Emergence of Consensus in a Multi-Robot Network: from Abstract Models to Empirical Validation

Vito Trianni[1], Daniele De Simone[2], Andreagiovanni Reina[3] and Andrea Baronchelli[4]

***Abstract*—Consensus dynamics in decentralised multiagent systems are subject to intense studies, and several different models have been proposed and analysed. Among these, the *naming game* stands out for its simplicity and applicability to a wide range of phenomena and applications, from semiotics to engineering. Despite the wide range of studies available, the implementation of theoretical models in real distributed systems is not always straightforward, as the physical platform imposes several constraints that may have a bearing on the consensus dynamics. In this paper, we investigate the effects of an implementation of the naming game for the *kilobot* robotic platform, in which we consider concurrent execution of games and physical interferences. Consensus dynamics are analysed in the light of the continuously evolving communication network created by the robots, highlighting how the different regimes crucially depend on the robot density and on their ability to spread widely in the experimental arena. We find that physical interferences reduce the benefits resulting from robot mobility in terms of consensus time, but also result in lower cognitive load for individual agents.**

***Index Terms*—Swarms; Agent-Based Systems; Distributed Robot Systems**

I. Introduction

COLLECTIVE decision-making is an essential capability of large-scale decentralised systems like robot swarms, and is often key to achieve the desired goal. In swarm robotics, a large number of robots coordinate and cooperate to solve a problem, and often consensus among the robots is necessary to maximise the system performance [1], [2], [3]. The design of controllers for consensus decision is often inspired by models of collective behaviour derived from studies in the ethology of social systems [4], [5], as well as from studies about the emergence of social conventions and cultural traits [6], [7], [8]. Theoretical models represent idealised instances of collective decentralised systems in which consensus can be somehow attained. Among the different available models, a particularly interesting case is the one of the *naming game* (NG), which represents the emergence of conventions in social systems, such as linguistic, cultural, or economic conventions [9], [10], [11]. The appeal of this model consists in the ability to describe the emergence of consensus out of a virtually infinite set of equivalent alternatives, yet requiring minimal cognitive load from the agents composing the system [10], [12]. Moreover, the NG has been successfully demonstrated on a network of mobile point-size agents [13]. Such a collective decision-making behaviour can be very useful in swarm robotics in case consensus is required with respect to a possibly large number of alternatives (e.g., the location and structure for cooperative construction [14], [15], or the most functional shape for self-assembly [16], [17]).

When dealing with the implementation of physical systems starting from theoretical models, however, several constraints may arise which may have a bearing on the collective dynamics. Indeed, small implementation details at the microscopic scale may have a large impact at the macroscopic level. Hence, it is important to study the effects of such constraints in relation to the dynamics predicted by the theoretical models. In this paper, we propose an implementation of the NG for the *kilobot* robotic platform [18]. Kilobots are low-cost autonomous robots designed for experimentation with large groups [17]. They can move on a flat surface and interact with close neighbours by exchanging short messages sent on an infrared channel. The collective behaviour of a kilobot swarm results solely from the individual decisions and inter-individual interactions, without any central unit directing the group dynamics. As a consequence, the implementation of the NG for the kilobots needs to be fully decentralised with games autonomously triggered by any robot at any time. Additionally, within a decentralised system, the concurrent execution of games by neighbouring robots becomes possible, in opposition to the rigorously sequential scheme typically adopted in theoretical studies. Hence, the interaction pattern among robots may be significantly altered, and the corresponding dynamics need to be carefully characterised. Finally, the embodiment of the robots determines physical interferences (i.e., collisions) that strongly influence the overall mobility pattern. It follows that abstract models of agent mobility must be contrasted with experimentation with robots, in which all the details of the physical platform can be taken into account.

In this paper, we study the effects of the motion and interaction patterns on the consensus dynamics, and we pay particular attention to both concurrent executions of games and physical interferences. First, we provide an abstract model of

Manuscript received: August 30, 2015; Revised: December 3, 2015; Accepted: January 5, 2016.

This paper was recommended for publication by Editor Nak Young Chong upon evaluation of the Associate Editor and Reviewers' comments. Vito Trianni acknowledges support from the project DICE (FP7 Marie Curie Career Integration Grant, ID: 631297).

