

Decentralized self-selection of swarm trajectories: from dynamical systems theory to robotic implementation

Guillaume Sartoretti, Max-Olivier Hongler, Marcelo Elias de Oliveira & Francesco Mondada

Swarm Intelligence

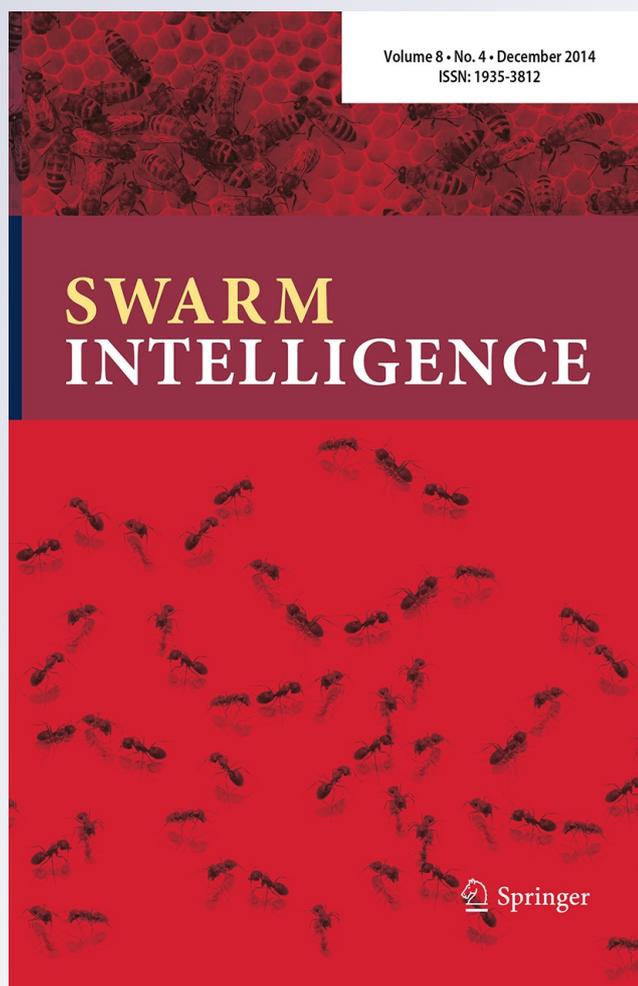
ISSN 1935-3812

Volume 8

Number 4

Swarm Intell (2014) 8:329-351

DOI 10.1007/s11721-014-0101-7



Your article is published under the Creative Commons Attribution license which allows users to read, copy, distribute and make derivative works, as long as the author of the original work is cited. You may self-archive this article on your own website, an institutional repository or funder's repository and make it publicly available immediately.

Decentralized self-selection of swarm trajectories: from dynamical systems theory to robotic implementation

Guillaume Sartoretti · Max-Olivier Hongler ·
Marcelo Elias de Oliveira · Francesco Mondada

Received: 20 December 2013 / Accepted: 15 October 2014 / Published online: 1 November 2014
© The Author(s) 2014. This article is published with open access at Springerlink.com

Abstract In this paper, we present a distributed control strategy, enabling agents to converge onto and travel along a consensually selected curve among a class of closed planar curves. Individual agents identify the number of neighbors within a finite circular sensing range and obtain information from their neighbors through local communication. The information is then processed to update the control parameters and force the swarm to converge onto and circulate along the aforementioned planar curve. The proposed mathematical framework is based on stochastic differential equations driven by white Gaussian noise (diffusion processes). Using this framework, there is maximum probability that the swarm dynamics will be driven toward the consensual closed planar curve. In the simplest configuration where a circular consensual curve is obtained, we are able to derive an analytical expression that relates the radius of the circular formation to the agent's interaction range. Such an intimate relation is also illustrated numerically for more general curves. The agent-based control strategy is then translated into a distributed Braitenberg-inspired one. The proposed robotic control strategy is then validated by numerical simulations and by implementation on an actual robotic swarm. It can be used in applications that involve large numbers of locally interacting agents, such as traffic control, deployment of communication networks in hostile environments, or environmental monitoring.

G. Sartoretti (✉) · M.-O. Hongler
EPFL/STI/IMT/LPM, Swiss Federal Institute of Technology in Lausanne,
1015 Lausanne, VD, Switzerland
e-mail: guillaume.sartoretti@epfl.ch

M.-O. Hongler
e-mail: max.hongler@epfl.ch

M. E. de Oliveira · F. Mondada
EPFL/STI/IMT/LSRO, Swiss Federal Institute of Technology in Lausanne,
1015 Lausanne, VD, Switzerland
e-mail: marcelo.eliasdeoliveira@epfl.ch

F. Mondada
e-mail: francesco.mondada@epfl.ch

Keywords Spatio-temporal pattern · Distributed swarm control · Brownian agents · Mixed canonical-dissipative dynamics · Mean-field approach · Braitenberg control mechanism · Robotics experimental validation

1 Introduction

How can one trigger the self-emergence of desired spatio-temporal patterns via a collection of interacting autonomous agents? Can we analytically unveil such collective behavior for scalable collections of agents? Is it possible to explicitly demonstrate how such collective behavior can possibly remain stable in the face of environmental noise? Progress in the engineering of swarm robotics relies on explicit and detailed answers to these questions, and that is the central goal of the present contribution. As no generic guidelines yet exist, engineering such collective behavior remains, in many respects, an open challenge. Bonabeau, Dorigo and Theraulaz illustrate how elementary, fully decentralized biological systems can achieve highly elaborate collective tasks [Bonabeau et al. \(1999\)](#). This pioneering work remains a rich and challenging source of inspiration for many further developments both in ethology [see the recent overview by [Leonard \(2013\)](#)] and in swarm robotics. Presently, serious effort is being made to import some basic features of these complex living systems into the engineering world of robotics. Obviously, animals remain intrinsically highly complex machines compared to actual robots and, therefore, direct applications remain rather limited. Nevertheless, recent attempts to combine artificial and natural collective systems by implementing the models observed in animal societies into that of robots shows promise ([Halloy et al. 2007](#); [Mondada et al. 2011](#)). Because of the complexity of the interactions taking place in such mixed societies, the models often remain partial and the link between them has not been fully established.

Several up-to-date reviews ([Kernbach 2013](#); [Brambilla et al. 2013](#)) summarize ongoing research in this area and the various engineering applications. Often the design of such robotic systems consists of a bottom-up trial-and-error exercise with no analytical link between the microscopic model (describing the single agent behavior) and the macroscopic one (describing the global result achieved). Microscopic and macroscopic models are then seen as separate approaches [Brambilla et al. \(2013\)](#), rather than steps in the engineering process. To bridge the gap between the micro- and macroscopic descriptions, [Martinoli et al. \(1999\)](#) constructed the first link between individual behavior and the statistical macroscopic model, which was successfully implemented on real robots. Along the same lines, [Lerman et al. \(2005\)](#) exhibited the emergence of collective behavior from individual robot controllers using a class of stochastic Markovian mathematical models. The authors validated their approach by performing experiments using real robots. More recently, [Hamann and Wörn \(2008\)](#) used an explicit representation of space together with methods of statistical physics to establish the link between microscopic and macroscopic models. Space heterogeneities are also considered by [Prorok et al. \(2011\)](#), who derived, from diffusion-type evolutions, a collective inspection scenario implemented on real robots. [Brambilla et al. \(2012\)](#) proposed a top-down design process built on iterative model construction and based on Probabilistic Computation Tree Logic (Deterministic Time Markov Chains). Their design methods were further validated on a group of physical *e-puck* robots; their iterative construction potentially enables other types of swarming behavior, depending on the skills of the acting programmer. [Berman et al. \(2009\)](#) likewise proposed a top-down design methodology, but at a higher level (i.e., at the sub-swarm level). Their decentralized strategy allows dynamical allocation of a swarm of robots to multiple predefined tasks. The approach by [Berman et al. \(2009\)](#) is based on population fractions,

which enable the design of stochastic control policies for each robot, and does not assume communication among robots. This strategy is analytically proven to achieve the desired allocation, and validation is made on a surveillance scenario involving 250 simulated robots distributed across four sites. More recently, [Berman et al. \(2011\)](#) have extended their work to spatially heterogeneous swarms. This control, however, can only generate static patterns of robot sub-swarms. In the present work, we aim to generate self-organized dynamic patterns (i.e. circulation of robots along closed planar curves).

Closely related to the present paper, [Hsieh et al. \(2007\)](#) synthesized controllers that allowed individual robots, when assembled in a swarm, to self-organize and circulate along a predefined closed curve. The system is fully distributed and relies only on local information, thus ensuring scalability. The controller can be analytically described, and does not require communication between the robots, thus simplifying its implementation. [Hsieh et al. \(2007\)](#) demonstrated convergence of the system for a certain class of curves and validated their theory with simulations. While the final behavior looks similar to what we are targeting here, we aim at directly deriving our controller from a stochastic dynamical model that can be analytically discussed. Our goal here is to allow the swarm to converge to a consensually selected curve among a given family (and not to a predefined curve as done in Hsieh et al.'s study). Conceptually, Hsieh et al.'s study belongs to the important family of gradient-based controls. Within this family, [Pimenta et al. \(2013\)](#) proposed a decentralized hydrodynamics-inspired controller that drives the swarm into a preassigned arrangement using a static and/or dynamic obstacle avoidance mechanism. They also tested their approach on swarms of actual robots, including *e-puck* robots. More closely related to our paper, [Sabattini et al. \(2012\)](#) developed a control strategy using decentralized time-dependent potential construction. Specifically, this mechanism allows a robotics swarm to track a closed curve (given by an implicit function, as in this paper), while keeping a minimal safety distance between the robots. In Sabattini et al.'s study [Sabattini et al. \(2012\)](#), the robots are given a predefined curve, whereas in our approach, the swarm consensually selects the closed curve that will be tracked.

