions between oscillators. In this section, we present the effect of three other controllers, in which the motion dynamics is coupled to the interactions between agents. In particular, we consider the same movement as before (i.e. straight lines and random reorientation at boundaries) with additional random reorientation of an agent upon emitting a pulse, receiving a pulse from another agent, or both emitting or receiving a pulse.

Synchronization curves using K-nearest neighbor interaction are presented in Figures D.1, D.2 and D.3 for the motion controllers with reorientation upon emitting a pulse, receiving a pulse, and both emitting or receiving a pulse, respectively. The resulting behavior in all cases is qualitatively similar to that presented in Chapter 5 (cf. Fig. 5.5). A nonmonotonic behavior is found for low $K$ and, as $K$ increases the monotonic behavior is retrieved.

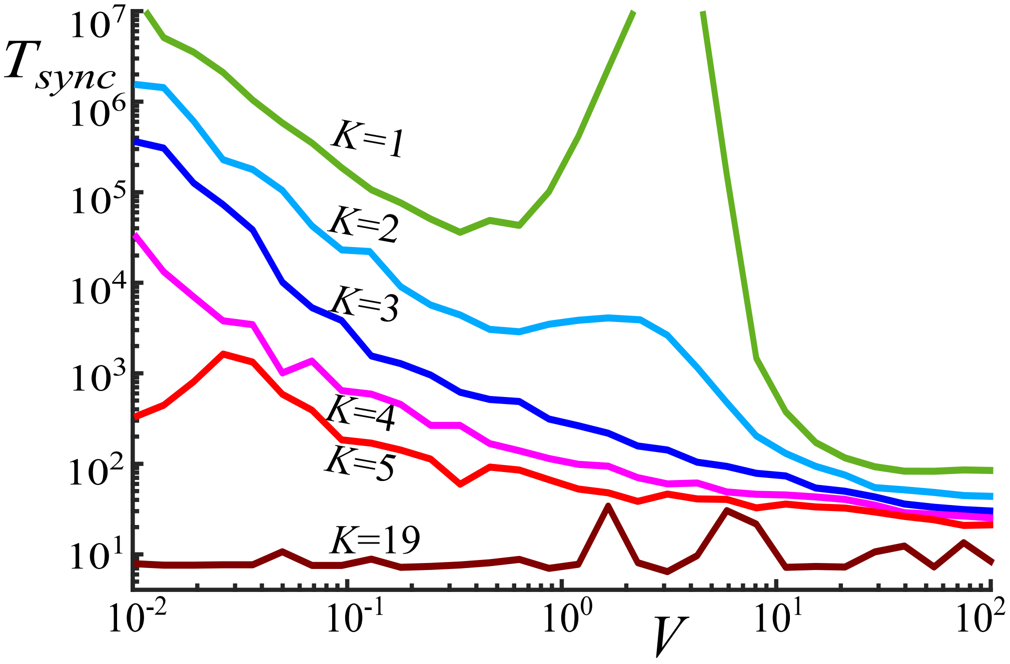


Figure D.1: Time $T_{sync}$ (in number of cycles) required to synchronize 20 units with K-nearest neighbors connectivity and moving with speed $V$ in a 2D square environment of side 100, and reorienting upon emitting a pulse.

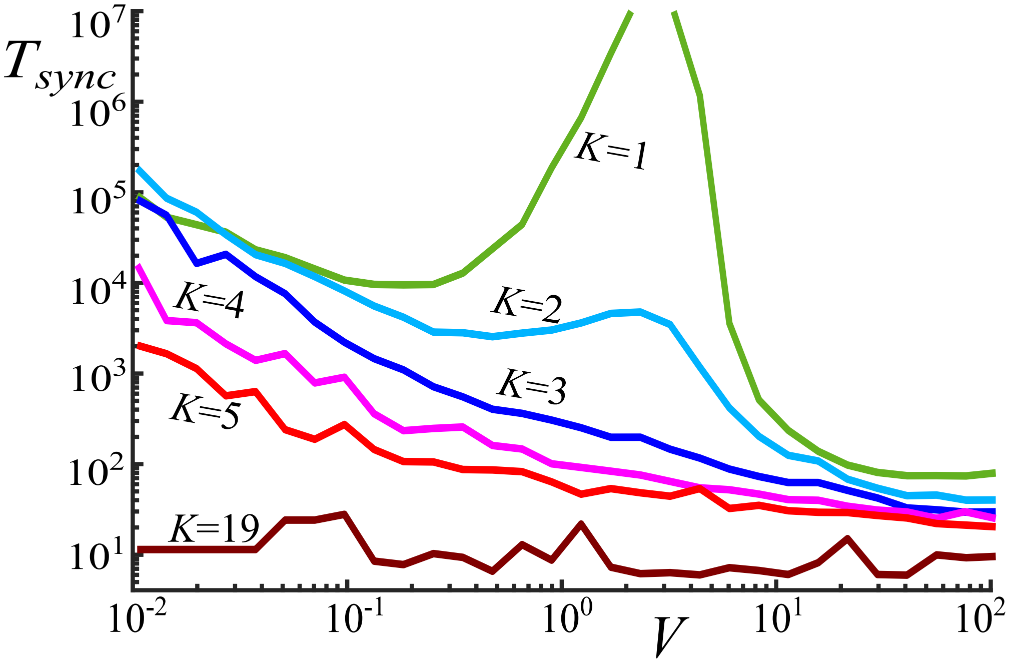


Figure D.2: Time $T_{sync}$ (in number of cycles) required to synchronize 20 units with K-nearest neighbors connectivity and moving with speed $V$ in a 2D square environment of side 100, and reorienting upon receiving a pulse.

