Firefly-Inspired Synchronization in Swarms of Mobile Agents

Fernando Perez-Diaz
Natural Robotics Lab
Sheffield Robotics &
Department of Computer Science
The University of Sheffield, UK
fernando.perez.diaz@sheffield.ac.uk

Ruediger Zillmer
Unilever R&D, Port Sunlight, UK
Ruediger.Zillmer@unilever.com

Roderich Groß
Natural Robotics Lab
Sheffield Robotics &
Department of Automatic Control and Systems Engineering
The University of Sheffield, UK
r.gross@sheffield.ac.uk

ABSTRACT

Recently, there has been growing interest in the synchronization of mobile pulse-coupled oscillators. We build on the work by Prignano et al. (Phys. Rev. Lett. 110, 114101) and show that agents that interact exclusively with others in their cone of vision can exhibit different synchronization regimes. 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 addition, we show that, not only the speed of the agents, but also their angle and range of interaction can tune the appearance of this intermediate regime.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Distributed Artificial Intelligence—*Multiagent systems*

Keywords

Mobile Agents, Agent-Based Simulation, Emergent Behavior, Complex Systems, Swarm Robotics, Artificial Social Systems

1. INTRODUCTION

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 biologist from the beginning of the 20th century, and a vast number of research articles appeared during that time, describing and modeling this behavior [4]. 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 firings of other individuals, either the timing or frequency of its own flashes gets slightly shifted in an attempt to match this external stimuli [15]. In such a way, spontaneous order appears without a leader, in a process of self-organization.

The synchronization of populations of pulse-coupled oscillators 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 unison chirping of crickets, or even the synchronization of women menstrual cycles [25, 13].

The ubiquity of synchronization in communities of locally interacting constituents gave rise to intensive research in mathematics, and complex networks and non-linear dynamics in physics [1], aiming for an understanding and a description of these multiple systems led by the seminal work of Winfree [24]. Most of this research focuses on static networks of oscillators. Recently, however, there has been increasing interest in the study of synchronization in networks of mobile oscillators [19, 7, 20]. These studies aim to understand what role topological changes, due to the mobility of the oscillators, play on the synchronization of theses systems.

In [20] and [19], Prignano et al. investigate a system similar to that of fireflies. Each oscillator moves in the plane and influences only the neighbors that lie within a certain range of interaction [20] (**case A**); or only its nearest neighbor [19] (**case B**). In *case A*, a monotonically descending relationship between the time it takes for the system to synchronize and the speed of the oscillators can be observed. This is what one would expect, given that at high speeds all the oscillators interact with each other frequently, whereas at low speeds the neighborhood of any particular individual does not often change, leading to a rapid local synchrony but taking a long time to achieve coherence at a global scale. Surprisingly, in *case B*, a non-monotonic dependence is found. While the slow and fast regimes remain the same, an intermediate regime appears where neither of these mechanisms work and synchronization is inhibited.

The self-organizing behavior of pulse-couple oscillators is akin to that of swarm intelligent systems, where a number of relatively simple and limited individuals can achieve a complex collective behavior through local interactions without any centralized control [3, 17, 2]. Therefore, it is natural to extend the aforementioned work to a swarms of robots. As a matter of fact, Christensen et al. [6] developed a swarm level fault detection behavior based on firefly synchroniza-

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

tion, and implemented it on the *swarm-bot* robotic platform. In their work, similarly to *case A*, the agents interacted with others within a certain range. However, the effect of the speed of the robots and their radius of interaction was not investigated.

In this work we study a different kind of interaction, where the agents are influenced only by others in their cone of vision. This is a potentially more realistic scenario for robotics than *case B* 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. We show that by tuning the angle and range of vision, both the monotonic and the non-monotonic behaviors of *case A* and *case B*, respectively, can be obtained.

Other recent works that researched pulse-coupled synchronization in swarms of robots include [5], which studied a synchronization based approach to task allocation; and [21, 22, 23], which utilize evolutionary approaches to obtain robot controllers that would lead to swarm synchronization. Some of these approaches involve the use a line of sight interaction, similar to the cone of vision. However, all of them deal with movement synchronization in the environment. Therefore, the impact of the speed of the agents was not examined.

This paper is organized as follows. Section 2 introduces the methods used, including the oscillator and neighborhood models as well as the synchronization metric. Section 3 presents the experiments performed in a particle simulator and compares the results with those of [19, 20]. Section 4 shows the effects of embodiment and physical constraints in a robot simulator. Section 5 concludes the paper.