Among non-gradient-based approaches, [Schwager et al. \(2011\)](#) demonstrated the stability and quantified the convergence rate of the control of a swarm of quad-rotors. Their approach is based on a combination of decentralized controllers and mesh networking protocols. Moreover, the authors only used local information and validated their control on a group of three real robots. Schwager et al.'s controller needs to know the relative position of its neighbors with high precision, while our goal is to be noise-resistant. [Sepulchre et al. \(2007\)](#) developed a methodology for the generation of stable circular motion patterns based on stabilizing feedbacks derived from Lyapunov functions. This theoretical work was then used to design the control of mobile sensor networks for optimal data collection in environmental monitoring [Leonard et al. \(2007\)](#). The global patterns generated in Sepulchre et al.'s study exhibit similarities with those presented in the present paper, but their control is “less” decentralized. Namely, Sepulchre et al. assume an all-to-all communication network between the robots, whereas in our work, we aim at using only local communication between neighbors within a given range. Given Sepulchre et al.'s assumption, the controller is also more sensitive to losses of communication between robots in the swarm. Finally, within non-gradient-based controls, it is worthwhile to mention [Tsiotras and Castro \(2011\)](#), who devised a decentralized mechanism, leading to *Spirograph*-like planar swarm trajectories. Their approach only relies on individually gathered information on distance and angle to neighbors.

With respect to the state of the art, whereas most research in control theory aims at finding strong theoretical guarantees, the present work follows a very different approach. We aim at bridging the gap between the purely mathematical considerations and the implementation on actual robots, and approach the problem from a mathematical angle. The mathematical models

are then implemented using a swarm of mobile robots. To some extent, our work is closer to that of biologists who use robotics as a validation tool for their theoretical models. Using this approach, we are able to achieve a decentralized consensual swarm control mechanism using only simple local information, such as the number of neighbors in a given range and their relative positions. In order to better link the model and the implementation, this article also includes a dictionary to help translation of the main concepts between the theoretical and the robotic control mechanisms. The developed controllers are not designed to achieve a fixed shape or trajectory, as often found in the state of the art, but to collectively choose one consensual shape. By tuning the control parameters, one can enable a given swarm to converge on different orbits without changing the behavior of each agent. Another important feature is the fact that the consensual orbit is not known by the agents, but is only determined by mutual interactions in a fixed range.

Although perfectly predictable analytically, this collective choice can be described as a *collective weakly emergent behavior*. Weak emergence is here understood in the S. and O. Kernbach's interpretation (Kernbach 2013; Kernbach et al. 2013) (i.e., macroscopic swarm behavior due to interactions between agents but having its origin in the model that describes the system's behavior). Our model has a tractable complexity, enabling us to explicitly describe the resulting emergent collective behavior, which is controllable. Specifically, our class of models presents a distributed mechanism, enabling agents to select a parameter while fixing one consensual traveling orbit among the given family in a distributed manner, and this feature holds even if noise corrupts their movements.

The paper is organized as follows: Sect. 2 exposes the basic mathematical formulation of agents' dynamics. Section 3 reports simulation experiments that fully match our theoretical finding even for finite N agents' populations (i.e., when the mean-field limit is not reached). Section 4 describes, in detail, how the actual implementation of our mathematical model was realized with a collection of e-puck autonomous robots.

2 Mathematical modeling

Our mathematical model consists of a set of N coupled stochastic differential equations (SDE) where the driving stochastic processes are White Gaussian Noise (WGN) representing a swarm of homogeneous Brownian agents Schweitzer (2003). The use of WGN in the dynamics implies that the resulting stochastic processes are Markovian. Accordingly, all probabilistic features of the model are known by solving the associated Fokker-Planck (FP) equation. A similar approach has already been used in several studies (Berman et al. 2011; Hamann and Wörn 2008; Prorok et al. 2011). Agents' interactions are ruled by specifying the radius of an observation disk (i.e., metric interactions) and the strength of the attraction/repulsion control parameter. The specific form of agents' individual dynamics and their mutual interactions are constructed from a Hamiltonian function, from which one derives both a canonical and a dissipative drift force (i.e., *mixed canonical-dissipative vector fields* Hongler and Rytter (1978)). By imposing the orthogonality between the canonical and dissipative vector fields, we are able, in the $N \rightarrow \infty$ mean-field limit, to explicitly calculate the stationary probability (i.e., invariant probability measure) characterizing the global agents' populations. For our class of models, we can analytically observe how a decentralized algorithm is able to generate a milling-type spatio-temporal pattern. In this pattern, all agents will circulate with constant angular velocity in the vicinity of a self-selected level curve from the input Hamiltonian function.

2.1 Single agent dynamics

Let us first consider the single agent Mixed Canonical-Dissipative (MCD) stochastic planar dynamics (a tutorial devoted to MCD can be found in Appendix section “MCD dynamics”):

$$\begin{pmatrix} dX(t) \\ dY(t) \end{pmatrix} = \mathbb{V} \begin{pmatrix} \partial_X H(X, Y) \\ \partial_Y H(X, Y) \end{pmatrix} dt + \sigma \begin{pmatrix} dW_X(t) \\ dW_Y(t) \end{pmatrix}, \quad \begin{pmatrix} X(0) = X_0 \\ Y(0) = Y_0 \end{pmatrix} \tag{1}$$

where \mathbb{V} stands for an antisymmetric (2×2) dynamical matrix defined by:

$$\mathbb{V} = \begin{pmatrix} U'[H(x, y)] & +\omega \\ -\omega & U'[H(x, y)] \end{pmatrix} \tag{2}$$

where $(X, Y) \in \mathbb{R}^2$ are the positions of the planar agent, $H(X, Y) : \mathbb{R}^2 \mapsto \mathbb{R}^+$ and $U[H] : \mathbb{R}^+ \mapsto \mathbb{R}$ ($U'[H]$ stands for the derivative of $U[H]$ with respect to H) are differentiable functions, $\sigma > 0$ and $\omega \in \mathbb{R}$ are constant parameters. The inhomogeneous terms $dW_X(t)$ and $dW_Y(t)$ are two independent WGN sources. We shall assume that $H(X, Y) = E, \forall E > 0$ defines closed concentric curves and $H(0, 0) = 0$. In addition, we require that $U[0] > 0$ and $\lim_{E \rightarrow \infty} U[H(E)] = -\infty$. When $\sigma = 0$, the (deterministic) dynamics Eq. (1) admits closed trajectories \mathcal{C} 's, which are defined by the solutions of the equation $U[H(X, Y)] = 0$. Depending on the sign of the curvature (i.e., second derivative) U'' taken at \mathcal{C} , one can either have attracting ($U'' < 0$) or repulsive ($U'' > 0$) limit orbits. An attracting \mathcal{C} drives any initial conditions (X_0, Y_0) to \mathcal{C} in asymptotic time. In Eq. (1), the antisymmetric part of \mathbb{V} produces a rotation in \mathbb{R}^2 with angular velocity ω . This defines a canonical (conservative) part of the vector field (i.e., the Hamiltonian part of the dynamics). Concurrently, the diagonal elements of \mathbb{V} model the non-conservative components of the dynamics (i.e., dissipation or supply of energy). From its structure, the non-conservative (deterministic) vector field of Eq. (1) is systematically perpendicular to the conservative one. This geometrical constraint reduces the dimensionality of the dynamics and allows the derivation of explicit and fully analytical results.

In the presence of WGN ($\sigma > 0$ in Eq. (1)), the resulting SDE describe a Markovian diffusion process, fully characterized by a bivariate transition probability density (TPD) measure

$$P(x, y, t | x_0, y_0, 0) dx dy = \text{Prob} \{x \leq X(t) \leq x + dx, y \leq Y(t) \leq y + dy | X(0) = x_0, Y(0) = y_0\}$$

The TPD $P(x, y, t | x_0, y_0, 0)$ is the solution of a FP equation:

$$\begin{cases} \partial_t P(x, y, t | x_0, y_0, 0) = \mathcal{F} \{P(x, y, t | x_0, y_0, 0)\}, \\ P(x, y, 0 | x_0, y_0, 0) = \delta(x - x_0)\delta(y - y_0) \end{cases} \tag{3}$$

where $\delta(\cdot)$ stands here for the Dirac probability mass and the FP operator reads:

$$\mathcal{F} \{ \cdot \} := -\partial_y [f_x(x, y) \{ \cdot \}] - \partial_x [f_y(x, y) \{ \cdot \}] + \frac{1}{2} \sigma^2 [\partial_{xx} + \partial_{yy}] \{ \cdot \} \tag{4}$$

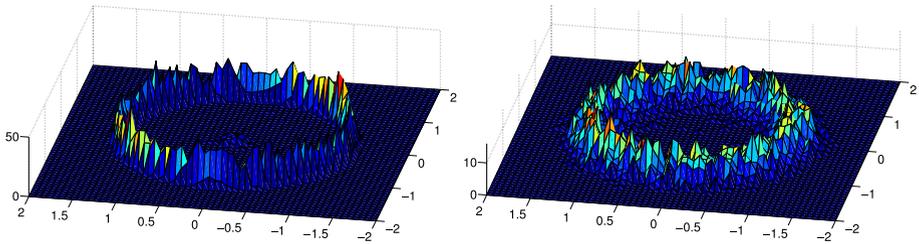


Fig. 1 “Ring”-shaped probability density on the plane, for $\sigma = 0.07$ and $\sigma = 0.35$. These simulations were run with $N = 5000$ agents, up to time $T = 10[s]$, with $\gamma = 1$

with

$$\begin{cases} f_x(x, y) := +\partial_y [H(x, y)] + U[H(x, y)]\partial_x [H(x, y)] \\ f_y(x, y) := -\partial_x [H(x, y)] + U[H(x, y)]\partial_y [H(x, y)] \end{cases}$$

For asymptotic time, the diffusion processes reach a stationary regime: $\lim_{t \rightarrow \infty} P(x, y, t \mid x_0, y_0, 0) = P_s(x, y)$, solving the stationary FP Eq. (3), for which we obtain:

$$P_s(x, y) = \mathcal{N} e^{(2/\sigma^2) U[H(x,y)]} \tag{5}$$

and \mathcal{N} is the probability normalization factor yielding $\int_{\mathbb{R}^2} P_s(x, y) dx dy = 1$.