[1]Vito Trianni is with the Institute of Cognitive Sciences and Technologies (ISTC), National Research Council (CNR), 00185 Rome, Italy vito.trianni@istc.cnr.it

[2]Daniele De Simone is with DIAG, the Department of Computer, Control and Management Engineering Antonio Ruberti at Sapienza University of Rome, 00185 Rome, Italy desimone@diag.uniroma1.it

[3]Andreagiovanni Reina is with the Department of Computer Science, University of Sheffield, Sheffield S1 4DP, UK a.reina@sheffield.ac.uk

[4]Andrea Baronchelli is with the City University London, London EC1V 0HB, UK a.baronchelli.work@gmail.com

Digital Object Identifier (DOI): see top of this page.

mobile agents playing the NG, in which physical interferences are ignored. Following previous studies [13], we analyse this model in the light of the communication network established by the agents, we show how the consensus dynamics are determined by agent density, mobility and interaction frequency, and we link our empirical findings with theoretical studies [12], [19]. Then, we contrast abstract models with large-scale simulations of the kilobots, as well as with real-world experiments. Here, physical interferences impact on the consensus dynamics by limiting the free diffusion of robots in the experimental arena, hence increasing consensus times. Still, the cognitive load for the individual agents is reduced for physical implementations, due to the lower number of alternatives that each agent must consider in average. Our adapted implementation of the NG is presented in Section II, while the corresponding consensus dynamics are discussed in Section III. Conclusions and future directions are presented in Section IV.

II. Model and Implementations

The naming game in its basic form [10] models pair-wise interactions in which two players—the speaker and the hearer—interact by exchanging a single word chosen by the speaker, and updating their inventory on the basis of the game success. Previous extensions of the model take into account different inventory updating and communication schemes [20] and also consider mobile agents [13]. In this work, we adopt a broadcasting scheme for the speaker agent, while inventory updating is performed only by the hearer agent, as detailed in the following (for details, see [20]).

When engaging in a NG, the speaker agent a_s selects a word w either randomly from its inventory, or inventing it anew should the inventory be empty (i.e., the set of possible choices for a new word w is virtually infinite). Then, it broadcasts w to all agents in its neighbourhood. Upon reception of w, the hearer agent a_h updates its inventory by either storing w if it was not found in a_h's inventory, or by removing all words but w if the latter was already known to a_h. By iterating the game multiple times, the entire system converges toward the selection of a single word shared by all agents [10], [20].

A. Multiagent simulations

We implement a decentralised version of the NG by letting each agent a autonomously take the role of speaker every τ_s s. Given that agents update their state at discrete steps of $\delta_t = 0.1\,\mathrm{s}$, they communicate every n_s steps so that a word is broadcast every $\tau_s = n_s \delta_t$ s to all neighbours within the range $d_i = 10\,\mathrm{cm}$. In this way, concurrent execution of games becomes possible, hence introducing an important difference from previous theoretical studies in which at any time only one game is executed by a randomly chosen agent and one of its neighbours [13], [10], [20], [19]. At the hearer side, multiple interactions are possible within any time interval δ_t, depending on the local density of agents. Hence, all words received in a single δ_t period are used sequentially to update the inventory. The list of received words is randomised before usage to account for the asynchronous reception of messages by the physical platform (see Section II-C). The algorithm for the multiagent implementation is shown in Fig. 1.

```
procedure NG(n_m, n_s)                  ▷ Implementation of the NG
    n_t ← n_t + 1
    if n_t mod n_m = 0 then             ▷ Change motion direction
        RandomTurn()
    end if
    MoveStraight()
    W ←ReceiveWords()                   ▷ Play the hearer role
    Randomise(W)
    for w ∈ W do
        UpdateInventory(w)
    end for
    if n_t mod n_s = 0 then             ▷ Play the speaker role
        w ← SelectWord()
        Broadcast(w)
    end if
end procedure
```

Fig. 1. The NG algorithm exploited in multi-agent simulations.

At the beginning of each simulation run, N agents are deployed uniformly random within a squared box of side L with periodic boundary conditions (e.g., a torus). This allows to focus on the effects of agent mobility and density without constraints from a bounded space [13]. Agents are dimensionless particles and therefore do not collide with each other. The agents neighbourhood is determined by all the agents within the interaction range d_i. By moving in space, the agent neighbourhood varies so that a dynamic communication network is formed. Agent mobility follows an uncorrelated random walk scheme, with constant speed $v = 1\,\mathrm{cm/s}$ and fixed step length $v\tau_m$, where τ_m represents the constant time interval between two consecutive uncorrelated changes of motion direction. This leads to a diffusive motion with coefficient $D \sim v^2 \tau_m$ [13]. In practice, agents change direction every n_m steps, so that $\tau_m = n_m \delta_t$ (see Fig. 1).