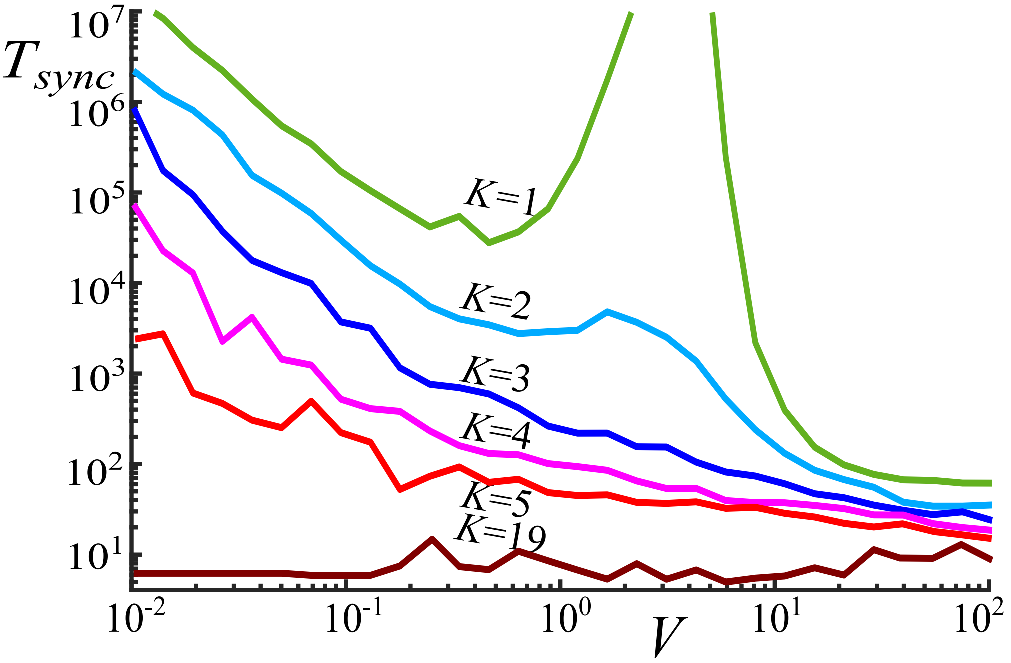


Figure D.3: Time $T_{sync}$ (in number of cycles) required to synchronize 20 units with K-nearest neighbors connectivity and moving with speed $V$ in a 2D square environment of side 100, and reorienting upon emitting or receiving a pulse.

## D.2 Cyclic Environment

Bounded environments were considered in the main text for practical realism whereas some previous approaches in the literature (Prignano et al., 2012, 2013; Wang et al., 2015) opted for periodic boundary conditions.

In this section we study the synchronization of agents using a cone of vision and moving in a cyclic square environment. That is, an environment with periodic boundary conditions, in which an oscillator crossing a boundary on one side of the environment will appear on the opposite side. This is topologically equivalent to a torus. Figure D.4 shows that the resulting behavior is qualitatively equivalent to those in Chapter 5 (cf. Fig. 5.6). A nonmonotonic behavior is found for short or narrow cones of vision, and the nonmonotonicity gradually vanishes with increasing cone dimensions. Note that in Fig. D.4 no reorientation of the units takes place.