Example Consider the Hamiltonian function $H(x, y) = x^2 + y^2$, $U[H] = \gamma \{ \mathcal{L}^2 H - \frac{1}{2} H^2 \}$ and $U'[H] = \gamma \{ \mathcal{L}^2 - H \}$. In this case, the deterministic dynamics coincides with the Hopf oscillator with a unique attracting \mathcal{C} that is a circle with radius \mathcal{L} . In this situation, the measure Eq. (5) exhibits the shape of a circular “ring” with its maximum located on the circle \mathcal{C} (see Fig. 1).

2.2 Multi-agent dynamics

Let us now build on the dynamics presented in Sect. 2.1 and consider a homogeneous swarm of N interacting agents exhibiting individual dynamics similar to the one described in Eq. (1). Specifically, for $k = 1, 2, \dots, N$, we now write:

$$\begin{pmatrix} dX_k(t) \\ dY_k(t) \end{pmatrix} = \mathbb{V}_{k,\rho,M}(t) \begin{pmatrix} \partial_{X_k} H(X_k, Y_k) \\ \partial_{Y_k} H(X_k, Y_k) \end{pmatrix} dt + \sigma \begin{pmatrix} dW_{X_k}(t) \\ dW_{Y_k}(t) \end{pmatrix}, \quad \begin{pmatrix} X_k(0) = X_{0,k} \\ Y_k(0) = Y_{0,k} \end{pmatrix} \tag{6}$$

where $\mathbb{V}_{k,\rho,M}$ stands for an antisymmetric (2×2) dynamical matrix defined by:

$$\mathbb{V}_{k,\rho,M} = \begin{pmatrix} \left[\frac{N_{k,\rho}(t)}{N} - \frac{1}{M} \right] + \gamma \left[\mathcal{L}_{k,\rho}(t) - \frac{1}{2} H(X_k, Y_k) \right] & + \frac{N_{k,\rho}(t)}{N} \\ - \frac{N_{k,\rho}(t)}{N} & \left[\frac{N_{k,\rho}(t)}{N} - \frac{1}{M} \right] + \gamma \left[\mathcal{L}_{k,\rho}(t) - \frac{1}{2} H(X_k, Y_k) \right] \end{pmatrix} \cdot M \cdot \omega, \tag{7}$$

where $M > 1$ and $\omega > 0$ are (time-independent) control parameters, $M \cdot \omega$ is a normalization factor and $dW_{(\cdot)}(t)$ are all independent standard WGN sources (homogeneity follows as we

have $H_k(X_k, Y_k) \equiv H(X_k, Y_k) \forall k$. The mutual interactions enter via the term $N_{k,\rho}(t)$ present in the dynamic matrix $\mathbb{V}_{k,\rho,M}(t)$. To define $N_{k,\rho}(t)$, we assume that each agent \mathcal{A}_k ($k = 1, 2, \dots, N$) is located at the center of an (individual) observation disk $\mathcal{D}_{k,\rho}(t)$ with common constant radius ρ . We further assume that \mathcal{A}_k is, in real time, able to count the number of neighboring fellows $N_{k,\rho}(t)$ that are contained in $\mathcal{D}_{k,\rho}(t)$. Finally, we introduce the instantaneous (geometric) inertial moment of the agents located in $\mathcal{D}_{k,\rho}(t)$, namely:

$$\mathcal{L}_{k,\rho}(t) := \frac{1}{N_{k,\rho}(t)} \sum_{j \in \mathcal{D}_{k,\rho}(t)} H(X_j(t), Y_j(t)) \geq 0. \tag{8}$$

To heuristically understand the collective behavior emerging from the set of SDE's given in Eq. (6), let us comment on the following features:

- (a) Large population of indiscernible agents. As we consider homogeneous populations, we may randomly tag one agent, say \mathcal{T} , and consider the actions of the others as an effective external field (often referred to as the mean-field point approach). This procedure reduces the nominal multi-agent dynamics to a single effective agent system, thus making the analytical discussion much easier while still taking into account interactions of the neighbors. The individual \mathcal{T} -diffusive dynamics on \mathbb{R}^2 are then assumed to be representative of the global swarm and follow from dropping the index k in Eqs. (6) and (7). The effective interactions terms $N_{\rho,M}$ and $\mathcal{L}_{\rho,M}$ are to be determined by auto-consistency from the TPD measure of \mathcal{T} . In the stationary regime, thanks to the orthogonal property between the canonical and the dissipative components of the vector fields, we obtain:

$$P_s(\mathbf{x}, \mathbf{y}) = [P_s(x, y)]^N = \left[\mathcal{N} e^{\frac{2\gamma}{\sigma^2} U[H(x,y)]} \right]^N \quad \mathbf{x} := (x_1, x_2, \dots, x_N) \tag{9}$$

For $H(X, Y) = \frac{X^2+Y^2}{2}$, the harmonic oscillator Hamiltonian, and for large signal-to-noise ratio $\mathcal{S} = 2\gamma/\sigma^2 \ll 1$, we can analytically express the value of the limit cycle's orbit:

$$\mathcal{L}_{\rho,M} = \frac{\rho}{\sqrt{2 - 2 \cos(\pi/M)}}. \tag{10}$$

- (b) Roles of the control parameters M and ρ . The origin $\mathbf{O} := (0, 0)$ is a singular point of the deterministic dynamics and its stability follows from linearizing the dynamics near \mathbf{O} . In the stationary regime, the linearization reads:

$$\mathbb{V}_{\rho,M}(\mathbf{0}) = \begin{pmatrix} \left[\frac{N_{\rho,M}}{N} - \frac{1}{M} \right] + \gamma \mathcal{L}_{\rho,M} & + \frac{N_{\rho,M}}{N} \\ - \frac{N_{\rho,M}}{N} & \left[\frac{N_{\rho,M}}{N} - \frac{1}{M} \right] + \gamma \mathcal{L}_{\rho,M} \end{pmatrix}$$

with eigenvalues:

$$\lambda_{\pm} = \underbrace{\left[N_{\rho,M}/N - 1/M \right]}_{\mathcal{R}\mathcal{E}} \pm \sqrt{-1} \left[\sqrt{N_{\rho,M}/N} \right]. \tag{11}$$

When $\mathcal{R}\mathcal{E} < 0$, (resp. > 0), \mathbf{O} is attractive (resp. repulsive). The very existence of a stationary regime implies that $P_s(x, y)$ vanishes for large values of $|x|$ and $|y|$. For agents far from \mathbf{O} , both $N_{k,\rho,M}$ and $\mathcal{L}_{k,\rho,M}$ will vanish (with high probability, except the agent at the center, no agents will be found inside $\mathcal{D}_{k,\rho}$), implying \mathbf{O} to be attractive as $\mathcal{R}\mathcal{E} < 0$; conversely, agents close to \mathbf{O} will be repelled. Ultimately, an equilibrium

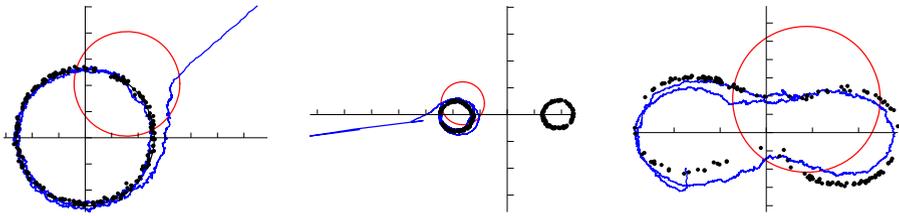


Fig. 2 Numerical simulation for $N = 200$ agents, $\rho = 1$, $\sigma = 0.07$ and $M = 4$. The black dots show the final position of the agent at time $T = 50[s]$, while the blue line gives the trajectory of one arbitrary agent (with its observation range in red). The black circle (hidden below the swarm of agents) gives the theoretically computed limit cycle. This shows the exactitude of Eq. (10), even for finite swarms. *Left* Harmonic oscillator Hamiltonian function $H(X, Y) = \frac{X^2+Y^2}{2}$. *Center* Cassini ovals Hamiltonian function $H(X, Y) = ((X - 1)^2 + Y^2) \cdot ((X + 1)^2 + Y^2)$, with $\rho = 0.4$, showing closed ovals without connections (see Yates (1947)). *Right* Same Hamiltonian function as the center figure, with $\rho = 0.8$, presenting one connected orbit around both attractors

consensus will be reached when the diagonal part of $\nabla_{\rho, M}$ vanishes which is achieved when Eq. (10) is attained.

- (c) Role of the dissipative terms $U'[H, t]$ and the parameter $\mathcal{L}_{\rho, M}$. Observe that the regulating term $U'[H, t]$ is itself not necessary to reach a consensual behavior. However, for large γ 's, it enhances the convergence rate toward equilibrium and reduces the variance around the mode of $P_s(x, y)$. This enables us to easily estimate the radius $\mathcal{L}_{\rho, M}$ given by Eq. (10).

According to Eqs. (9) and (10), we observe that, in the stationary regime, the swarm self-selects a consensual circular orbit with radius $\mathcal{L}_{\rho, M}$ that depends on both the observation range ρ and the effective attracting range M .

Numerical simulation results, obtained by applying the Euler method over discrete timesteps with $dt = 5 \cdot 10^{-3}[s]$, are shown in Figs. 2 and 3. The employed discretization works perfectly, as the numerical results coincide very closely with our theoretical ones (especially in the case of the theoretical limit cycle in Fig. 2, left).

Note that a Hamiltonian function can be constructed from a set of points along any desired closed curve on the plane (without self-intersection). A simple fit can be made on this set of points (e.g., least squares method), returning a high-degree polynomial form representing the Hamiltonian function. This function will be differentiable (by construction), and we can ensure $H(X, Y) > 0(\forall X, Y)$ by adding a constant; the Hamiltonian can then be directly used in our model.

3 Robotic implementation

3.1 Braitenberg control mechanism

The actual adaptation of our MCD dynamics is implemented on the model of a *Braitenberg control mechanism* (BCM) Braitenberg (1984). In the BCM, the speed of the robot's motor(s) is reactuated at discrete timesteps, solely based on the output value(s) of a series of sensors.