B. Robot simulations

We have developed a custom plugin for simulating kilobots within the ARGoS framework [21], paying particular attention to match the real robot features in terms of body size, motion speed and communication range. Communication is implemented by allowing the exchange of messages between neighbours within the range d_i. No failures in communication have been simulated, assuming that the channel can support communication even with high densities. We will discuss this choice in the light of the obtained results in Section IV.

Concerning the motion pattern, kilobots are limited to three modes of motion: forward motion when both left and right motors are activated, and left or right turns when only one motor is activated. Turning is performed while pivoting on one of the kilobot legs. We have therefore implemented a differential drive motion scheme centred between the two backward legs of the kilobot, with speed $v = 1\,\mathrm{cm/s}$ for forward motion and angular speed $\omega = \pi/5\,\mathrm{s}^{-1}$ for turning. A multiplicative gaussian noise applied at every simulation cycle (standard deviation $\sigma = 0.4$) simulates the imprecise motion of kilobots. With such an implementation, the individual motion is still diffusive, but with a lower coefficient due to the delay

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2016.2519537, IEEE Robotics and Automation Letters

introduced by turning. Additionally, collision avoidance is not possible with the kilobot onboard sensors, and robots are let free to crash into walls and each other. The ARGoS framework provides a 2D dynamics physics engine that handles collisions between robots and with walls with an integration step size $\delta_t = 0.1\,\mathrm{s}$, which proves sufficient for our purposes. Collisions determine a further reduction in the diffusion speed, as we will discuss in Section III-B. Robots are deployed randomly within a squared box of side L surrounded by walls. To avoid overlapping of robots, the initial positions are determined by dividing the arena in cells wide enough to contain a single kilobot, and randomly placing kilobots into free cells.

C. Kilobot implementation

The implementation of the NG for kilobots requires handling transmission and reception of messages, and implementing the random walk. We use the kilobot API from Kilobotics [22], which provides two callback functions for transmission and reception of 10-byte messages, functions for distance estimation of the message source, and a counter that is updated approximately 32 times per second (i.e., $\delta_t \simeq 1/32$). Broadcast is allowed every τ_s seconds by opportunely activating the transmission callback. Communication interferences among robots are treated through the CSMA-CD protocol (carrier-sensing multiple access with collision detection) with exponential back-off, meaning that upon detection of the occupied channel, message sending is delayed within an exponentially increasing range of time slots. This introduces an additional level of asynchrony that must be tolerated by the collective decision-making process, as the exact timing of communication cannot be completely controlled. Upon reception of any message, the corresponding callback function is activated, and the NG is immediately played exploiting the content of the received message. Given that the maximum communication distance may vary across different robots, we capped the maximum distance to d_i by software, estimating the source distance and ignoring messages from sources farther than d_i.

The motion pattern implements the random walk exactly as performed in simulation, exploiting the internal random number generator for uniformly distributed turning angles. Forward motion v and angular speed ω of each kilobot have been calibrated to obtain a roughly constant behaviour across different robots and to match the parameter values used in simulation. The code for the controller is written in a `C`-like language (`AVR C`) and fits in about 200 lines: it is available in the multimedia material at http://ieeexplore.ieee.org. In experimental runs, kilobots are initially positioned randomly following indications from the ARGoS simulator in equivalent conditions. This provides an unbiased initialisation and supports comparison with simulations in Section III-C.

III. Consensus Dynamics

The most important quantity to evaluate the consensus dynamics following the NG process is the time of convergence t_c, i.e., the time required for the entire group to achieve consensus. Previous studies demonstrated that consensus is the only possible outcome, even though in particular cases it can be reached only asymptotically [10]. Another relevant metric for the NG in multiagent systems is the maximum memory M required for the agents, in average, until convergence: given that each agent needs to store a possibly large number of words, it is important to study how the memory requirements scale with the system size, especially in the perspective of the implementation for real robots that entail limited memory and minimal processing power to search large inventories.