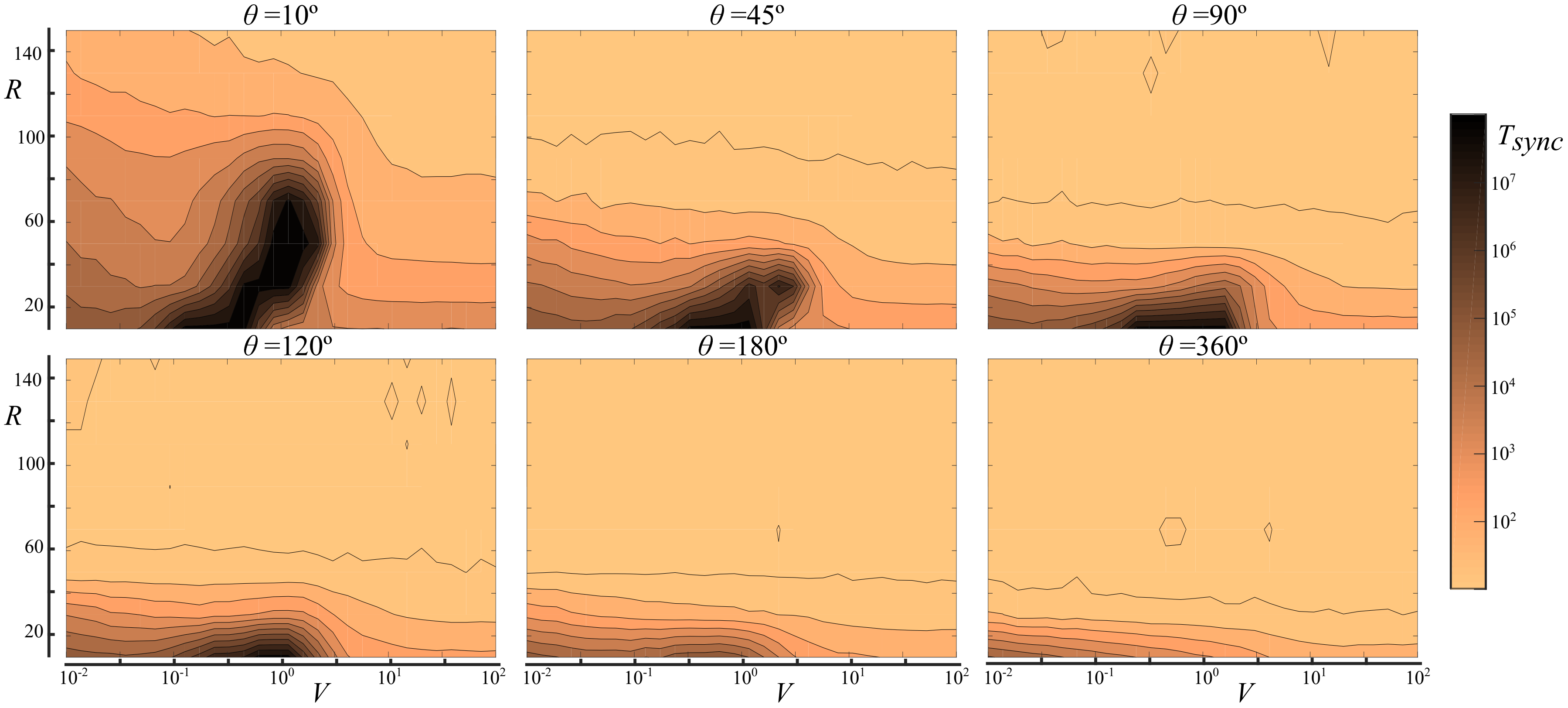


Figure D.4: Time $T_{sync}$ (in number of cycles) required to synchronize 20 units with cone of vision connectivity (with radius $R$ and angle $\theta$) and moving with speed $V$ in a cyclic 2D square environment of side 100 using the multiplicative PRC.

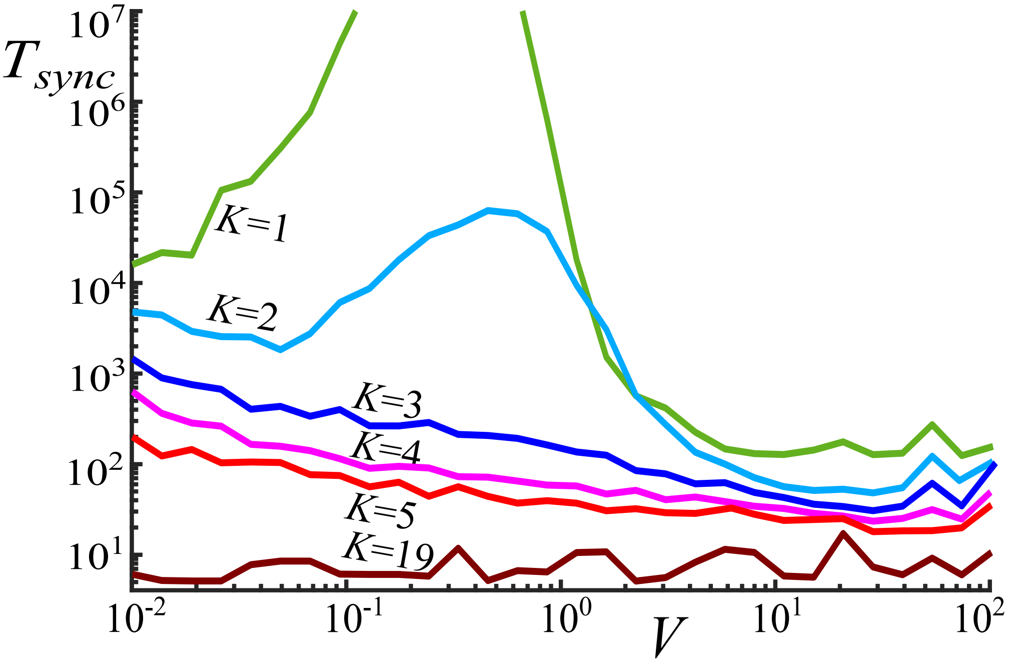


Figure D.5: Time $T_{sync}$ (in number of cycles) required to synchronize 20 units with K-nearest neighbors connectivity and moving with speed $V$ in a 1D segment of side 100, and reorienting upon emitting a pulse.