In our case, we use cylindrical robots, equipped with eight light sensors (s_0, s_1, \dots, s_7) evenly distributed along their circular perimeter and two driving motors (left and right), as pictured in Fig. 4. The robot's movement evolves on the plane, with a light source located at the origin.

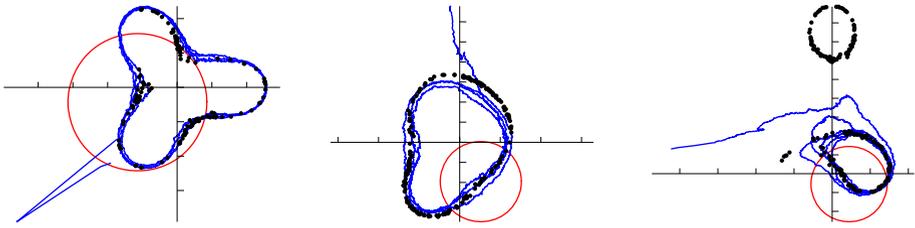


Fig. 3 Numerical simulation for $N = 200$ agents, $\rho = 1$, $\sigma = 0.07$ and $M = 4$, until time $T = 25[s]$ (left) or $T = 50[s]$ (center and right). *Left* Hamiltonian function with three attracting poles $H(X, Y) = ((X - 1)^2 + Y^2) \cdot ((X + \frac{1}{2})^2 + (Y - \frac{\sqrt{3}}{2})^2) \cdot ((X + \frac{1}{2})^2 + (Y + \frac{\sqrt{3}}{2})^2)$. *Center* Asymmetric Hamiltonian function $H(X, Y) = (((X - \frac{1}{2})^2 + Y^2) \cdot ((X + \frac{1}{2})^2 + (Y - \frac{\sqrt{3}}{2})^2) \cdot ((X + \frac{3}{4})^2 + (Y + \frac{\sqrt{27}}{4})^2))^{\frac{1}{3}}$. *Right* Hamiltonian function $H(X, Y) = (((X - 1)^2 + Y^2) \cdot ((X + 1)^2 + Y^2) \cdot (X^2 + (Y - 4)^2))^{\frac{1}{3}}$, resulting in two disconnected orbits

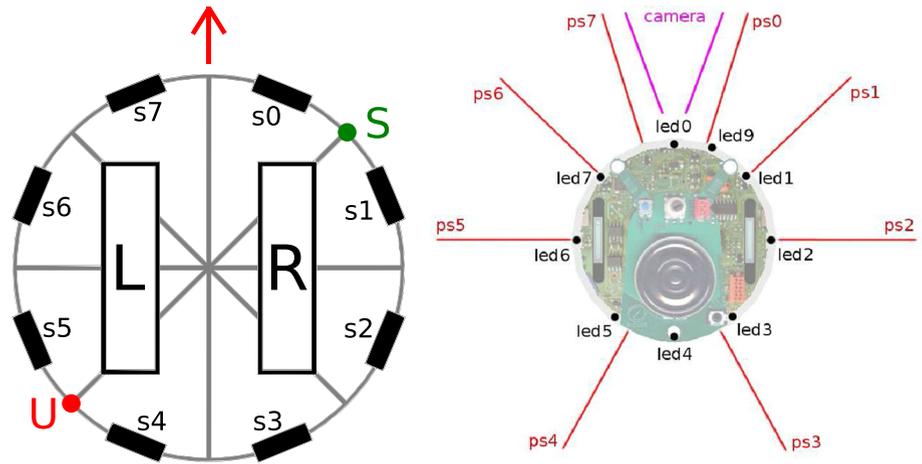


Fig. 4 *Left* Theoretical robot, seen from above. Note the eight sensors evenly distributed on the perimeter of the robot's body and the two motors L and R . The point S (resp. U) shows the stable (resp. unstable) facing direction of the light source in the BCM. *Right* Actual e-puck robot in the same position; the eight proximity sensors (ps0,ps1,...,ps7) are nearly evenly distributed on the perimeter

The light source serves as the origin O in the mathematical model, acting as an attractor for the robots, while also being the center of the limit cycle the robots will ultimately arrange and circulate on. Our goal is to let a swarm of robots arrange and circulate in a circle around the light source, following a similar control mechanism to Eq. (1), with the Hamiltonian function of the Hopf oscillator. This choice of circular trajectories allows us to compare the behavior of an agent swarm (with analytical results) and a robotic swarm.

The light sensors gather the following information:

1. $N_\rho(\mathbf{t})$: the number of instantaneous neighbors in a fixed circular range around each robot ($N_\rho(t) \geq 1 \forall t$ as every robot always detects itself),
2. $\{S_0(\mathbf{t}), S_1(\mathbf{t}), \dots, S_7(\mathbf{t})\}$: the normalized light intensity ($\in [0; 1]$, 1 for the sensor(s) receiving the most light) measured on each sensor.

The gathered information enables us to update the velocities at discrete timesteps according to the rule:

$$\begin{aligned}
 V_R(t + 1) &= \alpha \cdot N_\rho(t) + \beta \cdot \left(\overbrace{S_0(t) + \sum_{i=5}^7 S_i(t)}{=:S_L(t)} - \overbrace{\sum_{i=1}^4 S_i(t)}{=:S_R(t)} \right) \\
 V_L(t + 1) &= \alpha \cdot N_\rho(t) + \beta \cdot \left(\begin{matrix} S_R(t) & - & S_L(t) \end{matrix} \right),
 \end{aligned}
 \tag{12}$$

where α and β are two positive parameters, controlling the translation/rotation velocities. This pair of controls in Eq. (12) effectively adapts Eq. (1) (in Cartesian coordinates) to a robotics paradigm, in terms of left/right motor speeds (speed and heading). In Eq. (12), we reduce the number of sensors to two by grouping them into the left group (S_L) and the right group (S_R).

The first term $\alpha \cdot N_\rho(t)$ accounts for the forward speed of the robot, and the other term regulates the heading. Indeed, when $\alpha = 0$, the speeds have the same norm but opposite signs, implying that the robot can only turn around its main axis. In this case, the robot stabilizes with the light source facing S , i.e., with the light source at the same distance from sensors 0 and 1, with $S_L(t) \equiv S_R(t) \Rightarrow V_L(t) = V_R(t) = 0$. This ending position is the only stable steady state; when the light source is facing the point U , at the same distance from sensors 4 and 5, it is unstable.

When the light initially stands on the left of the robot (i.e., facing sensors 5, 6, 7, or 0), $V_L(t)$ will exceed $V_R(t)$ (as $S_L(t) > S_R(t)$). In this configuration, the robot will turn on its axis until the light faces the point S . The same argument holds when the light initially stands on the right of the robot.

When the first term in Eq. (12) (the forward speed) is **constant and positive**, this control mechanism drives the robot to the light source to turn around it. The robot constantly moves forward, keeping the light source facing S , thereby creating a rotating trajectory around the light source.

The radius of the resulting orbit will be directly proportional to α , and inversely proportional to β : an increase in α results in a larger distance covered before the next heading update (and hence, a larger orbit). Conversely, an increase in β induces a sharper heading adaptation (and hence, a smaller orbit). For a fixed β , the radius of the limit cycle depends only on α .

In our case, we fix α and β , thus letting the forward speed be proportional to the number of instantaneous neighbors. This forces the orbit to shrink when few neighbors are detected, and increase in the other configuration, mimicking the MCD dynamics. While in the mathematical model of Sect. 2, the parameters ρ and M selected the limit cycle’s radius, here ρ , α , and β contribute to the orbit selection. Accordingly, the parameter M will be adjusted by an ad-hoc combination of α and β .

Assuming the swarm of robots is large enough to ensure local communications, the proposed BCM will let the swarm converge to a consensual orbit and circulate on it at a constant speed. This convergence does not depend on the initial position and heading of the robots. As the central light is a global attractor, robots will also never drive away from it.

3.2 Physical implementation on e-puck robots

The formerly exposed BCM has been implemented on the open-hardware e-puck robot specifically designed for experimental swarm robotics [Mondada et al. \(2006\)](#). The e-puck is a cylindrical robot, of approximately 70 millimeters in diameter, with 8 infrared (IR)

proximity sensors (emitter and receptor) located nearly evenly on its perimeter (see Fig. 4). Each sensor is composed of an IR emitter and a receptor, and can be used passively (only the receptor, to sample ambient light), or actively (emitter/receptor, for proximity search or message exchanges).

Our implementation uses only IR sensors (passively and actively, alternately), and the two motors. Each timestep is separated into five phases:

1. Message broadcasting: Each robot first broadcasts a message to all neighboring robots via IR (20–40 cm of range, depending on the angles of the emitter/receptor sensors). The message contains its ID, and the number of agents that have been counted at the previous timestep $N_\rho(t - 1)$.
2. Message reading: The received messages are read to measure $N_\rho(t)$, the number of detected neighbors. A filtering step also allows detection of robots that were in range in the previous steps, but whose messages were not received (to correct possible miscommunication between robots). A memory of the detection of the other robots in the last 5 steps is maintained, and a *majority window algorithm (MWA)* is applied to correct for robots not being correctly detected. The filtered value $N'_\rho(t)$ is thereby constructed. The MWA counts as a neighbor any robot that has been in range at least 3 times in the last 5 timesteps, or that has just been detected during the present timestep. The number of neighbors can only be underestimated, as we use an unmodulated light source (on a continuous current), and the IR communication between the e-pucks are modulated. This means that no false positives can occur in the communication among the robots, and only false negatives (undetected neighbors) can happen. To try and reduce the influence of non-detection of neighbors, we artificially force the number of neighbors at time t to be bounded below by $(N_\rho(t - 1) - 1)$. This mechanism ensures a smooth behavior of the speeds of the motors, by avoiding large jumps in the number of neighbors over time. We define $V_\rho(t)$ to be the set of the values $N_\rho(t - 1)$ received from the neighbors at time t . Then the final value $N_\rho(t)$ is defined as:

$$N_\rho(t) := \max \left(N_\rho(t - 1) - 1, \frac{N'_\rho(t) + \sum_i V_\rho(t)_i}{N'_\rho(t) + 1} \right). \tag{13}$$

3. Light source sampling: The intensity of the light source is passively sampled on each of the eight IR sensors to obtain $\{\tilde{S}_0(t), \tilde{S}_1(t), \dots, \tilde{S}_7(t)\}$. The normalized values are then computed as $S_i(t) = \tilde{S}_i(t) / \max_i \{\tilde{S}_i(t)\}$ for $i = 0, 1, \dots, 7$.
4. Velocity re-actuation: With the data gathered, the velocity of each motor is computed and updated following Eq. (12), with $\alpha = 100$ and $\beta = 20$, and a base speed of 100:

$$\begin{cases} V_R(t + 1) = 100 + 100 \cdot N_\rho(t) + 20 \cdot (S_L(t) - S_R(t)) \\ V_L(t + 1) = 100 + 100 \cdot N_\rho(t) + 20 \cdot (S_R(t) - S_L(t)). \end{cases} \tag{14}$$

The speeds here are given in *steps/second*. 1000 steps correspond to one full rotation of a wheel, whose diameter is approximately 41mm. A speed of 500 steps/s corresponds to $\frac{500}{1000} \cdot \pi \cdot 41 \approx 64.4$ [mm/s].

