Impact of the Update Time on the Aggregation of Robotic Swarms through Informed Robots

Antoine Sion[1], Andreagiovanni Reina[2], Mauro Birattari[2], and Elio Tuci[1]

[1] Faculty of Computer Science, University of Namur, Belgium
{antoine.sion,elio.tuci}@unamur.be
[2] IRIDIA, Université Libre de Bruxelles, Belgium
{andreagiovanni.reina,mauro.birattari}@ulb.be

Abstract. Self-organised aggregation is one of the basic collective behaviours studied in swarm robotics. In this paper, we investigate an aggregation problem occurring on two different sites. Previous studies have shown that a minority of robots, informed about the site on which they have to aggregate, can control the final distribution of the entire robot swarm on the sites. We reproduce this strategy by adapting the previous probabilistic finite-state machine to a new simulated robotic platform: the Kilobot. Our simulation results highlight that the update time (i.e., the amount of time a robot waits before making a decision on leaving a site) impacts the dynamics of the aggregation process. Namely, a longer update time lowers the number of robots wandering in the arena, but can slow down the dynamics when the target final distribution is far from the one initially formed. To ensure a low number of wandering robots while maintaining a quick convergence towards the target final distribution of the swarm, we introduce the concept of a dynamic update time increasing during the aggregation process.

1 Introduction

Swarm robotics studies the coordination of decentralised robot swarms displaying self-organised collective behaviours emerging from local interactions between the agents. The goal is to design swarms that are robust to the loss of robots, scalable in size, and flexible to different environments and tasks [2]. Several basic collective behaviours have been identified and studied in robotic swarms such as collective motion, task allocation, pattern formation, or consensus [16, 25]. In this paper, we study an aggregation behaviour. This collective behaviour is often a prerequisite used to group a part or the totality of the swarm physically before starting another behaviour requiring the proximity of the robots [27]. Two types of control systems have been used to achieve aggregation in swarms of robots: probabilistic finite-state machines and artificial neural networks. The first one is inspired by biology studies that investigate the aggregation behaviours of animals like cockroaches [17] or bees [28] and have been adapted to robotic systems [14, 27]. The second one employs neural networks as controllers to generate aggregation behaviours through training in simulation [7, 13].

The controller used in this paper is a probabilistic finite-state machine composed of simple states such as leaving a site or exploring the environment (see Figure 1a). We study an aggregation problem on two distinct sites where a swarm of robots is equipped with the controller of Figure 1a. A minority of robots in the swarm are informed about which of the two sites they have to aggregate on. Black-informed robots selectively avoid to aggregate on the white site and only aggregate on the black site, and white-informed robots avoid the black site and only aggregate on the white site. By adjusting the ratio of informed robots choosing to aggregate exclusively on the black or the white site, the designer can bias the entire swarm to reach a target distribution (e.g., 70% of robots on the black site and 30% on the white site). This method used to steer the group dynamic of the aggregation process is inspired by collective behaviours observed in biology where a minority of individuals aware of pertinent environmental information dictate the group behaviour [5, 3]. The concept of informed robots has already been applied to a number of diverse collective behaviours in swarm robotics such as flocking [4, 8, 9], collective decision making [22, 6] or, as in our study, self-organised aggregation [11, 12, 10, 15].

Firat et al. [10, 11] originally introduced informed robots in an aggregation task on two sites in order for the entire swarm to aggregate on only one of them. Informed robots in this study were programmed to aggregate on only one of the two sites. This was extended by another study [12] allowing informed robots to be divided in two groups reflecting the target final distribution on the two sites. In parallel, an analytical model for this problem was developed by Gillet et al. [15], showing that the quantity of informed robots needed to have an effect on the final distribution was dependent of the site carrying capacity. In subsequent work [26], we simplified the controller designed by Firat et al. [12] by employing a memoryless reactive controller based on a simpler communication protocol, and we conducted a comparative study showing that the novel controller resulted in a more flexible behaviour of the swarm, together with equal or better performance compared to the original controller.

Our end goal is to implement this controller on physical robots on a different robotic platform than the one used in [26] to show that the controller is not robot-specific but generic for other mobile robots capable of minimal motion and minimal communication. Moreover, in swarm robotics, the validation of the simulated controller on the physical robots is of paramount importance due to the so-called reality gap. This gap generally translates in a performance drop when porting the control software on the physical robots due to the differences between the simulation and the real world [19]. In this paper, we adapt the controller to the new robotic platform by taking into account the different sensors, actuators and programming language while keeping the same individual behaviour found in [26]. Then, we recreate the experimental setup in simulation to validate our modifications. In this study, we also investigate the impact of the update time on the aggregation dynamics of the swarm: our results indicate a trade-off between the speed of the aggregation process and the number of robots not aggregating on any of the two sites when the system has reached a stable final distribution.