Following previous studies, it is useful to look at the (static) interaction network resulting by linking all agents that are within interaction range. Given N agents confined in a $L \times L$ space and interacting over a range d_i, the resulting network has average degree $\langle k \rangle = \pi N d_i^2 / L^2$ [23], [13]. Given that in our case all parameters are constant but the agent density (as determined by N), two values are critical:

1) $N_1 = N_{\langle k \rangle = 1}$ is the group size at which the average degree is around 1, meaning that each agent has in average one other agent to interact with. Below this value, interactions are sporadic and determined by the agent mobility, while above this value interactions are frequent as small clusters of agents appear.
2) $N_c = N_{\langle k \rangle \simeq 4.51}$ corresponds to the critical group size for a percolation transition [23]. Above N_c, the network is characterised by a giant component of size N.

Given the broadcasting rule employed for the NG in this paper, it is clear that the characteristics of the interaction network are fundamental. If there exists a giant component, information can spread quickly. If otherwise robots are mostly isolated, they will not be able to interact and convergence would be slower, as discussed in the following.

A. Influence of density, mobility and interaction frequency

To determine the consensus dynamics and the effects of the different parameters of the system, we run multiagent simulations with small and large groups in a squared arena of size $L = 1\,\mathrm{m}$. In this condition, we have $N_1 \simeq 32$ and $N_c \simeq 143$. Figure 2 reports the consensus time t_c for different parameterisations varying $N \in [10, 500]$ and $\tau_m, \tau_s \in [10, 50]s$ (see also the dataset provided at http://ieeexplore.ieee.org).

Looking at the results, we note that t_c is a decreasing function of N. Indeed, a higher density corresponds to a higher number of concurrently executed games, and results in a faster convergence. For $N > N_c$ and small τ_s, the consensus time collapses to the same value for varying τ_m (see for instance the top-left panel in Fig. 2). Above the percolation threshold, agent mobility plays a minor role and the consensus dynamics can be related to the characteristics of the static network of interactions. Especially for low values of τ_m, the dynamics closely correspond to those of static agents interacting on a random geometric network [19]. Here, the agreement process proceeds through the formation of clusters of agents with local consensus separated by an interface of "undecided" agents, and consensus dynamics recall the coarsening on regular lattices [12]. This is confirmed by the left panel in Fig. 3, which shows how the convergence time t_c scales with $\tau_s \in [1, 500]s$ for $N = 300$. It is possible to appreciate a kind of power-law scaling $t_c \simeq \tau_s^\gamma$, with $\gamma \simeq 0.5$. This indicates that the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2016.2519537, IEEE Robotics and Automation Letters

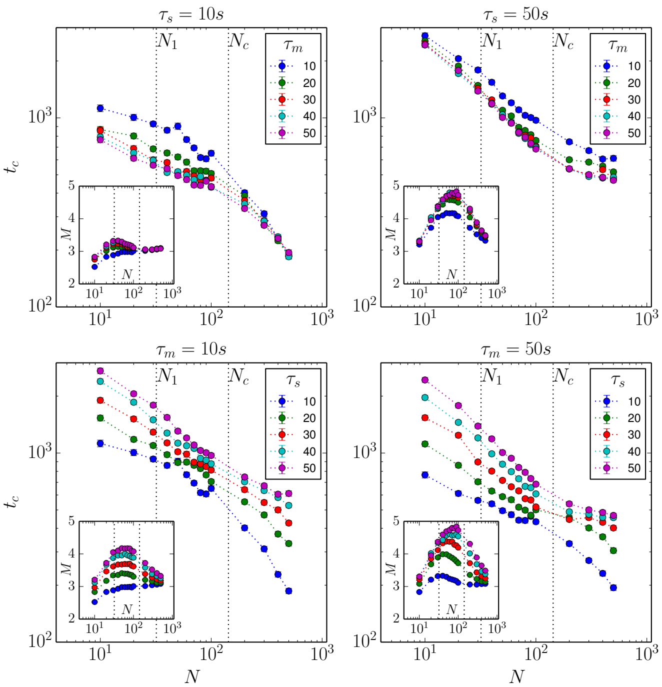

Fig. 2. Results from multiagent simulations. Each panel shows the dependence of the convergence time t_c on the system size N for different parameterisation. Statistical error bars are not visible on the scale of the graphs. Vertical dotted lines indicate the thresholds for $N_1 = 32$ and $N_c = 143$. The insets show the memory requirements M plotted against the system size N for the same parameterisations.