5. Avoidance mechanism: A proximity search is performed on each sensor to detect obstacles; avoidance is implemented by adapting the motors' velocities. The avoidance mechanism used is a standard Braitenberg-inspired avoidance mechanism. It has been designed to avoid collisions between e-puck robots, but cannot guarantee collision avoidance with the light source. Its pseudo-code is given in Algorithm 1.

```

Data:  $\alpha = 100$ ;

// Something ahead, stop for this timestep
if obstacle_front() then
    VR(t+1) = 0;
    VL(t+1) = 0;
end
// Something ahead slightly left/right, turn the other direction
if obstacle_front_right() then
    VL(t+1) = VL(t+1) -  $\alpha$ ;
end
if obstacle_front_left() then
    VR(t+1) = VR(t+1) -  $\alpha$ ;
end
// Something left/right, turn slightly the other direction
if obstacle_right() then
    VL(t+1) = VL(t+1) -  $\alpha/2$ ;
end
if obstacle_left() then
    VR(t+1) = VR(t+1) -  $\alpha/2$ ;
end

right_motor(VR(t+1));
left_motor(VL(t+1));

```

Algorithm 1: Avoidance mechanism

Figure 5 summarizes the robot's main loop.

Different noise sources are inherent to this robotic implementation. First, the motors' noise: each robot can have a slightly different speed from its programming (the e-puck motors sometimes miss some motor steps, thus changing their forward speed and heading). Second, the sensors' noise in sampling the light source can also slightly change the robot's heading in the BCM. Finally, we must account for the noise in the number of neighbors, as they change its forward speed in the BCM. This final source is the biggest source of noise in the BCM. All of these noise sources add up, letting the robots follow noisy trajectories, as expected from the mathematical model of Eq. (1). We follow the assumption that these three noise sources are similar, in terms of behavior to the two independent WGN sources from the mathematical model. Indeed, the superposition of multiple independent noise sources can be simplified to a WGN source, by the central limit theorem (see Gillespie (1992), Sect. 1.6).

With the IR sensors acting as the neighbors' detection mechanism, we can express the following expected theoretical results. On the limit cycle, each robot has three neighbors in average (himself and the two nearest neighbors). This is due to the small range, and the fact that the robots also block the IR rays; thus, only the closest neighbors can be detected. As on the limit cycle, the robots always have the light at $\frac{\pi}{2}$ on their right, so the value of the sensors can also be estimated. Indeed, sensors 1 and 2 are closest to the light source, and approximately at the same distance. Their values will most likely be around 1. Sensors 0 and 3 will also have nearly the same values, and the other sensors will not receive any light, so that:

$$\Delta S := S_R - S_L \approx S_1 + S_2 + S_3 - S_0 \approx 2.$$

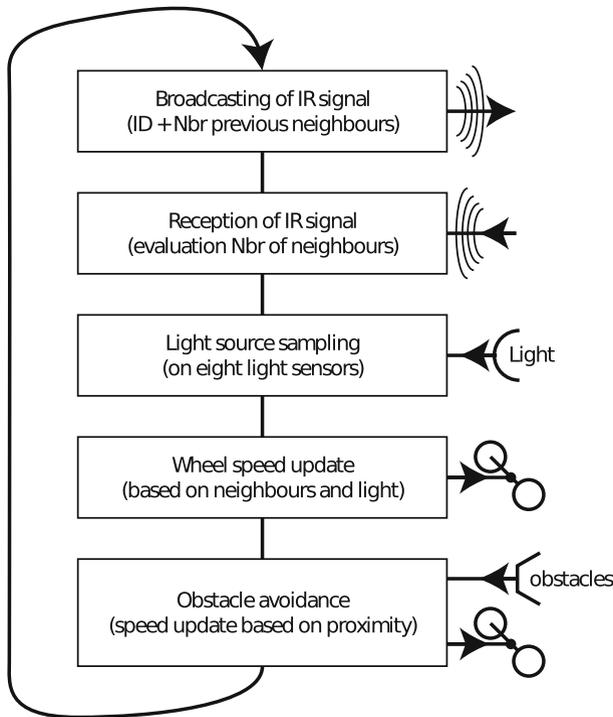


Fig. 5 Organigram of the robot’s main programming loop

This allows us to consider the average speeds on the limit cycle as being constants, and to rewrite the BCM on the limit cycle as:

$$\begin{cases} V_R = 3\alpha + 2\beta \\ V_L = 3\alpha - 2\beta. \end{cases} \tag{15}$$

A relation between these speeds and the interaxial distance of the robots, leads to the radius of the limit cycle

$$\mathcal{L}_{\alpha,\beta} = \frac{e}{2} \frac{V_L + V_R}{V_L - V_R} = \frac{3\alpha \cdot e}{4\beta} \tag{16}$$

The angular speed can also be expressed as

$$\omega_{\alpha,\beta} = \frac{V_L - V_R}{e} = \frac{4\beta}{e}.$$

3.3 Dictionary theoretical model-robotic implementation

A formal proof of the convergence of our BCM would be very difficult to provide; this is why we chose to illustrate that our BCM exhibits the same characteristic behavior as the mathematical model of Sect. 2, whose convergence we have formally proven. To better link the model and the robotic implementation, Table 1 illustrates the translation of these main characteristics between the theoretical and robotic control mechanisms.

Table 1 Theoretical and robotics implementations of the main control mechanisms in our model, and their respective control parameter (s)

Characteristic of the MCD dynamics	Theoretical control mechanism	Robotic control mechanism
Attraction/repulsion with respect to the origin	Symmetric part of the dynamical matrix \mathbb{V} Control parameter: M, ρ	Forward speed in the Braitenberg control mechanism Control parameter: α, ρ
Rotation around the origin	Anti-symmetric part of the dynamical matrix \mathbb{V} Control parameter: ω, ρ	Heading correction in the Braitenberg control mechanism Control parameter: β, ρ
Attraction to the limit cycle	$U'[H, t]$ term in the symmetric part of the dynamical matrix \mathbb{V} Control parameter: ρ	Transmission of $N_\rho(t - 1)$ to/from neighbors, and filtering step of $N_\rho(t)$ Control parameter: ρ , majority window size
Repartition on the limit cycle	Noise sources $dW(t)$. Control parameter: σ	Robotics noise sources (motors imprecisions, communications errors, environmental noise, etc.). Avoidance mechanism.

The explicit expression of the limit cycle in both cases is also summarized, to show how both depend on their control parameters

$$\mathcal{L}_{\rho, M} = \frac{\rho}{\sqrt{2 - 2 \cos(\pi/M)}} \qquad \mathcal{L}_{\alpha, \beta} = \frac{3\alpha \cdot e}{4\beta}.$$

In the BCM, we see that our initial choice of setting α and β fixes the radius of the limit cycle (and the angular speed on it). If we choose a larger swarm of robots, we should not see a difference in the radius as long as we do not change these two control parameters. Theoretically, we could see the swarm trying to arrange on a circle whose perimeter is too small for all the robots to fit. This means that we should see α and β as functions of N , to be sure that all robots can fit on the consensual limit cycle. We should have the perimeter of the limit cycle at least larger than N times the diameter d of one robot:

$$2\pi \cdot \mathcal{L}_{\alpha, \beta} = 2\pi \cdot \frac{3\alpha \cdot e}{4\beta} \geq N \cdot d.$$

This result from the BCM matches the mathematical model, where the radius of the limit cycle depends only on the fraction of neighbors M and the radius of observation ρ . In the mathematical model, changing the swarm number does not change the limit cycle’s radius either.