2. 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 the results of [20, 19], 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* [14].

2.1 Oscillator Model

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

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

where its phase, $\phi_i \in [0, 1]$, grows linearly in time with period τ, 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, ϕ_n, is updated multiplicatively by a factor ϵ [20, 19, 6], as follows (see Fig. 1 left):

$$\phi_i\left(t^-\right) = 1 \Rightarrow \begin{cases} \phi_i\left(t^+\right) = 0 \\ \phi_n\left(t^+\right) = (1+\epsilon)\,\phi_n\left(t^-\right). \end{cases} \tag{2}$$

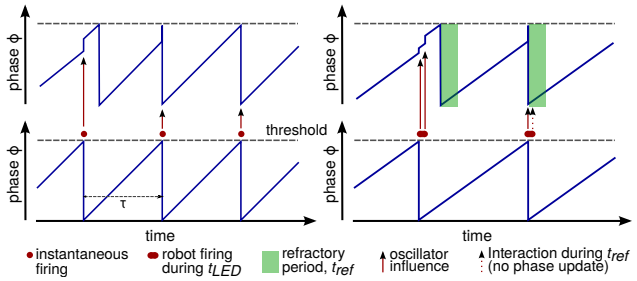

Figure 1: Dynamics of two oscillators, where the bottom one influences the top one. Left: Oscillator model and interaction as described in Sec. 2.1. Right: Oscillator as implemented in the robot simulations. During the refractory period (only shown for the upper oscillator) the oscillator is not influenced by any interaction. Outside the refractory period multiple consecutive interactions are allowed, as it will benefit synchronization.

Table 1: Dimensions of the Cone of Vision

Parameter	Value
R	$[0.05L, \infty]$
θ	$[10°, 360°]$

2.2 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 angle θ, and oriented in the direction of motion of A (see Fig. 2 top-left). If A is a neighbor of B, then B will influence A when applying Eq. 2.

The bottom-left panel in Fig. 2 shows the *interaction range* used in *case A* [20]. 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 *case A*.

In contrast to the *nearest neighbor interaction* (*case B*) described in [19] where each oscillator has precisely 1 neighbor, in our model any given agent could have from 0 to potentially $N-1$ neighbors (see Fig. 2 right for examples of different interactions). In this aspect, this is similar to the *interaction range* scenario of *case A*.

The dimensions of the cone of vision that were studied are shown in Table 1. Note that any R greater than $\sqrt{2}L$, the length of the diagonal, is equivalent to an infinite range.

2.3 Particle Simulation

The particle simulation was 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 (1) is integrated exactly.

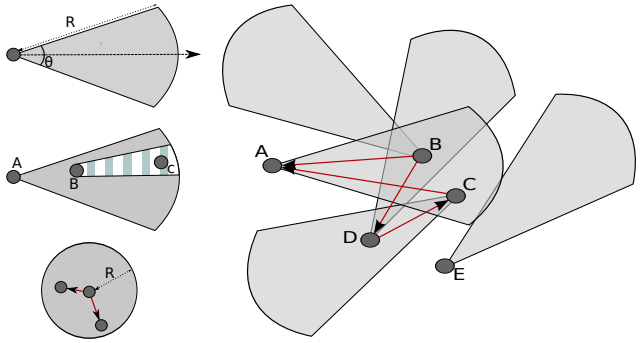

Figure 2: Top-left: Cone of interaction. Center-left: Occlusion in the robot simulation; C cannot be seen from A. Bottom-left: Interaction range used in [20]. The red arrows indicates the direction of the interaction. Right: Example of different interactions. A is neighbor of B and C; B has two neighbors, A and D; A and E have no neighbors and E is not a neighbor of any other agent.

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.

2.4 Robot Simulation

2.4.1 Robot and Simulator

The robot simulation was performed using the open-source Enki simulation toolkit [12]. 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). 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. [8, 9] 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 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 [14].

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.

2.4.2 Oscillator

The phase of the robot's oscillator is updated according to the discrete version of (1),

$$\phi_i(t_n) = \frac{\Delta t}{\tau}\phi_i(t_{n-1}), \tag{3}$$

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.

2.4.3 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 (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. 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 [6]. 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 [11, 18] (see Fig. 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. In practice this could be implemented by rotating the robots on the spot to observe the full environment. To account for this the LEDs should be kept on for longer and the refractory period increased accordingly.