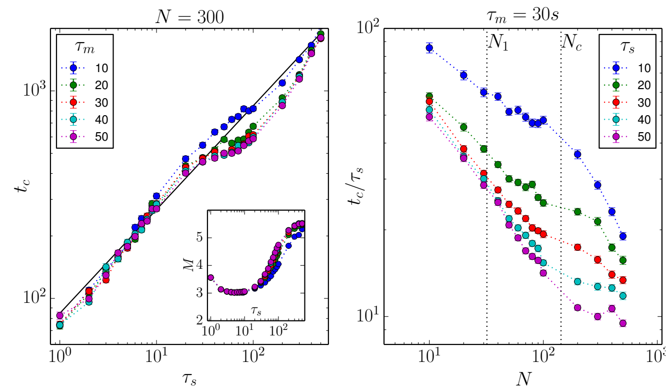

Fig. 3. (Left) Scaling of convergence time as a funciton of τ_s. The black solid line $t_c \simeq \tau_s^{0.5}$ serves as a guide for the eye to appreciate the power law scaling of the convergence time t_c. (Right) Rescaling convergence time by τ_s, representing the average number of broadcasts before convergence. Statistical error bars are not visible on the scale of the graphs.

convergence dynamics are mostly determined by τ_s, while τ_m plays a relatively minor role, hence confirming the above mentioned resemblance with coarsening on lattices or random geometric networks. Similarly to fully-connected networks [10], log-periodic oscillations are visible in the power law scaling, so that for some values of τ_s, mobility happens to be more relevant, with large τ_m determining a lower convergence time (see also Fig. 2 top-right).

Below the percolation threshold N_c, agents form temporary clusters that dissolve due to the agent mobility. If the density is still high enough to ensure frequent interactions ($N > N_1$), the dynamics are determined more by the mobility of agents than by the broadcasting period τ_s. This is visible in the bottom-left and bottom-right panels of Fig. 2, where convergence times tend to coalesce for different values of τ_s and $N \in [N_1, N_c]$. Instead, for very low densities ($N < N_1$), agent-agent contacts are infrequent and last for short periods of time, so that many broadcasts go unnoticed. In this condition, high mobility is important as much as short broadcasting periods to ensure faster convergence (see Fig. 2).

To evaluate the effects of the broadcasting period more thoroughly, it is useful to look at the rescaled time t_c/τ_s indicating the average number of broadcasts each agent transmitted (see Fig. 3 right). We note lower values of the rescaled agreement time for larger values of τ_s, meaning that the number of broadcasts required for convergence diminishes for longer broadcasting periods, recalling the slower-is-faster effect observed in many complex systems [24].

A look at the memory requirements reveals that M is generally constrained to low values, which makes the NG implementation affordable for physical systems (see the insets in Fig. 2 and also the dataset provided at http://ieeexplore.ieee.org). For $N_1 < N < N_c$, mobility plays a significant role, with larger values of τ_m corresponding to larger M. The transient formation of small clusters enhances the requirements of memory the more the agents are able to travel between clusters that agree on different words. Similarly, if we look at the bottom panels, we notice that higher values of τ_s determine higher values for M. Here, slow convergence leads to agents diffusing in the arena and being exposed to multiple options, hence increasing the memory requirements. For $N > N_c$, instead, mobility is less important and the memory requirements are bound to the interaction period. The scaling analysis presented in the left panel of Fig. 3 shows that the memory requirements increase drastically with τ_s, confirming that slower convergence implies also larger memory requirements. On the other hand, frequent interactions lead to the quick formation of few clusters, so that the individual memory requirements are limited to few words, especially for those agents at the interface between clusters. As τ_s decreases further, the effect of concurrent executions of games starts to be visible with an increase in the memory requirements as a result of the higher probability of agents to simultaneously exchange different words.

B. Influence of physical interferences

The consensus dynamics described above refer to an ideal system that neglects the physical embodiment of robots. Embodiment leads to collisions with walls and among robots that constrain mobility. We study the influence of embodiment by comparing multiagent with robotics simulations performed in similar conditions to what described above (see Fig. 4 for selected parameterisations, and also the complete dataset provided at http://ieeexplore.ieee.org). We first note that the convergence time t_c is in general higher for robotic simulations, as a consequence of the slower diffusion in space resulting from the turning time, which introduces a stochastic delay in