3.4 Experimental validation

To demonstrate the convergence of our BCM regardless of the initial conditions, tests were carried out with a swarm of eight robots, from 25 different (random) initial situations (positions and headings of the robots). The choice of these initial conditions was made to experimentally validate the inter- and intra-convergence variability. The headings changes in similarly located robots in different initial conditions (e.g., experiments 3 and 17) show how the control mechanism can achieve convergence to a consensual orbit with initial conditions

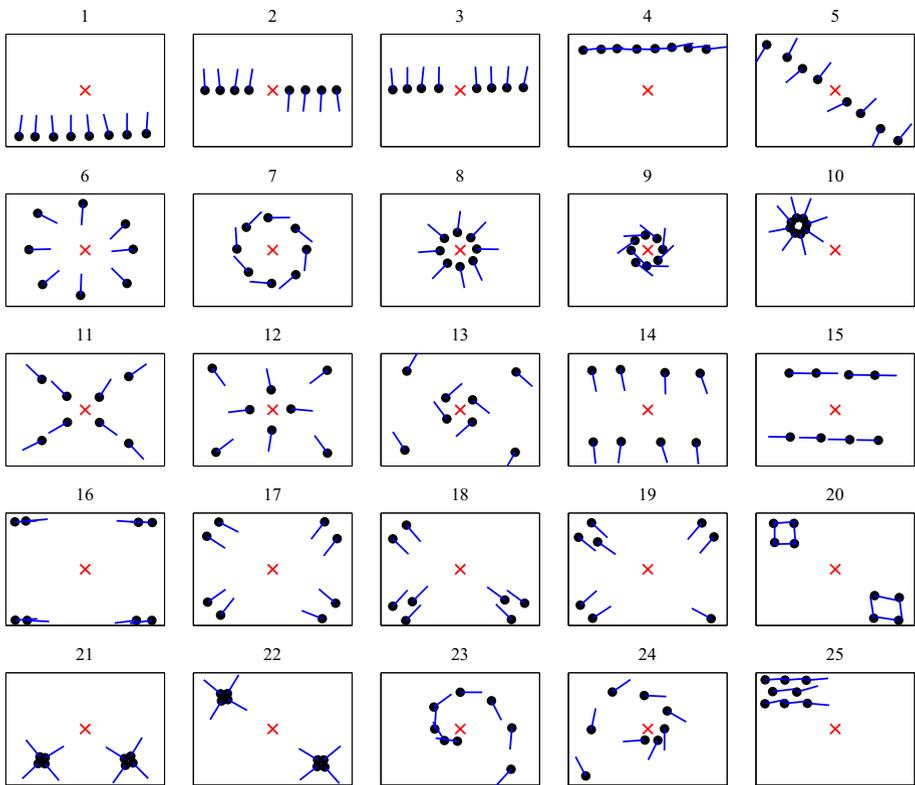


Fig. 6 Initial positions and headings of the eight robots in each of the 25 experiments. Each circle corresponds to a robot, and the corresponding line denotes the heading. The center cross represents the position of the light source

close or far from the final orbit. The agents were placed systematically in a $200 \times 200 \text{ cm}^2$ rectangular horizontal arena. The initial positions and headings of the robots are shown in Fig. 6, along with the position of the IR source, which remains unchanged during the experiments. Each experiment lasted from 1 to 4 minutes; long enough to let the robots rearrange from their initial positions to the limit cycle, and then maintain the limit cycle.

The tracking step allowed us to obtain the position of each robot through all of the 25 experiments. Figure 7 shows the tracking step applied to the videos, while Fig. 8 shows the resulting tracked trajectories. Details on the multi-agent tracking tool developed can be found in Appendix section “Multi-agent tracking tool”.

3.5 Experiment results

Data extracted from tracking have been used to quantitatively assess convergence, based on the following measures:

1. Mean and standard deviation of each robot’s radius in the last 15% of each experiment (empirically found to be in the limit cycle regime for every initial situation).
2. Mean and standard deviation of each robot’s angular speed in the same time window.

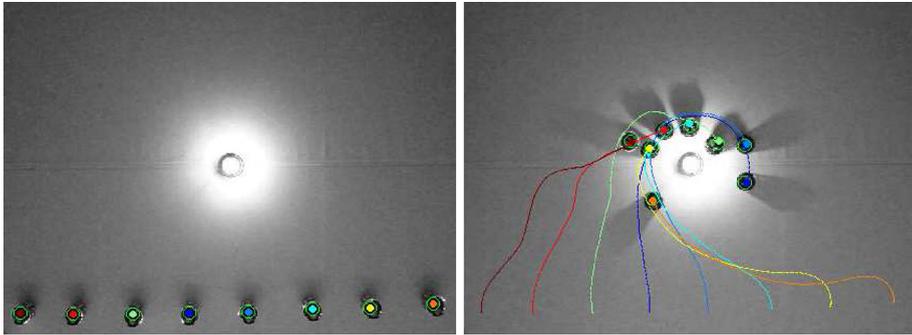


Fig. 7 *Left* A randomly selected initial condition assuming eight aligned equally spaced agents in the inferior working space region. The IR radiation source is located at the center of the rectangular arena. *Right* Respective agents’ control mechanism and their mutual interactions. It is important to observe that the proposed Kalman filter is capable of determining the position of an agent with high precision independently of lighting conditions and other agents’ positions

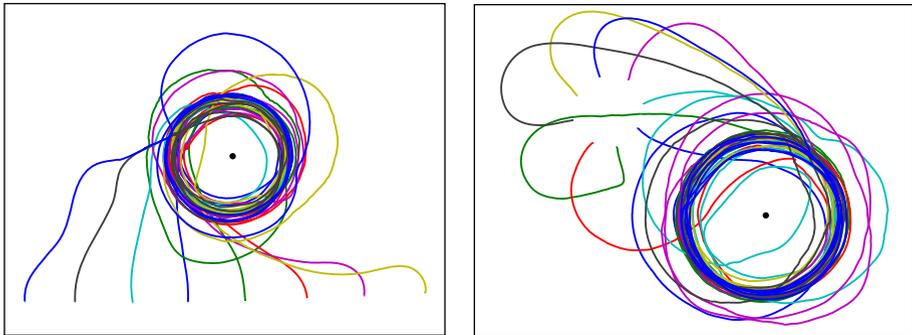


Fig. 8 Full extracted trajectories of the eight robots through the duration of experiments, for initial conditions 1 (*left*) and 10 (*right*)

The extracted data are shown in Fig. 9 for each experiment, where these averages and standard deviations are pictured for each robot on the left side, and as one average for the whole swarm on the right. It is worthwhile to note the very low fluctuations on the average radius and angular speeds for the swarm, through the whole set of initial conditions.

As a measure of the uniformity of the robots’ dispersion on the limit cycle, we now recall the order parameter (see [Acebrón et al. \(2005\)](#)) for a group of N oscillators on \mathbb{R}^2 in polar coordinates:

$$\mathcal{R}(t) = \left| \frac{1}{N} \sum_{i=1}^N r_i(t) \cdot e^{i\theta_i(t)} \right|, \tag{17}$$

which takes a real positive value. Note that if all the oscillators share the same phase $\theta_i(t)$ at a given time, the value for $\mathcal{R}(t)$ will be high. Conversely, if all the phases of the oscillators are uniformly distributed over $[0 : 2\pi]$, \mathcal{R} will vanish.

In this paper, we used information on the position in the last 15% of each run to determine the phase of each robot through time, and compute $\mathcal{R}(t)$ for the swarm. Figure 10 reports the average in this time window (along with the standard deviation) for each experiment.

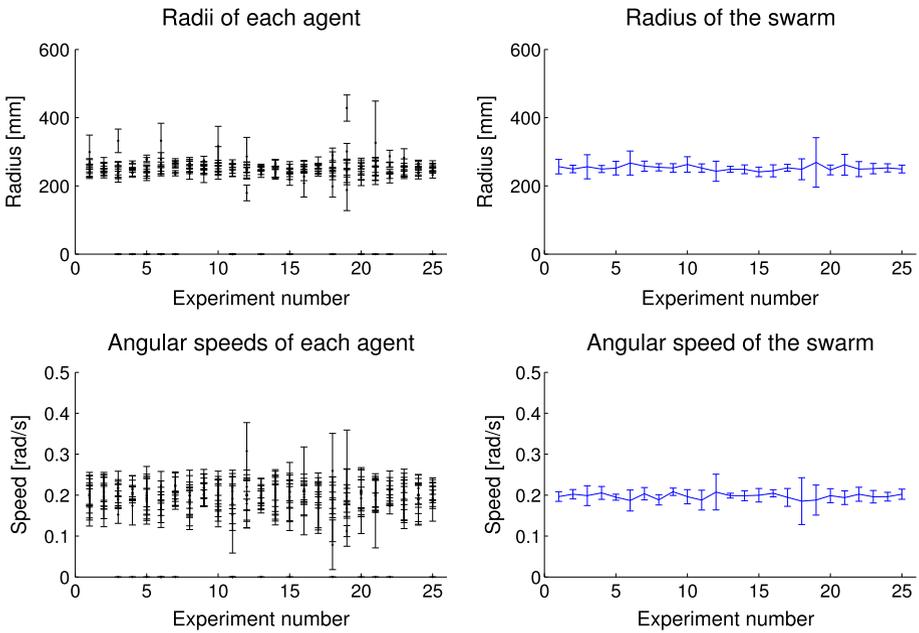


Fig. 9 Average radius/angular speed in the last 15 % of each experiment. *Left* For each robot. *Right* Averaged for the whole swarm

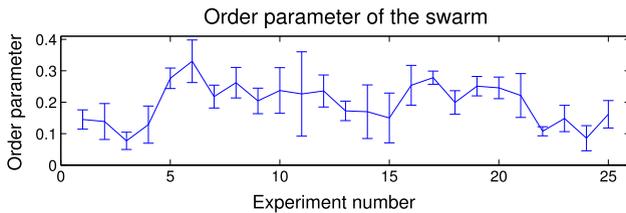


Fig. 10 Order parameter \mathcal{R} of the swarm for each initial situation, averaged over the last 15 % of each experiment

Remark: The positions of robots that got stuck against the IR source during an experiment are shown in the initial configurations plot in Fig. 6, even though their data were not used in the analysis. Over the 25 experiments, 15 robots got stuck, with a maximum of 2 stuck robots for one initial condition (7.5% of unusable data). As the IR source is an attractive point, and because of the avoidance mechanism used, the robots sometimes change their course and end up hitting the light source. A robot in this situation cannot get back to the limit cycle, and stays at the same spot with 0 velocity until the end of the experiment. In the mathematical model of Sect. 2, agents can leave the attractor only because of the noise source. Our robots cannot do the same, due to the physical presence of the IR source blocking their path.

This issue could not be solved by using a different avoidance mechanism, as avoiding a light source, solely with the use of the IR sensors is impossible (at short range, the IR sensors are saturated by the light source). This result conforms to our expectations, as it fits the theoretical predictions as well as the underlying Eq. (1).

3.6 Comparison with the theoretical results

Theoretical results derived in Sect. 3.2 lead to the following limit cycle radius and angular speed (with the e-puck robot’s interaxial distance $e = 52[mm]$):

$$\mathcal{L}_{\alpha,\beta} = \frac{3\alpha \cdot e}{4\beta} = \frac{(100 + 3 \cdot 100) \cdot 52}{4 \cdot 20} = 260 [mm] \quad \omega_{\alpha,\beta} = \frac{4\beta}{e} = \frac{4 \cdot \frac{20}{1000} \cdot 41\pi}{52} \approx 0.198. \tag{18}$$

These results very closely match our experimental results, where the average radius of the limit cycle over the 25 experiments is $\overline{\mathcal{L}_{\alpha,\beta}} := 252.7 [mm]$, and the average angular speed is $\overline{\omega_{\alpha,\beta}} := 0.197$.