2.5 Synchronization Metric

In order to measure the level of synchrony of the system, a certain oscillator, ϕ_1, is selected as reference, and upon its kth firing, at time T_k, we calculate the order parameter [20, 19], defined as follows,

$$\eta(T_k) = \frac{1}{N}\sum_{i=1}^{N}\cos\left(2\pi\phi_i(T_k)\right). \tag{4}$$

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 $\eta(T_k) = 0$ for a totally uniform phase distribution, to $\eta(T_k) = 1$ for complete synchronization.

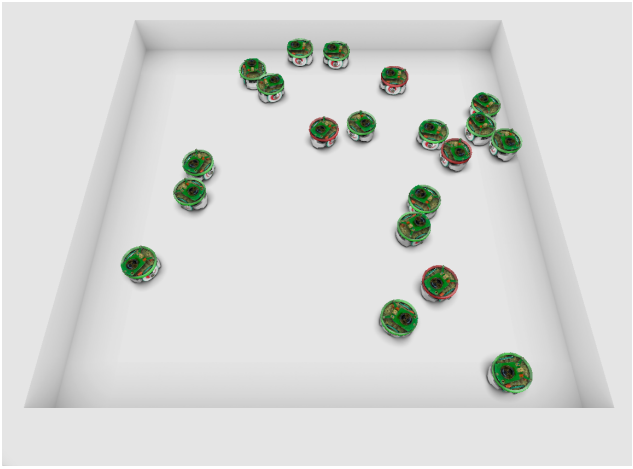

Figure 3: The robot environment in ENKI showing 20 e-pucks in a 100 cm square arena. The robots with the red LED ring turned on are firing.

Table 2: Parameters of the Particle Simulation

Parameter	Value
N	20
L	200 cm
τ	1 s
ϵ	0.1
V	$[10^{-2}, 10^{2}]$ cm/s
η_{sync}	$1 - 10^{-6}$
T_{cens}	10^{7}

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).

3. PARTICLE SIMULATION RESULTS

We performed the particle simulations with the parameters shown in Table 2. We studied the effect of varying the cone dimensions for 30 different speeds, V. We explored a set of 18 angles of view, θ, and 14 radii, R, from the ranges in Table 1. For each combination of V, θ, R the simulations were repeated 100 times, where the initial phases, positions and orientations of the oscillators were randomly chosen.

Figure 4 shows the average synchronization time, T_{sync}, as a function of the speed of the oscillators, for several angles, θ, and radii, R, of the cone of interaction. We obtained both the monotonic dependence of *case A* and the non-monotonic relationship of *case B*. For large angles and large cone radii $Tsync$ is a strictly decreasing function of V. However, as the size of the cone decreases (either θ 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 (*case B*), for the same setup is depicted with a dashed-line. The resemblance to our case becomes apparent in the second plot in Fig. 4 ($R = 0.25L$), where the intermediate regime coincides for the same range of speeds.

For further insight, Figure 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 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 $Tsync$ 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 *case A*, the synchronization time decreases monotonically with V and R as in [20].

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 [19], 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 7 shows the shift in speeds of T_{sync} from $L = 200$ cm to $L = 100$ cm.

Table 3: Parameters of the Robot Simulation

Parameter	Value
N	20
L	200 cm
τ	8 s
ϵ	0.1
V	$[10^{-2}, 12.8]$ cm/s
η_{sync}	0.95
T_{cens}	10^{4}
Δt	0.1 s
t_{LED}	0.4 s
t_{ref}	0.8 s

4. ROBOT SIMULATION RESULTS

We performed the robot simulations with the parameters shown in Table 3. For the initial experiments we used a scenario of the same size as in the particle simulation ($L = 200$ 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 $\tau = 8$ s in order to keep the number of 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. 4 to a system with 20 agents, $\eta_{sync} = 0.95$ implies that the firing time of any individual is not shifted further than 0.01τ from all other firings, which is equivalent