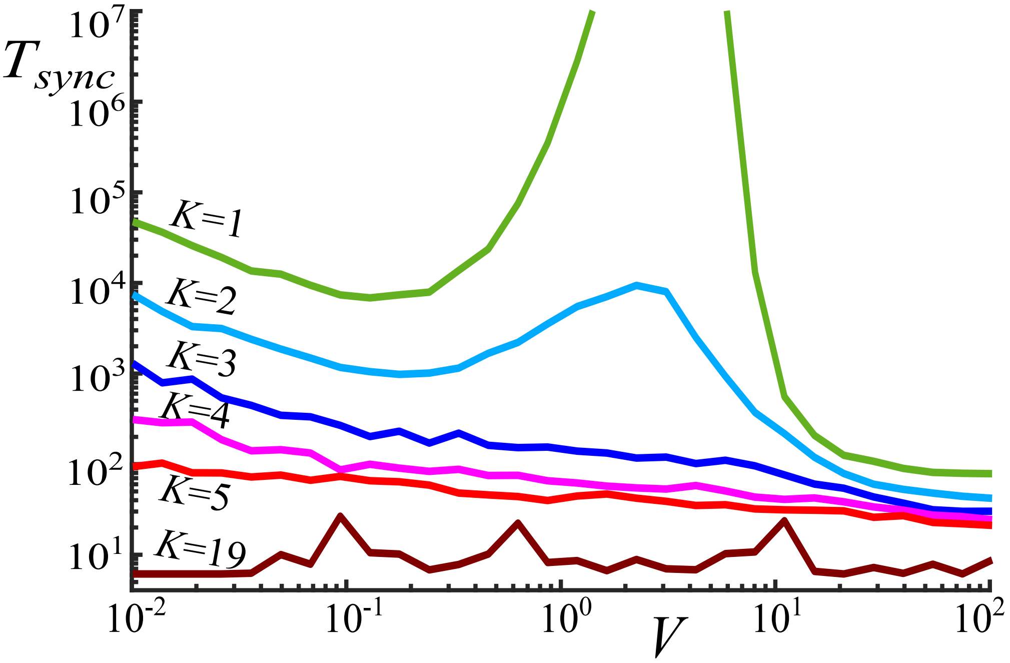


Figure D.6: Time $T_{sync}$ (in number of cycles) required to synchronize 20 units with K-nearest neighbors connectivity and moving with speed $V$ in a 3D cubic environment of side 100, and reorienting upon emitting a pulse.

## D.3 Beyond Two Dimensions

All the results presented in this thesis and all the literature in MPCOs synchronization are concerned with agents moving in a two-dimensional environments. This section show that our findings are also applicable to agents moving in spaces of different dimensionality. The synchronization curves for agents interacting with their K-nearest neighbor in a one-dimensional (1D) segment and a three-dimensional cubic environment are shown in Figures D.5 and D.6, respectively. The results are qualitatively equivalent to those of Chapter 5 (cf. Fig. 5.5). Note that the considered environments are bounded and random reorientation occurs at the environment boundary. In the 1D environment only reorientation in the opposite direction is possible.

# Bibliography

Akyildiz, I. F. and Kasimoglu, I. H. (2004). Wireless sensor and actor networks: research challenges. *Ad Hoc Networks*, 2(4):351–367.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002a). Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002b). A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114.

An, Z., Zhu, H., Li, X., Xu, C., Xu, Y., and Li, X. (2011). Nonidentical linear pulse-coupled oscillators model with application to time synchronization in wireless sensor networks. *IEEE Transactions on Industrial Electronics*, 58(6):2205–2215.

Arenas, A., Díaz-Guilera, A., Kurths, J., Moreno, Y., and Zhou, C. (2008). Synchronization in complex networks. *Physics Reports*, 469(3):93–153.

Beni, G. (2005). From swarm intelligence to swarm robotics. *Swarm Robotics, Lecture Notes in Computer Science*, 3342:1–9.

Berdan, R., Prodromakis, T., Khiat, A., Salaoru, I., Toumazou, C., Perez-Diaz, F., and Vasilaki, E. (2013). Temporal processing with volatile memristors. In *2013 IEEE International Symposium on Circuits and Systems*, pp. 425–428. IEEE.

Blum, C. and Groß, R. (2015). Swarm intelligence in optimization and robotics. In *Springer Handbook of Computational Intelligence*, pp. 1291–1309. Springer.

Boccaletti, S., Kurths, J., Osipov, G., Valladares, D., and Zhou, C. (2002). The synchronization of chaotic systems. *Physics Reports*, 366(1):1–101.

Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems.* Oxford University Press.

Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2013). Swarm robotics: A review from the swarm engineering perspective. *Swarm Intelligence*, 7(1): 1–41.

Buck, J. (1988). Synchronous rhythmic flashing of fireflies. II. *Quarterly Review of Biology*, pp. 265–289.

Buranapanichkit, D., Deligiannis, N., and Andreopoulos, Y. (2015). Convergence of desynchronization primitives in wireless sensor networks: A stochastic modeling approach. *IEEE Transactions on Signal Processing*, 63(1):221–233.

Carbone, P., Cazzorla, A., Ferrari, P., Flammini, A., Moschitta, A., Rinaldi, S., Sauter, T., and Sisinni, E. (2013). Low complexity UWB radios for precise wireless sensor network synchronization. *IEEE Transactions on Instrumentation and Measurement*, 62(9):2538–2548.

Castillo-Cagigal, M., Brutschy, A., Gutiérrez, A., and Birattari, M. (2014). Temporal task allocation in periodic environments. *Swarm Intelligence, Lecture Notes in Computer Science*, 8667:182–193.