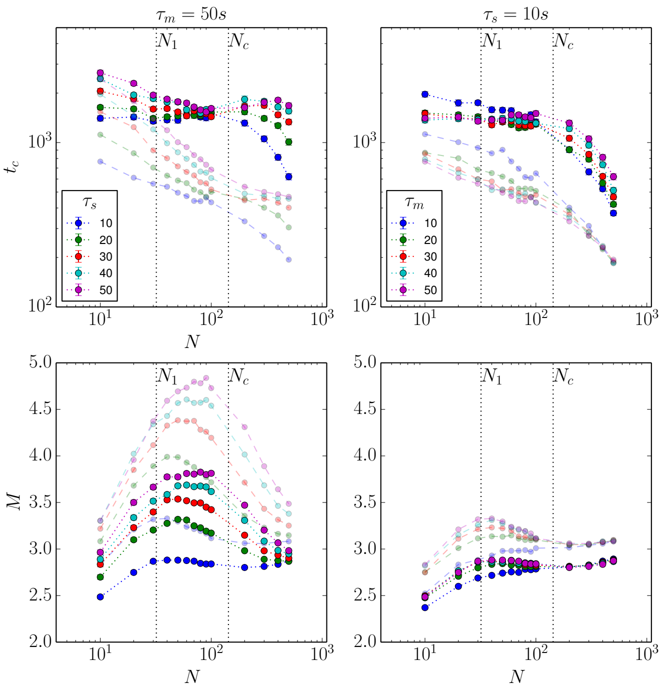

Fig. 4. Comparison between multiagent and robot simulations. Average values from multiagent simulations are shown with transparent colours, and serve as reference to appreciate the results from robotic simulations. Statistical error bars are not visible on the scale of the graphs. (Top) Convergence time t_c. (Bottom) Required memory M. (Left) Results for $\tau_m = 50$ s, and varying τ_s. (Right) Results for $\tau_s = 10$ s, and varying τ_m.

the random walk pattern, and due to the physical boundaries that prevent robots to freely move. For instance, in the top-left panel of Fig. 4 we show the case for $\tau_m = 50$ s: here, collisions lead to an approximately constant t_c for $N_1 < N < N_c$, no matter what is the broadcasting time τ_s. Indeed, the slower diffusion and the formation of small clusters determine the convergence time more than the interaction frequency. Collisions with walls and with other robots lead to the formation of stable clusters in which consensus can be quickly achieved. Such clusters dissolve at a slower pace for larger values of τ_m, due to robots turning away less often. Hence, the effects of mobility are diluted especially when it is supposed to play an important role, i.e., when $N < N_c$ (see the complete dataset provided at http://ieeexplore.ieee.org). Collisions influence the convergence dynamics also for $N > N_c$, although to a lesser extent, as can be seen in the top-right panel in Fig. 4: for large τ_m, convergence is slower due to the formation of clusters that do not interact frequently, as collisions prevent robots to mix as much as in the ideal multiagent case.

The low ability to mix due to collisions has an effect also on the required memory M, which is in general lower for robot simulations (see bottom panels of Fig. 4). The slower diffusion of robots in space and the existence of boundaries limit the spreading of different words into the robot network, hence resulting in lower memory requirements.

C. Experiments with real robots

To validate our results with respect to the real robotic platform, we performed comparative experiments in a smaller arena ($L = 45$ cm). In this condition, we have $N_1 \simeq 6$ and $N_c \simeq 29$, which led us to use smaller groups of robots ($N \in \{5, 20, 35\}$) to explore the system behaviour as the characteristics of the static interaction network vary. Given the smaller dimensions, we also explored smaller values for the latencies τ_s and τ_m, and we decided to set both to the same value $\tau_a \in \{2.5, 5, 7.5\}$ s. We have performed 20 runs with real robots for each of the 9 experimental conditions (3 group sizes $\times$ 3 latencies), for a total of 180 runs. Figure 5 shows a sequence of frames from one run performed with 20 kilobots. It is possible to note that initially multiple small clusters are present in which robots have the same word (here represented by the color of the onboard LED). As time goes by, clusters disappear and eventually one single word is chosen. A supplementary video—available at http://ieeexplore.ieee.org—reports our implementation and shows several runs of the NG.

For each experimental run, we have recorded the convergence time t_c, and the obtained results have been contrasted with simulations in comparable conditions—same arena dimensions L and same latencies as determined by τ_a—with multiagent and robotic simulations. Figure 6 shows that the statistics are aligned between kilobot simulations and real kilobots, and both present a slightly larger convergence time with respect to multiagent simulations. We also note that in general, the convergence time for $\tau_a = 2.5$ s is lower in case of real robots than in simulation, while this is not always true for larger latencies. This is an effect of interferences in communication due to simultaneous broadcasts, which leads to the loss of some communication messages. When messages get lost, the convergence dynamics are actually faster because the exchanging of different words by robots broadcasting at the same time gets reduced. This is in line with the observations made in Section III about the influence of the broadcasting period, indicating that convergence is faster when there are less broadcasts. With kilobots, a reduction of the number of broadcasts due to interference results from small τ_s and large N (see Fig. 6).