3.7 Generalization to other Hamiltonian functions

The choice of a BCM limits the characteristics of the Hamiltonian functions that can be adapted to a hardware implementation. The scope of this paper was not to study the limit of validity of our model; other types of control mechanism may adapt to general Hamiltonian functions. Nevertheless, the BCM described here for the case of the harmonic oscillator ($H(x, y) = \frac{1}{2}(x^2 + y^2)$) can be extended to more complex Hamiltonian functions. For example, one can obtain Cassini oval-like curves by placing the robots in an environment with two light sources. Using exactly the same control mechanism of Eq. (12), we obtain a swarm behavior similar to the mathematical case with the Cassini ovals Hamiltonian function. Depending on the observation range ρ of the robots, cases with one connected consensual orbit or two separated orbits can arise. This time, robots sample the ambient light coming from both light sources, acting as the two focal points from the Cassini Ovals Hamiltonian function.

Tests have not yet been carried out on an actual swarm of robots, but numerical results are shown in Fig. 11. These results are obtained by simulating Eq. (12) with $dt = 0.1s$, on a swarm of 100 robots starting with random positions and headings on the plane. The two light sources are positioned at $(\pm 100, 0)$, and WGN sources are added to the (exact) gathered information to simulate the noise actual robots would experience:

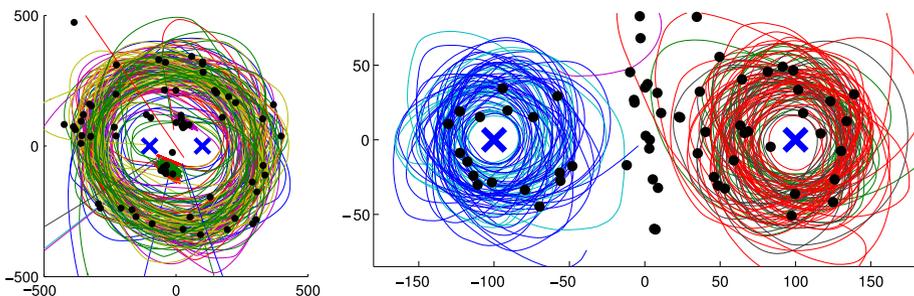


Fig. 11 Numerical simulation of the BCM on a swarm of 100 robots, with two light sources located at $(\pm 100, 0)$. With an observation range of $\rho = 100$, the swarm consensually rotates around both light sources (*left*), whereas a smaller range of $\rho = 10$ lets the swarm split into two sub-swarms rotating around each light source (*right*). The trajectory of 10 randomly selected robots are shown, along with the final position of all robots as *black dots*

$$\begin{cases} V_R(t + 1) = \alpha \cdot N_\rho(t) + \beta \cdot (S_L(t) - S_R(t) + \sigma_I dW_R(t)) \\ V_L(t + 1) = \alpha \cdot N_\rho(t) + \beta \cdot (S_R(t) - S_L(t) + \sigma_I dW_L(t)) \end{cases} \tag{19}$$

Empirically chosen values for the WGN sources are $\sigma_I = 0.05$ for the light source sampling, and $\sigma_N = 0.5$ for the number of instantaneous neighbors ($\tilde{N}_V(t) := N_V(t) + \sigma_N dW_N(t)$).

4 Conclusion

For a whole class of interacting autonomous agents evolving in an environment subject to noise or external perturbations, we have been able to perform an analytical characterization corroborated by an extensive set of simulation experiments and ultimately to implement the control mechanism on a swarm composed of eight e-puck robots. The possibility to simultaneously complete these complementary approaches on the same class of control mechanisms is exceptional, and originates from the simplicity of the underlying dynamics (i.e., MCD dynamics). However, despite its simplicity, our modeling framework incorporates a number of basic features rendering it generic, namely: non-linearities in the control mechanism; finite range interactions between agents; self-organization mechanisms leading to dynamic consensual spatio-temporal patterns; and environment noise, which corrupts ideal trajectories and sensor measurements. Beyond the pure pedagogic insights offered by this class of models, the results provide strong evidence that built-in stability of the control mechanism and the resilient behavior observed during the actual implementation open the door for more elaborate implementations. The modeling framework exposed here is more than a mere class of models; it offers a constructive method to deduce generalized classes of fully solvable agent dynamics. In particular, extension of the MCD dynamics via the introduction of Hamiltonian functions, involving more degrees of freedom, will allow analysis of the robots' evolution in higher dimension (from planar to 3-D spaces).

Acknowledgments Detailed comments from anonymous referees strongly contributed to the presentation and quality of this paper. Guillaume Sartoretto and Max-Olivier Hongler are partly funded by the Swiss National Science Foundation and the ESF project “*Exploring the Physics of Small Devices*”. Marcelo Elias de Oliveira is supported by the EU-ICT-FET project “ASSISIBf”, No. 601074 and by the ESF project “H2Swarm”. This last project is funded by the Swiss National Science Foundation under grant number 134317.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

Appendix: MCD Dynamics

Consider the dynamics of a harmonic oscillator with Hamiltonian function $H(X, Y) = \frac{1}{2} [X^2 + Y^2]$. The resulting dynamics reads:

$$\begin{cases} \dot{X} = +Y = +\frac{\partial}{\partial Y} H(X, Y) \\ \dot{Y} = -X = -\frac{\partial}{\partial X} H(X, Y) \end{cases} = \underbrace{\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}}_{\mathbb{V}_0} \times \begin{pmatrix} \frac{\partial}{\partial X} H(X, Y) \\ \frac{\partial}{\partial Y} H(X, Y) \end{pmatrix}. \tag{20}$$

Clearly, the trajectories solving Eq. (20) are circles on the plane \mathbb{R}^2 with radius given by the initial conditions (X_0, Y_0) . We shall now introduce energy dissipation into the nominally (energy-conserving) Hamiltonian system Eq. (20). This will be achieved by adding diagonal terms to the matrix \mathbb{V}_0 :

$$\begin{aligned} \begin{cases} \dot{X} = +Y - \gamma X = +\frac{\partial}{\partial Y} H(X, Y) + \overbrace{U'(H) \frac{\partial}{\partial X} H(X, Y)}^{\text{dissipative term}} \\ \dot{Y} = -X - \gamma Y = -\frac{\partial}{\partial X} H(X, Y) + \overbrace{U'(H) \frac{\partial}{\partial Y} H(X, Y)}^{\text{dissipative term}} \end{cases} \\ = \underbrace{\begin{pmatrix} U'(H) + 1 \\ -1 & U'(H) \end{pmatrix}}_{\mathbb{V}} \times \begin{pmatrix} \frac{\partial}{\partial X} H(X, Y) \\ \frac{\partial}{\partial Y} H(X, Y) \end{pmatrix}. \end{aligned} \tag{21}$$

where $U(H)$ is a function of the Hamiltonian and we denote $U'(H) := \frac{d}{dH} U(H)$. For instance, the choice $U(H) = -H \Rightarrow U'(H) = -1$ simply yields a linearly damped dynamics. In this case, the trajectories will simply be spirals on \mathbb{R}^2 , approaching the origin as time t increases. Observe from Eqs. (20) and (21) that the conservative part of the dynamical vector field (VF) is orthogonal to the dissipative part: we have the vanishing scalar product

$$\underbrace{\left(+\frac{\partial}{\partial Y} H(X, Y), -\frac{\partial}{\partial X} H(X, Y) \right)}_{\text{conservative VF}} \cdot \underbrace{\left(U'(H) \frac{\partial}{\partial X} H(X, Y), U'(H) \frac{\partial}{\partial Y} H(X, Y) \right)}_{\text{dissipative VF}} = 0. \tag{22}$$

The dynamics of Eq. (21) will finally be driven by a couple of external noise sources $dW_X(t), dW_Y(t)$ that will be taken as standard independent WGN processes:

$$\begin{cases} \dot{X} = +Y - \gamma X = +\frac{\partial}{\partial Y} H(X, Y) + U'(H) \frac{\partial}{\partial X} H(X, Y) + \sigma dW_X(t) \\ \dot{Y} = -X - \gamma Y = -\frac{\partial}{\partial X} H(X, Y) + U'(H) \frac{\partial}{\partial Y} H(X, Y) + \sigma dW_Y(t) \end{cases} \tag{23}$$

Accordingly, the planar dynamics in Eq. (23) simultaneously model: (i) a canonical driving, (ii) an energy dissipation mechanism, and (iii) an energy source delivered by the noise sources. The use of WGN implies that $X(t)$ and $Y(t)$ are Markovian processes, for which all probabilistic characteristics of the realizations are entirely given by transition probability densities $P(x, y, t \mid x_0, y_0, t_0) := P$ for $t > t_0$. Moreover, $P(x, y, t \mid x_0, y_0, t_0)$ obeys the FP equation:

$$\begin{aligned} \frac{\partial}{\partial t} P = & -\frac{\partial}{\partial x} \left\{ \left[+\frac{\partial}{\partial y} H(x, y) + U'(H) \frac{\partial}{\partial x} H(x, y) \right] P \right\} \\ & -\frac{\partial}{\partial y} \left\{ \left[-\frac{\partial}{\partial x} H(x, y) + U'(H) \frac{\partial}{\partial y} H(x, y) \right] P \right\} \\ & + \frac{\sigma^2}{2} \left[\frac{\partial^2}{\partial x^2} P \right] + \frac{\sigma^2}{2} \left[\frac{\partial^2}{\partial y^2} P \right]. \end{aligned} \tag{24}$$

Asymptotically in time in Eq. (23), a balance between dissipation and supply of energy is reached and this leads to a stationary regime, in which all probabilistic features of the dynamics become time invariant. This stationary regime is described by the stationary probability measure $P_s(x, y)$ solving the stationary (FP) that results when the left-hand side in