Chevallier, S., Bredeche, N., Paugam-Moisy, H., and Sebag, M. (2011). Emergence of temporal and spatial synchronous behaviors in a foraging swarm. In *2011 European Conference on Artificial Life*.

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

Cornejo, A. and Kuhn, F. (2010). Deploying wireless networks with beeps. In *International Symposium on Distributed Computing*, pp. 148–162. Springer.

Corral, Á., Pérez, C. J., Diaz-Guilera, A., and Arenas, A. (1995a). Synchronization in a lattice model of pulse-coupled oscillators. *Physical Review Letters*, 75(20): 3697.

Corral, Á., Pérez, C. J., Díaz-Guilera, A., and Arenas, A. (1995b). Self-organized criticality and synchronization in a lattice model of integrate-and-fire oscillators. *Physical Review Letters*, 74(1):118.

Cortés, J., Martínez, S., and Bullo, F. (2006). Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Transactions on Automatic Control*, 51(8):1289–1298.

Degesys, J., Rose, I., Patel, A., and Nagpal, R. (2007). Desync: Self-organizing desynchronization and TDMA on wireless sensor networks. In *Proceedings of the 6th International Conference on Information Processing in Sensor Networks*, pp. 11–20. ACM.

Deligeorges, S., Cakiades, G., George, J., Wang, Y., and Doyle, F. (2015a). A mobile self synchronizing smart sensor array for detection and localization of impulsive threat sources. In *2015 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 351–356. IEEE.

Deligeorges, S., Lavey, C., Cakiades, G., George, J., Wang, Y., Ez, F., and Doyle, F. (2015b). A mobile acoustic sensor fusion network using biologically inspired sensors and synchronization. In *18th International Conference on Information Fusion*, pp. 1717–1723. IEEE.

Dimidov, C., Oriolo, G., and Trianni, V. (2016). Random walks in swarm robotics: An experiment with kilobots. In *International Conference on Swarm Intelligence*, pp. 185–196. Springer.

Dorigo, M., Trianni, V., Şahin, E., Groß, R., Labella, T. H., Baldassarre, G., Nolfi, S., Deneubourg, J.-L., Mondada, F., Floreano, D., et al. (2004). Evolving self-organizing behaviors for a swarm-bot. *Autonomous Robots*, 17(2-3):223–245.

Doyle, M. J., Xu, X., Gu, Y., Perez-Diaz, F., Parrott, C., and Groß, R. (2016). Modular Hydraulic Propulsion: A robot that moves by routing fluid through itself. In *2016 IEEE International Conference on Robotics and Automation*, pp. 5189–5196. IEEE.

Eccles, W. and Vincent, J. (1920). British patent spec. CLXIII. p. 462.

Ermentrout, B. (1991). An adaptive model for synchrony in the firefly pteroptyx malaccae. *Journal of Mathematical Biology*, 29(6):571–585.

Fister, I., Yang, X.-S., and Brest, J. (2013). A comprehensive review of firefly algorithms. *Swarm and Evolutionary Computation*, 13:34–46.

Fujiwara, N., Kurths, J., and Díaz-Guilera, A. (2011). Synchronization in networks of mobile oscillators. *Physical Review E*, 83(2):025101.

Fujiwara, N., Kurths, J., and Díaz-Guilera, A. (2016). Synchronization of mobile chaotic oscillator networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26(9):094824.

Gauci, M., Chen, J., Li, W., Dodd, T. J., and Groß, R. (2014a). Clustering objects with robots that do not compute. In *Proceedings of AAMAS 2014*, pp. 421–428.

Gauci, M., Chen, J., Li, W., Dodd, T. J., and Groß, R. (2014b). Self-organized aggregation without computation. *The International Journal of Robotics Research*, p. 0278364914525244.

Gómez-Gardeñes, J., Nicosia, V., Sinatra, R., and Latora, V. (2013). Motion-induced synchronization in metapopulations of mobile agents. *Phys. Rev. E*, 87:032814.

Hartbauer, M. and Römer, H. (2007). A novel distributed swarm control strategy based on coupled signal oscillators. *Bioinspiration & Biomimetics*, 2(3):42.

Hong, Y.-W. and Scaglione, A. (2005). A scalable synchronization protocol for large scale sensor networks and its applications. *IEEE Journal on Selected Areas in Communications*, 23(5):1085–1099.

Horvath, V., Gentili, P. L., Vanag, V. K., and Epstein, I. R. (2012). Pulse-coupled chemical oscillators with time delay. *Angewandte Chemie International Edition*, 51(28):6878–6881.

Horvath, V., Kutner, D. J., Chavis III, J. T., and Epstein, I. R. (2015). Pulse-coupled BZ oscillators with unequal coupling strengths. *Physical Chemistry Chemical Physics*, 17(6):4664–4676.

Huygens, C. (1673). Horologium oscillatorium. Apud F. Muguet, Paris, France. English translation: The pendulum clock. Iowa State University Press, Ames, 1986.

Izhikevich, E. M. (1999). Weakly pulse-coupled oscillators, FM interactions, synchronization, and oscillatory associative memory. *IEEE Transactions on Neural Networks*, 10(3):508–526.