IV. Conclusions

This study finds itself at the interface between theoretical investigations and robotics implementation. The results observed in the multiagent simulations can be of interest for complex systems studies as they highlight the effects of concurrency and of different latencies in the motion and interaction patterns, as determined by implementation constraints. Concurrency is customary in multi-robot systems and artificial decentralised systems in general. Hence, accounting for it into abstract models is important to provide usable predictions. We have found here that concurrent executions of games are particularly important for aspects like the maximum memory M, and future studies should better characterise such effects in terms of the probability of observing concurrent executions at any time.

Robotics simulations and experiments with kilobots showed how embodiment influences the consensus dynamics by limiting the diffusion of information into the system: on the one hand, collisions lead to the formation of clusters that dissolve

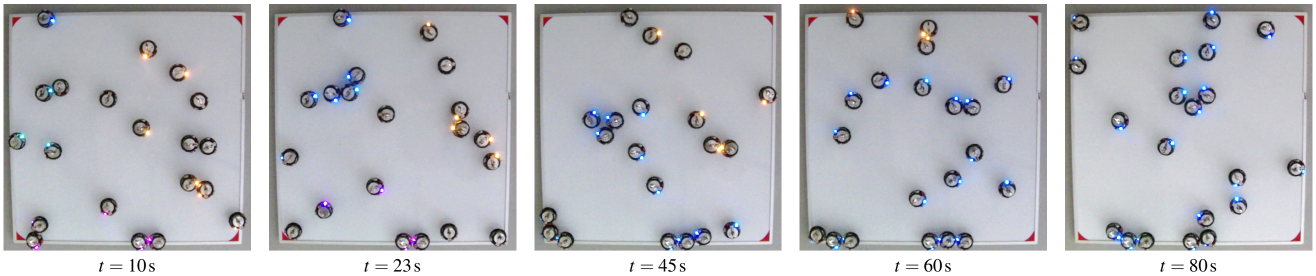

Fig. 5. Different shots of an experiment with 20 kilobots, with $\tau_s = \tau_m = 2.5\,\mathrm{s}$ (see also the supplementary video at http://ieeexplore.ieee.org). Robots lighting their LED have only one word in their inventory, while no color signal indicates more than one word or an empty inventory. Different words correspond to different colours.

slower for larger τ_m, leading to slower convergence times. On the other hand, the memory requirements of robots is reduced as only few robots at the interface between clusters experience more than two words at the same time. Future studies should attempt a more precise description of the diffusive motion of agents under physical constraints, in order to obtain better predictions in terms of the expected interaction network. Additionally, the communication protocol employed by kilobots and the observed interferences need to be better characterised. Simulations should account for uncertain reception of messages, as well as for the exponential back-off used during transmission when the channel is busy. By including such features, we expect to deliver precise estimations of the system behaviour even for very large group sizes and short broadcasting periods.

REFERENCES

[1] C. A. C. Parker and H. Zhang, "Cooperative decision-making in decentralized multiple-robot systems: the best-of-n problem," *IEEE Transactions on Mechatronics*, vol. 14, no. 2, pp. 240–251, 2009.

[2] G. Sartoretti, M.-O. Hongler, M. E. de Oliveira, and F. Mondada, "Decentralized self-selection of swarm trajectories: from dynamical systems theory to robotic implementation," *Swarm Intelligence*, vol. 8, no. 4, pp. 329–351, 2014.

[3] A. Reina, G. Valentini, C. Fernández-Oto, M. Dorigo, and V. Trianni, "A design pattern for decentralised decision making," *PLoS ONE*, vol. 10, no. 10, pp. e0 140 950–18, 2015.

[4] T. D. Seeley, P. K. Visscher, T. Schlegel, P. M. Hogan, N. R. Franks, and J. A. R. Marshall, "Stop signals provide cross inhibition in collective decision-making by honeybee swarms," *Science*, vol. 335, no. 6064, pp. 108–11, 2012.

[5] C. R. Reid, S. Garnier, M. Beekman, and T. Latty, "Information integration and multiattribute decision making in non-neuronal organisms," *Animal Behaviour*, vol. 100, no. C, pp. 44–50, 2015.