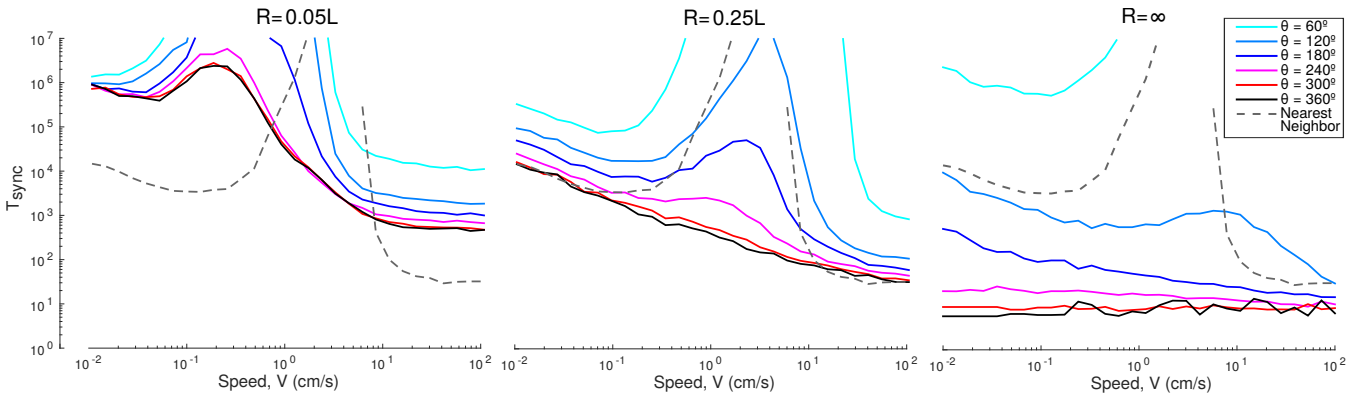

Figure 4: Log-log plot of T_{sync} as a function of the speed of the oscillators for the particle simulation. The results for three ranges of interaction, R, are shown. Different color lines represent different angles of interaction. The dashed line represents the nearest neighbor interaction of *case B* for the parameters of Table 2.

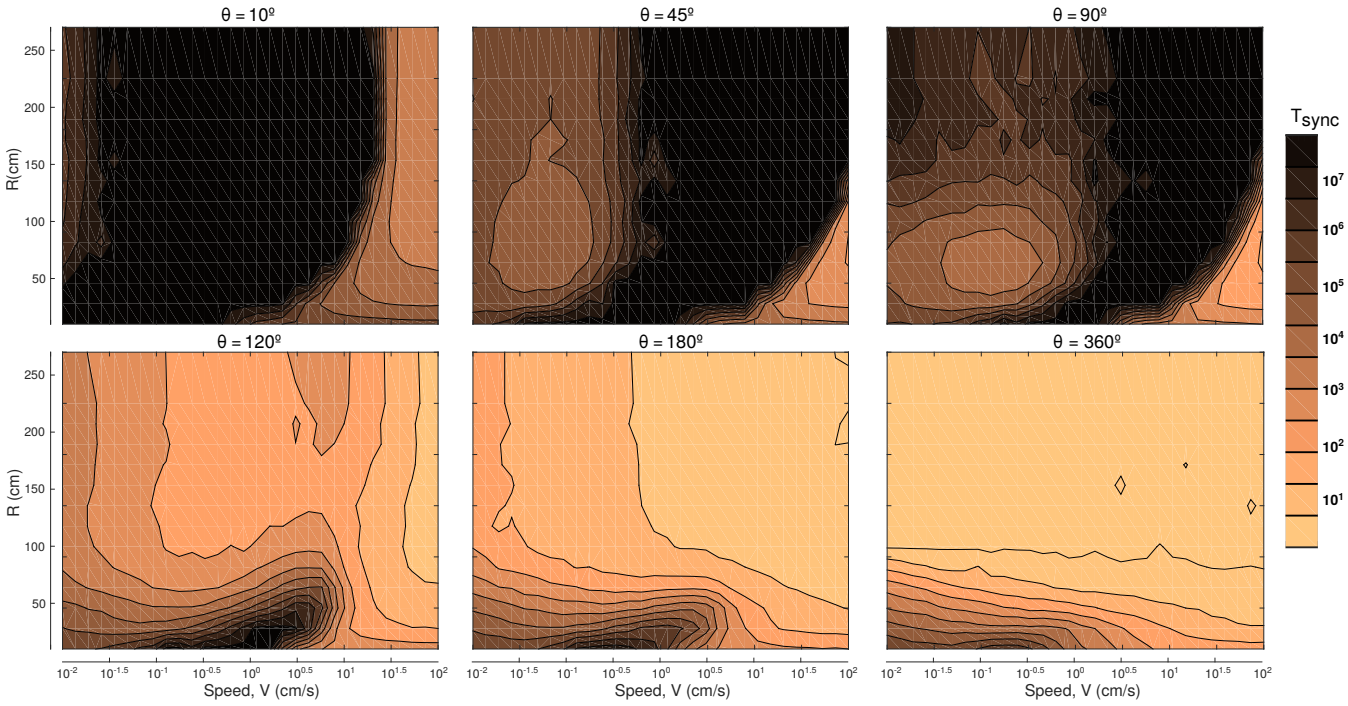

Figure 5: Contour plot showing the change in T_{sync} as a function of the speed of the oscillators, V, and their ranges of interaction, R, for the particle simulation. The results for six different angles of vision are shown. $\theta = 360$ is equivalent to *caseA*.