Janagal, L. and Parmananda, P. (2012). Synchronization in an ensemble of spatially moving oscillators with linear and nonlinear coupling schemes. *Phys. Rev. E*, 86:056213.

Kaplan, E. L. and Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53(282):457–481.

Khaluf, Y., Mathews, E., and Rammig, F. J. (2011). Self-organized cooperation in swarm robotics. In *14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops*, pp. 217–226. IEEE.

Kuramoto, Y. (1975). Self-entrainment of a population of coupled non-linear oscillators. In *International Symposium on Mathematical Problems in Theoretical Physics*, pp. 420–422. Springer.

Kuramoto, Y. (1991). Collective synchronization of pulse-coupled oscillators and excitable units. *Physica D: Nonlinear Phenomena*, 50(1):15–30.

Laurent, P. (1917). The supposed synchronal flashing of fireflies. *Science*, 45(1150): 44–44.

Levis, D., Pagonabarraga, I., and Diaz-Guilera, A. (2016). Synchronization in dynamical networks of locally coupled self-propelled oscillators. *arXiv preprint arXiv:1608.02423*.

Lin, J., Morse, A., and Anderson, B. (2007). The multi-agent rendezvous problem. Part 1: The synchronous case. *SIAM Journal on Control and Optimization*, 46 (6):2096–2119.

Magnenat, S., Waibel, M., and Beyeler, A., (2007). Enki: The fast 2D robot simulator. `http://home.gna.org/enki/`.

Marella, S. and Ermentrout, G. B. (2008). Class-II neurons display a higher degree of stochastic synchronization than class-I neurons. *Physical review E*, 77(4): 041918.

Mauroy, A. and Sepulchre, R. (2008). Clustering behaviors in networks of integrate-and-fire oscillators. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 18(3):037122.

Melhuish, C., Holland, O., and Hoddell, S. (1999). Convoying: Using chorusing to form travelling groups of minimal agents. *Robotics and Autonomous Systems*, 28(2):207–216.

Mirollo, R. E. and Strogatz, S. H. (1990). Synchronization of pulse-coupled biological oscillators. *SIAM Journal on Applied Mathematics*, 50(6):1645–1662.

Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J.-C., Floreano, D., and Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, Volume 1, pp. 59–65. IPCB: Instituto Politécnico de Castelo Branco.

Murray, J. D. (2002). In *Mathematical Biology I: An Introduction*, Volume 17 of *Interdisciplinary Applied Mathematics*. Springer.

Natrella, M. (2010). Nist/sematech e-handbook of statistical methods. NIST/SEMATECH Available online at http://www.itl.nist.gov/div898/handbook/.

Nouyan, S., Groß, R., Bonani, M., Mondada, F., and Dorigo, M. (2009). Teamwork in self-organized robot colonies. *IEEE Transactions on Evolutionary Computation*, 13(4):695–711.

Núñez, F., Wang, Y., and Doyle, F. J. (2015). Global synchronization of pulse-coupled oscillators interacting on cycle graphs. *Automatica*, 52:202–209.

Nymoen, K., Chandra, A., and Torresen, J. (2013). The challenge of decentralised synchronisation in interactive music systems. In *IEEE 7th International Conference on Self-Adaptation and Self-Organizing Systems Workshops*, pp. 95–100. IEEE.

Pagliari, R. and Scaglione, A. (2011). Scalable network synchronization with pulse-coupled oscillators. *IEEE Transactions on Mobile Computing*, 10(3):392–405.

Parker, L. E. (2008). Multiple mobile robot systems. In *Springer Handbook of Robotics*, pp. 921–941. Springer.

Patel, A., Degesys, J., and Nagpal, R. (2007). Desynchronization: The theory of self-organizing algorithms for round-robin scheduling. In *First International Conference on Self-Adaptive and Self-Organizing Systems*, pp. 87–96. IEEE.

Pazó, D. and Montbrió, E. (2014). Low-dimensional dynamics of populations of pulse-coupled oscillators. *Physical Review X*, 4(1):011009.

Penrose, M. (2003). Random geometric graphs. Number 5 in Oxford Studies in Probability. Oxford University Press.

Perez-Diaz, F., Zillmer, R., and Groß, R. (2015). Firefly-inspired synchronization in swarms of mobile agents. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pp. 279–286. IFAAMAS.

Perez-Diaz, F., Zillmer, R., and Groß, R. (2016a). Emergence and inhibition of synchronization in robot swarms. In *Proceedings of the 13th International Symposium on Distributed Autonomous Robotic Systems (to appear)*. Springer.

Perez-Diaz, F., Zillmer, R., and Groß, R., (2016b). Emergence and inhibition of synchronization in robot swarms. Online supplementary material. `http://naturalrobotics.group.shef.ac.uk/supp/2016-004`.

Perez-Diaz, F., Zillmer, R., and Groß, R. (unpublished). Control of synchronization regimes in networks of mobile interacting agents. *Physical Review Applied (submitted)*.

Peskin, C. S. (1975). *Mathematical aspects of heart physiology.* Courant Institute of Mathematical Sciences, New York University.

Pikovsky, A., Rosenblum, M., and Kurths, J. (2001). *Synchronization: A Universal Concept in Nonlinear Sciences.* Cambridge University Press.