[6] V. Sood and S. Redner, "Voter model on heterogeneous graphs," *Physical Review Letters*, vol. 94, no. 17, pp. 178 701–4, 2005.

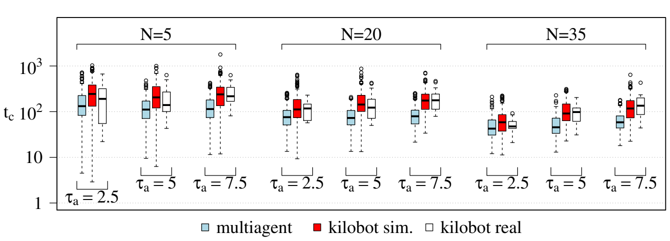

Fig. 6. Comparison between multiagent simulations, kilobot simulations and real kilobots. For each condition, 200 runs were performed in simulation, while 20 runs were performed with physical robots. Boxes represent the interquartile range, horizontal lines mark the median, whiskers extend to 1.5 times the first quartiles, and dots represent outliers.

[7] C. Castellano, S. Fortunato, and V. Loreto, "Statistical physics of social dynamics," *Reviews of Modern Physics*, vol. 81, no. 2, pp. 591–646, 2009.

[8] D. Brockmann and D. Helbing, "The hidden geometry of complex, network-driven contagion phenomena," *Science*, vol. 342, no. 6164, pp. 1337–1342, 2013.

[9] L. Steels, "A self-organizing spatial vocabulary," *Artificial life*, vol. 2, no. 3, pp. 319–332, 1995.

[10] A. Baronchelli, M. Felici, V. Loreto, E. Caglioti, and L. Steels, "Sharp transition towards shared vocabularies in multi-agent systems," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2006, no. 06, p. P06014, 2006.

[11] D. Centola and A. Baronchelli, "The spontaneous emergence of conventions: An experimental study of cultural evolution," *Proceedings of the National Academy of Sciences*, vol. 112, no. 7, pp. 1989–1994, 2015.

[12] A. Baronchelli, L. Dall'Asta, A. Barrat, and V. Loreto, "Topology-induced coarsening in language games," *Physical Review E*, vol. 73, no. 1, p. 015102, 2006.

[13] A. Baronchelli and A. Díaz-Guilera, "Consensus in networks of mobile communicating agents," *Physical Review E*, vol. 85, no. 1, p. 016113, 2012.

[14] J. Werfel, K. Petersen, and R. Nagpal, "Designing Collective Behavior in a Termite-Inspired Robot Construction Team," *Science*, vol. 343, no. 6172, pp. 754–758, 2014.

[15] T. Soleymani, V. Trianni, M. Bonani, F. Mondada, and M. Dorigo, "Bio-inspired construction with mobile robots and compliant pockets," *Robotics and Autonomous Systems*, vol. 74, pp. 340–350, 2015.

[16] R. O'Grady, A. L. Christensen, and M. Dorigo, "SWARMORPH: Multirobot morphogenesis using directional self-assembly," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 738–743, 2009.

[17] M. Rubenstein, A. Cornejo, and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm," *Science*, vol. 345, no. 6198, pp. 795–799, 2014.

[18] M. Rubenstein, C. Ahler, N. Hoff, A. Cabrera, and R. Nagpal, "Kilobot: A low cost robot with scalable operations designed for collective behaviors," *Robotics and Autonomous Systems*, vol. 62, no. 7, pp. 966–975, 2014.

[19] Q. Lu, G. Korniss, and B. K. Szymanski, "Naming games in two-dimensional and small-world-connected random geometric networks," *Physical Review E*, vol. 77, no. 1, p. 016111, 2008.

[20] A. Baronchelli, "Role of feedback and broadcasting in the naming game," *Physical Review E*, vol. 83, no. 4, p. 046103, 2011.

[21] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. A. Di Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, and M. Dorigo, "ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems," *Swarm Intelligence*, vol. 6, no. 4, pp. 271–295, 2012.

[22] SSR Lab, Harvard University, Cambridge, MA, "Kilobotics. all the resources needed to get started with the kilobots," https://www.kilobotics.com, 2013, [Online; accessed 12 January 2016].

[23] N. Fujiwara, J. Kurths, and A. Díaz-Guilera, "Synchronization in networks of mobile oscillators," *Physical Review E*, vol. 83, no. 2, p. 025101, 2011.

[24] C. Gershenson and D. Helbing, "When slower is faster," *Complexity*, vol. 21, no. 2, pp. 9–15, 2015.