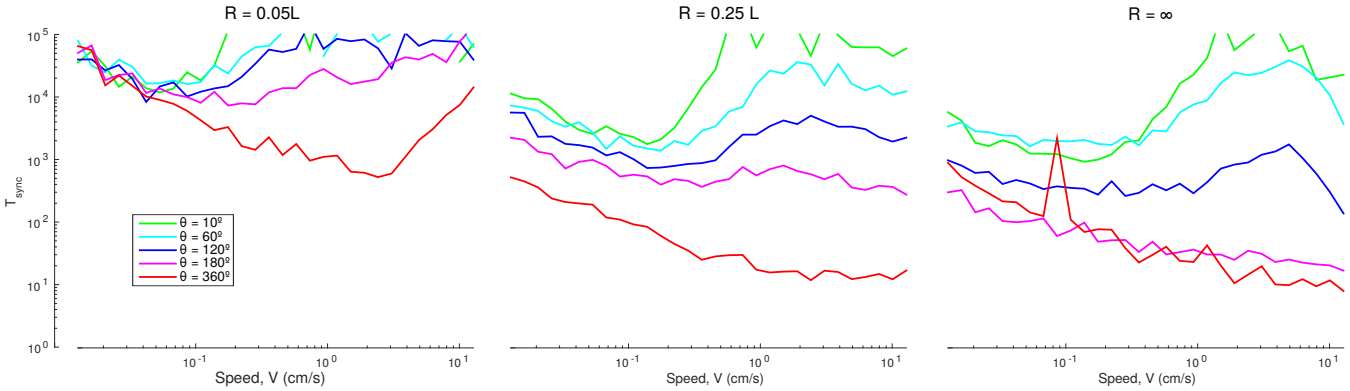

Figure 6: Log-log plot of T_{sync} as a function of the speed of the oscillators for the robot simulation. The results for three ranges of interaction, R, are shown. Different color lines represent different angles of interaction.

to the threshold suggested in [6]. $T_{cens} = 10^4$ with $\tau = 8\,\mathrm{s}$ 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, θ, and six interaction radii, R, from the ranges in Table 1. For every combination of V, θ 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.

The results of these experiments are shown in Figure 6. Despite the changes in implementation with respect to the particle simulation, the same essential behavior is found (cf. Fig. 4). The non-monotonic 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 $L = 100\,\mathrm{cm}$ for some of the studied angles (Fig. 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.

5. CONCLUSIONS

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

Prigano et al. [19] hypothesized that the occurrence of the intermediate regime in *case B* (that is, the case where only the agent that is nearest to a firing agent is influenced) 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 *case A* [20] (that is, the case 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, 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 [19] suggested a possible discontinuity in the intermediate regime of *case B*, where the synchronization is completely inhibited (i.e. $T_{sync} = \infty$). In our case, the synchronization time is computationally incalculable for small cones of vision. Nevertheless, we observe a gradual change of the behaviour 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.

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. In the future, we intend to validate the results of this paper using a swarm of physical e-puck robots.

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 [6] 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.

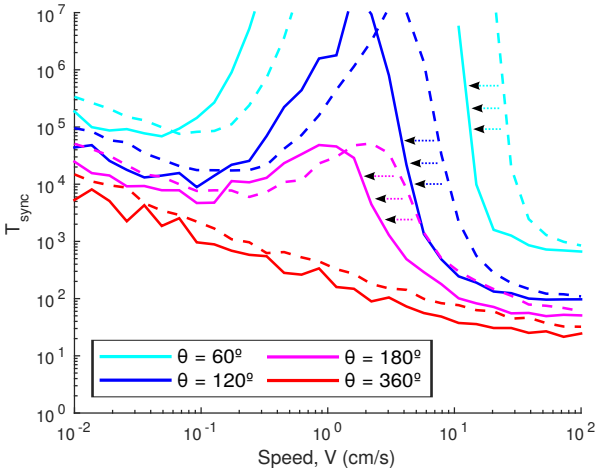

Figure 7: Effects of decreasing the environment size for $R = 0.25L$ in the particle simulation. The dashed lines represent the log-log plot of T_{sync} as a function of V for $L = 200\,\text{cm}$ whereas the solid lines depict the corresponding curves for $L = 100\,\text{cm}$.