Politi, A. and Torcini, A. *Nonlinear Dynamics and Chaos: Advances and Perspectives*, chapter Stable Chaos, pp. 103–129. Springer, (2010).

Prignano, L., Sagarra, O., Gleiser, P. M., and Diaz-Guilera, A. (2012). Synchronization of moving integrate and fire oscillators. *International Journal of Bifurcation and Chaos*, 22(07):1250179.

Prignano, L., Sagarra, O., and Díaz-Guilera, A. (2013). Tuning synchronization of integrate-and-fire oscillators through mobility. *Physcal Review Letters*, 110: 114101.

Proskurnikov, A. V. and Cao, M. (2015). Synchronization of pulse-coupled oscillators and clocks under minimal connectivity assumptions. *arXiv preprint arXiv:1510.02338.*

Ranganathan, P., Morton, R., Richardson, A., Strom, J., Goeddel, R., Bulic, M., and Olson, E. (2010). Coordinating a team of robots for urban reconnaissance. In *Proceedings of the Land Warfare Conference.*

Rayleigh, J. W. S. B. (1896). *The theory of sound.* Macmillan.

Ren, W., Beard, R. W., and Atkins, E. M. (2005). A survey of consensus problems in multi-agent coordination. In *Proceedings of the American Control Conference*, pp. 1859–1864.

Ren, W. et al. (2007). Information consensus in multivehicle cooperative control. *IEEE Control Systems Magazine*, 27(2):71–82.

Richard, P., Bakker, B. M., Teusink, B., Dam, K., and Westerhoff, H. V. (1996). Acetaldehyde mediates the synchronization of sustained glycolytic oscillations in populations of yeast cells. *European Journal of Biochemistry*, 235(1-2):238–241.

Rosenblum, M. and Pikovsky, A. (2003). Synchronization: From pendulum clocks to chaotic lasers and chemical oscillators. *Contemporary Physics*, 44(5):401–416.

Şahin, E. (2004). Swarm robotics: From sources of inspiration to domains of application. In *International Workshop on Swarm Robotics*, pp. 10–20. Springer.

Sarkar, S. and Parmananda, P. (2010). Synchronization of an ensemble of oscillators regulated by their spatial movement. *Chaos*, 20(4):043108.

Scaglione, A. and Hong, Y.-W. (2003). Opportunistic large arrays: Cooperative transmission in wireless multihop ad hoc networks to reach far distances. *IEEE Transactions on Signal Processing*, 51(8):2082–2092.

Sepulchre, R., Paley, D. A., and Leonard, N. E. (2007). Stabilization of planar collective motion: All-to-all communication. *IEEE Transactions on Automatic Control*, 52(5):811–824.

Shen, C., Chen, H., and Hou, Z. (2013). Mobility-enhanced signal response in metapopulation networks of coupled oscillators. *Europhysics Letters*, 102(3): 38004.

Shen, C., Chen, H., and Hou, Z. (2014). Mobility and density induced amplitude death in metapopulation networks of coupled oscillators. *Chaos*, 24(4):043125.

Silva, F., Correia, L., and Christensen, A. L. (2015). Modelling synchronisation in multirobot systems with cellular automata: Analysis of update methods and topology perturbations. In *Robots and Lattice Automata*, pp. 267–293. Springer.

Simeone, O., Spagnolini, U., Bar-Ness, Y., and Strogatz, S. H. (2008). Distributed synchronization in wireless networks. *IEEE Signal Processing Magazine*, 25(5): 81–97.

Sperati, V., Trianni, V., and Nolfi, S. (2008). Evolving coordinated group behaviours through maximisation of mean mutual information. *Swarm Intelligence*, 2(2-4): 73–95.

Stam, C. v. and Van Straaten, E. (2012). The organization of physiological brain networks. *Clinical Neurophysiology*, 123(6):1067–1087.

Strogatz, S. (2003). *Sync: The Emerging Science of Spontaneous Order*. Hyperion, New York.

Strogatz, S. H. (2014). *Nonlinear dynamics and chaos: With applications to physics, biology, chemistry, and engineering*. Westview Press.

Sutantyo, D. and Levi, P. (2013). A bio-inspired TDMA scheduling algorithm for underwater robotic swarms. In *IEEE International Conference on Robotics and Biomimetics*, pp. 1107–1112. IEEE.

Sutantyo, D. and Levi, P. (2015). Decentralized underwater multi-robot communication using bio-inspired approaches. *Artificial Life and Robotics*, 20(2):152–158.

Tateno, T. and Robinson, H. (2007). Phase resetting curves and oscillatory stability in interneurons of rat somatosensory cortex. *Biophysical Journal*, 92(2):683–695.

Trenkwalder, S., Lopes, Y., Kolling, A., Christensen, A., Prodan, R., and Groß, R. (2016). Openswarm: An event-driven embedded operating system for miniature robots. In *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (to appear)*.

Trianni, V. and Campo, A. (2015). Fundamental collective behaviors in swarm robotics. In *Springer Handbook of Computational Intelligence*, pp. 1377–1394. Springer.

Trianni, V. and Nolfi, S. (2007). Minimal communication strategies for self-organising synchronisation behaviours. In *2007 IEEE Symposium on Artificial Life*, pp. 199–206. IEEE.