Eq. (24) equals zero. The cylindrical symmetry of Eq. (23) implies that, in the stationary regime, all probabilistic features of the model must themselves be cylindrically invariant; hence, $P_s(x, y) = P_s[H(x, y)] = P_s(H)$. By direct calculation, we can indeed verify that

$$P_s(H) = \mathcal{N} e^{\frac{2}{\sigma^2}U(H)}, \quad (\text{with } \mathcal{N} \text{ the normalization factor}). \quad (25)$$

Now, provided that Eq. (25) is integrable, meaning that $\mathcal{N} < \infty$ (i.e., implying an overall dissipative dynamics), the stationary probability characteristics of the dynamics are given by Eq. (25). In particular, for the linearly dissipative case obtained when $U(H) = -H$, we consistently obtain a bivariate Gaussian density with cylindrical symmetry:

$$P_s(H) = \mathcal{N} e^{\frac{2}{\sigma^2}U(H)} = \frac{2}{\pi\sigma^2} e^{-\frac{1}{\sigma^2}[x^2+y^2]}.$$

Multi-agent tracking tool

To determine the states of every single agent in our experiments, we developed a multi-agent tracking tool based on computer vision techniques and the Kalman filter. This task was particularly challenging due to the very inhomogeneous lighting conditions in the acquired videos (mainly because of the central light source). This acquisition setup is composed of an overhead digital camera equipped with a wide angle lens (Canon EF-S 10-22mm f/3.5-4.5 USM, mounted on a Canon EOS 7D body). Our multi-agent tracking tool has been developed in C++ using the GNU compiler collection (GCC 4.6.3) and standard computer vision algorithms included in the open source computer vision library (OpenCV, v. 2.3.1-7, <http://opencv.org/>).

The employed technique follows three steps:

1. The image foreground is extracted for further processing using a precomputed mean background.
2. The Gaussian scale-space representation of the extracted foreground $I_F(x, y)$ containing the agents was convolved with a Gaussian kernel, as $L(x, y, \sigma) = G(x, y, \sigma) * I_F(x, y)$, where, in a two-dimensional space, the Gaussian function is given by

$$G(x, y; \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (26)$$

and σ is the standard deviation. The outcome of this convolution at a specific scale is strongly dependent on the relationship between the sizes of the blob structures in the image domain. Since our background is characterized by high pixel values and the blobs dimensions are known, every agent can be characterized by a local extrema of the resulting Gaussian scale-space representation $L(x, y, \sigma)$.

3. Assuming that the initial frame does not present any false positives, every single detected blob in the initial frame represents an agent (see Fig. 7, left). The detected spatial locations of the agents are used as inputs for the Kalman filter Kalman (1960), where the initial speeds of the agents are assumed to be zero. The estimates of the system states are based on the initial conditions and the sequence of the detected blobs along the experiments. Since the radius of the cylindrical robots are known, as well as their directions and linear speeds, a new measurement is easily casted as a good candidate input for the Kalman filter or as a new input for the designed regressive filter.

In this work, the agents are assumed to follow non-uniform translational motions, with a constant acceleration between consecutive frames. The uncertainties associated with the measurements of the positions induced by different lighting conditions are treated as white noise. The states x_k^i of the i -th agent at time k as well as the measurement z_k^i are described by linear SDE in the image domain:

$$\begin{cases} x_k^i = \Phi x_{k-1}^i + w_{k-1} \\ z_k^i = H x_k^i + v_k \end{cases} \quad (27)$$

where $x_k^i \in \mathfrak{R}^4$, $z_k^i \in \mathfrak{R}^2$.

w_{k-1} and v_k are WGN sources, respectively, representing the process and measurement noises. They are assumed to be mutually statistically independent, and to be drawn from zero-mean multivariate Gaussian distributions with covariances Q and R . The covariances' matrices can be described as a function of the experimental parameters, such as the lighting variability across the experiments, the number of agents, etc. However, in our experiments, they were assumed to be constant. The evolution of the current state vector x_k^i over time is described by the transition matrix Φ derived from the non-uniform translational motion equations

$$\Phi = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (28)$$

where Δt is equal to 0.1 s, as the videos were acquired at a constant frequency equal to 10 Hz, the time elapsed between consecutive frames was assumed to be constant.

H is a 2×4 transformation matrix that related the predicted state vectors x_k^i into the measurement domain:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \quad (29)$$

Remark: The videos of all the experiments can be found online, at http://www.youtube.com/channel/UCCgwiDNHoyG9RJ2_pdyewmg.

References

- Acebrón, J. A., Bonilla, L. L., Vicente, C. J. P., Ritort, F., & Spigler, R. (2005). The kuramoto model: A simple paradigm for synchronization phenomena. *Reviews of Modern Physics*, 77(1), 137–185.
- Berman, S., Halasz, A., Hsieh, M., & Kumar, V. (2009). Optimized stochastic policies for task allocation in swarms of robots. *IEEE Transactions on Robotics*, 25(4), 927–937.
- Berman, S., Kumar, V., & Nagpal, R. (2011). Design of control policies for spatially inhomogeneous robot swarms with application to commercial pollination. In 2011 IEEE International Conference on Robotics and Automation (pp. 378–385). Piscataway, NJ: IEEE.
- Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: From natural to artificial systems*. New York, NY: Oxford University Press.
- Braitenberg, V. (1984). *Vehicles: Experiments in synthetic psychology*. Cambridge, MA: MIT Press.
- Brambilla, M., Ferrante, E., Birattari, M., & Dorigo, M. (2013). Swarm robotics: A review from the swarm engineering perspective. *Swarm Intelligence*, 7(1), 1–41.
- Brambilla, M., Pinciroli, C., Birattari, M., & Dorigo, M. (2012). Property-driven design for swarm robotics. *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - (Vol. 1, pp. 139–146)*. Richland, SC: AAMAS '12 International Foundation for Autonomous Agents and Multiagent Systems.
- Gillespie, D. T. (1992). *Markov Processes*. Waltham, MA: Academic Press.

- Halloy, J., Sempo, G., Caprari, G., Rivault, C., Asadpour, M., Tâche, F., et al. (2007). Social integration of robots into groups of cockroaches to control self-organized choices. *Science*, 318(5853), 1155–1158.
- Hamann, H., & Wörn, H. (2008). A framework of space time continuous models for algorithm design in swarm robotics. *Swarm Intelligence*, 2(2–4), 209–239.
- Hongler, M. O., & Rytter, D. M. (1978). Hard mode stationary states generated by fluctuations. *Zeitschrift für Physik B Condensed Matter and Quanta*, 31(3), 333–337.
- Hsieh, M. A., Loizou, S., & Kumar, V. (2007). Stabilization of multiple robots on stable orbits via local sensing. In: Proceedings—IEEE International Conference on Robotics and Automation, (pp. 2312–2317). Piscataway, NJ: IEEE.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems 1. *Journal of Basic Engineering in D*, 82, 35–45.
- Kernbach, S. (Ed.). (2013). *Handbook of collective robotics: Fundamentals and challenges*. New York: Pan Stanford Publishing.
- Kernbach, S., Häbe, D., Kernbach, O., Thenius, R., Radspieler, G., Kimura, T., et al. (2013). Adaptive collective decision-making in limited robot swarms without communication. *International Journal of Robotics Research*, 32(1), 35–55.
- Leonard, N. E. (2013). Multi-agent system dynamics: Bifurcation and behavior of animal groups? In *IFAC Proceedings Volumes (IFAC-PapersOnline)* (Vol. 9, pp. 307–317). Amsterdam: Elsevier.
- Leonard, N. E., Paley, D. A., Lekien, F., Sepulchre, R., Fratantoni, D. M., & Davis, R. E. (2007). Collective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE*, 95(1), 48–74.
- Lerman, K., Martinoli, A., & Galstyan, A. (2005). A review of probabilistic macroscopic models for swarm robotic systems. In: *Lecture Notes in Computer Science* (Vol. 3342, pp. 143–152). New York, NY: Springer.
- Martinoli, A., Ijspeert, A. J., & Mondada, F. (1999). Understanding collective aggregation mechanisms: From probabilistic modelling to experiments with real robots. *Robotics and Autonomous Systems*, 29(1), 51–63.
- Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., et al. (2006). The e-puck, a robot designed for education in engineering. *Robotics*, 1(1), 59–65.
- Mondada, F., Martinoli, A., Correll, N., Gribovskiy, A., Halloy, J. I., Siegart, R., et al. (2011). A general methodology for the control of mixed natural-artificial societies. In S. Kernbach (Ed.), *Handbook of collective robotics* (pp. 399–428). New York: Pan Stanford.
- Pimenta, L. C. D. A., Pereira, G. A. S., Michael, N., Mesquita, R. C., Bosque, M. M., Chaimowicz, L., et al. (2013). Swarm Coordination Based on Smoothed Particle Hydrodynamics Technique. *IEEE Transactions on Robotics*, 29, 383–399.
- Prorok, A., Correll, N., & Martinoli, A. (2011). Multi-level spatial modeling for stochastic distributed robotic systems. *The International Journal of Robotics Research*, 30, 574–589.
- Sabattini, L., Secchi, C., & Fantuzzi, C. (2012). Closed-curve path tracking for decentralized systems of multiple mobile robots. *Journal of Intelligent & Robotic Systems*, 71(1), 109–123.
- Schwager, M., Michael, N., Kumar, V., & Rus, D. (2011). Time scales and stability in networked multi-robot systems. In: *Proceedings—IEEE International Conference on Robotics and Automation* (pp. 3855–3862). Piscataway, NJ: IEEE.
- Schweitzer, F. (2003). *Brownian agents and active particles*. New York, NY: Springer.
- Sepulchre, R., Paley, D. A., & Leonard, N. E. (2007). Stabilization of planar collective motion: All-to-all communication. *IEEE Transactions on Automatic Control*, 52(5), 811–824.
- Tsiotras, P., & Castro, L. I. R. (2011). Extended multi-agent consensus protocols for the generation of geometric patterns in the plane. In *Proceedings of the 2011 American Control Conference* (pp. 3850–3855). Piscataway, NJ: IEEE.
- Yates, R. C. (1947). *A handbook on curves and their properties*. Reston, VA: National Council of Teachers of Mathematics.