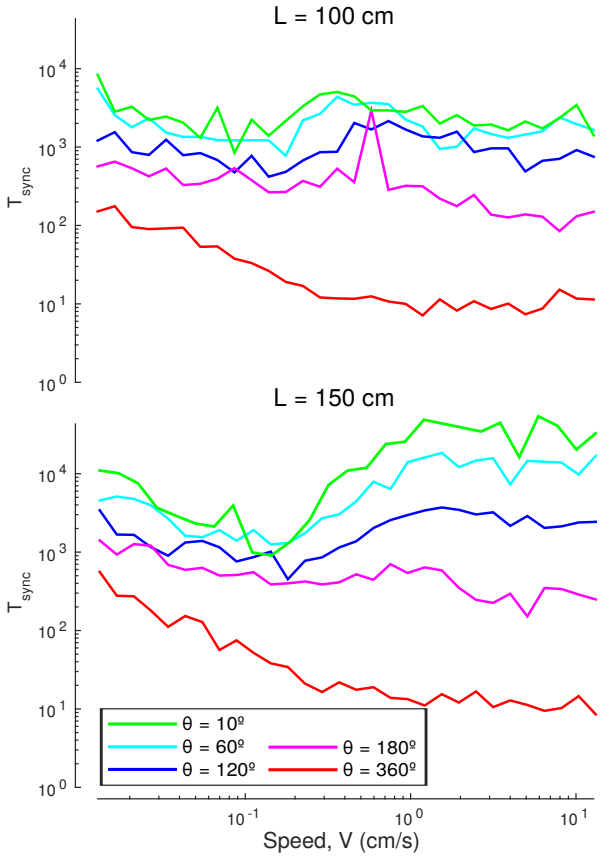

Figure 8: Effects of decreasing the environment size for $R = 0.25L$ in the robot simulation. The solid lines represent the log-log plot of T_{sync} as a function of V for $L = 100\,\text{cm}$ (top) and $L = 150\,\text{cm}$ (bottom).

REFERENCES

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

[2] G. Beni. From swarm intelligence to swarm robotics. *Swarm Robotics. Lecture Notes in Computer Science*, 3342:1-9, 2005

[3] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm intelligence: From natural to artificial systems*. Oxford University Press, 1999.

[4] J. Buck. Synchronous rhythmic flashing of fireflies. II. *Quarterly Review of Biology*, Vol. 63, No. 3 265–289, 1988.

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

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

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

[8] M. Gauci, J. Chen, W. Li, T. J. Dodd, and R. Groß. Clustering objects with robots that do not compute. *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*. International Foundation for Autonomous Agents and Multiagent Systems pages 421–428, 2014.

[9] M. Gauci, J. Chen, W. Li, T. J. Dodd, and R. Groß. Self-organized aggregation without computation. *The International Journal of Robotics Research*, 33(8):1145-1161, 2014

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

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

[12] S. Magnenat, M. Waibel, and A. Beyeler. Enki: The fast 2D robot simulator, 2007.

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

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

[15] J. D. Murray. *Mathematical Biology I: An Introduction, vol. 17 of Interdisciplinary Applied Mathematics*. Springer, New York, USA, 2002.

[16] M. Natrella. NIST/SEMATECH e-handbook of statistical methods. 2010.

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

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

[19] L. Prignano, O. Sagarra, and A. Díaz-Guilera. Tuning synchronization of integrate-and-fire oscillators through mobility. *Physical Review Letters*, 110(11):114101, 2013.

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

[21] V. Sperati, V. Trianni, and S. Nolfi. Self-organised path formation in a swarm of robots. *Swarm Intelligence*, 5(2):97–119, 2011.

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

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

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

[25] A. T. Winfree. *The geometry of biological time*, Interdisciplinary Applied Mathematics, vol. 12, Springer-Verlag New York, 2001.

APPENDIX

A. CENSORED DATA

As described in Sec. 2.5, 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 [10]. Assume that out of M trials, there are K where $T_{sync} < T_{cens}$, the procedure for calculating K-M estimator is as follows [16]:

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, α and γ. Lastly, we can calculate the mean of T_{sync} as

$$\alpha\Gamma\left(1 + \frac{1}{\gamma}\right),$$

and its variance as

$$\alpha^2\Gamma\left(1 + \frac{2}{\gamma}\right) - \left[\alpha\Gamma\left(1 + \frac{1}{\gamma}\right)\right]^2,$$

where Γ 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.