Trianni, V. and Nolfi, S. (2009). Self-organizing sync in a robotic swarm: A dynamical system view. *IEEE Transactions on Evolutionary Computation*, 13(4): 722–741.

Trianni, V. and Nolfi, S. (2010). Re-engineering evolution-a study in self-organising synchronisation. In *Proceedings of the Alife XII Conferenc*, pp. 561–568.

Trianni, V. and Nolfi, S. (2011). Engineering the evolution of self-organizing behaviors in swarm robotics: A case study. *Artificial Life*, 17(3):183–202.

Trianni, V., De Simone, D., Reina, A., and Baronchelli, A. (2016). Emergence of consensus in a multi-robot network: From abstract models to empirical validation. *IEEE Robotics and Automation Letters*, 1(1):348–353.

Tyrrell, A., Auer, G., and Bettstetter, C. (2006). Fireflies as role models for synchronization in ad hoc networks. In *Proceedings of the 1st International Conference on Bio Inspired Models of Network, Information and Computing Systems*. ACM.

Tyrrell, A., Auer, G., and Bettstetter, C. (2007). Biologically inspired synchronization for wireless networks. In *Advances in Biologically Inspired Information Systems*, pp. 47–62. Springer.

Uriu, K., Ares, S., Oates, A. C., and Morelli, L. G. (2013). Dynamics of mobile coupled phase oscillators. *Physical Review E*, 87:032911.

van Vreeswijk, C. (1996). Partial synchronization in populations of pulse-coupled oscillators. *Physical Review E*, 54(5):5522.

Vig, L. and Adams, J. A. (2006). Multi-robot coalition formation. *IEEE Transactions on Robotics*, 22(4):637–649.

Wang, J., Xu, C., Feng, J., Chen, M. Z., Wang, X., and Zhao, Y. (2015). Synchronization in moving pulse-coupled oscillator networks. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 62(10):2544–2554.

Wang, Y. and Doyle III, F. J. (2012). Optimal phase response functions for fast pulse-coupled synchronization in wireless sensor networks. *IEEE Transactions on Signal Processing*, 60(10):5583–5588.

Wang, Y., Núñez, F., and Doyle III, F. J. (2012). Energy-efficient pulse-coupled synchronization strategy design for wireless sensor networks through reduced idle listening. *IEEE Transactions on Signal Processing*, 60(10):5293–5306.

Wang, Y., Nunez, F., and Doyle, F. J. (2013). Statistical analysis of the pulse-coupled synchronization strategy for wireless sensor networks. *IEEE Transactions on Signal Processing*, 61(21):5193–5204.

Wang, Y., Núñez, F., and Doyle III, F. J. (2014). Mobility induced network evolution speeds up synchronization of wireless sensor networks. In *Proceedings of the American Control Conference*, pp. 3553–3558. IEEE.

Wang, Y., Maxwell, T., Bear, E., and Anglea, T. (2016). A unified communication and control approach for decentralized heading alignment in robot networks. In *2016 Chinese Control and Decision Conference*, pp. 2548–2553. IEEE.

Werner-Allen, G., Tewari, G., Patel, A., Welsh, M., and Nagpal, R. (2005). Firefly-inspired sensor network synchronicity with realistic radio effects. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pp. 142–153. ACM.

Winfield, A. F. (2000). Distributed sensing and data collection via broken ad hoc wireless connected networks of mobile robots. In *Distributed Autonomous Robotic Systems 4*, pp. 273–282. Springer.

Winfree, A. T. (1967). Biological rhythms and the behavior of populations of coupled oscillators. *Journal of Theoretical Biology*, 16(1):15–42.

Winfree, A. T. (2001). *The geometry of Biological Time.* Springer.

Wischmann, S. and Pasemann, F. (2006). The emergence of communication by evolving dynamical systems. In *International Conference on Simulation of Adaptive Behavior*, pp. 777–788. Springer.

Wischmann, S., Hülse, M., Knabe, J. F., and Pasemann, F. (2006). Synchronization of internal neural rhythms in multi-robotic systems. *Adaptive Behavior*, 14(2): 117–127.

Wu, Y.-C., Chaudhari, Q., and Serpedin, E. (2011). Clock synchronization of wireless sensor networks. *IEEE Signal Processing Magazine*, 28(1):124–138.

Yick, J., Mukherjee, B., and Ghosal, D. (2008). Wireless sensor network survey. *Computer networks*, 52(12):2292–2330.

Yu, C., Werfel, J., and Nagpal, R. (2010). Collective decision-making in multi-agent systems by implicit leadership. In *Proceedings of the 2010 International Conference on Autonomous Agents and Multiagent Systems*, pp. 1189–1196. IFAAMAS.

Zhang, Z., Long, K., Wang, J., and Dressler, F. (2014). On swarm intelligence inspired self-organized networking: Its bionic mechanisms, designing principles and optimization approaches. *IEEE Communications Surveys & Tutorials*, 16 (1):513–537.

Zillmer, R., Brunel, N., and Hansel, D. (2009). Very long transients, irregular firing, and chaotic dynamics in networks of randomly connected inhibitory integrate-and-fire neurons. *Physical Review E*, 79(3):031909.