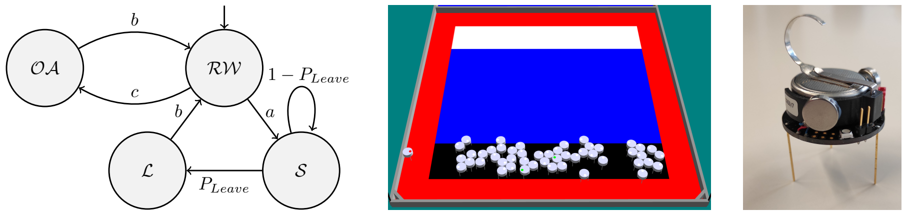
Finally, we propose a new simple method acting on the update time to avoid a slowdown of the dynamics while keeping the number of robots wandering in the arena low; fast convergence of the number of robots on the sites is indeed required to stay inside the limits of the robots' battery autonomy. Our methodology is presented in Section 2 along the robotic platform and our results in Section 3.

2 Materials and Methods

The scenario we consider investigates the aggregation process of a swarm of 50 robots inside a squared arena containing a black and a white site (see Figure 1b). We control the aggregation by introducing a bias in the swarm through the use of a minority of informed robots. Informed robots are robots that are programmed to selectively aggregate on one of the two sites only. With a swarm of size N containing N_{sb} black-informed robots only aggregating on the black site and N_{sw} white-informed robots only aggregating on the white site, we can define the proportion of informed robots in the swarm by $\rho_I = \frac{N_{sb}+N_{sw}}{N}$. In a similar manner, the proportion of black-informed robots and white-informed robots relative to the total number of informed robots are defined as $\rho_{sb} = \frac{N_{sb}}{N_{sb}+N_{sw}}$ and $\rho_{sw} = \frac{N_{sw}}{N_{sb}+N_{sw}}$, respectively. Starting from positions and orientations randomly chosen in the arena following a uniform distribution, the swarm should redistribute itself on the aggregation sites to obtain a final distribution of the robots approaching $N \times \rho_{sb}$ on the black site and $N \times \rho_{sw}$ on the white site.

The robotic platform used for the implementation of our controller is the Kilobot [24], which is a low-cost robot designed to facilitate the testing of collective behaviours requiring a large number of robots (see Figure 1c). Two vibrations motors are used for locomotion, allowing a forward speed of $1\,\mathrm{cm}/\mathrm{s}$ and a rotation speed of $45\,^{\circ}/\mathrm{s}$. The robot is also equipped with an infrared LED emitter and an infrared photodiode receiver which allow communication between robots up to 10 cm by sending light that reflects on the ground. There is also a visible light sensor on the top of the robot and a three-colour LED to visualise the state in which the robot is. These robots are widely employed also because they can easily be programmed in batches using an overhead controller that sends infrared messages.

While it is user friendly, the Kilobots are limited by the simplicity of their sensors. This restricts the types of experiments that can be performed with the robots as they can not get feedback on their physical environment (other than with the light sensor). To overcome this problem, solutions have been proposed such as the ARK system [23] which allows the use of virtual sensors through an overhead controller and a camera tracking system. In this paper, we equip the Kilobots with virtual sensors using the Kilogrid [29]; an electronic table made of a grid of infrared modules allowing communication to and from the Kilobots. Each module (10 cm × 10 cm) is divided equally into four cells, each capable of communicating with the Kilobots above them. Each cell is also equipped with two RGB LEDs that can be colored to inform the user. The Kilogrid monitors continuously the position and state of each Kilobot, and allows the implementa-



(a) (b) (c)

Fig. 1. (a) The probabilistic finite-state machine controller composed of four states: Random Walk ($\mathcal{RW}$), Stay ($\mathcal{S}$), Leave ($\mathcal{L}$) and Obstacle Avoidance ($\mathcal{OA}$). Letters in lowercase represent events triggering the transition between states: a is finding a suitable site for aggregating, b is getting out of a site or the obstacle area, and c is entering the obstacle area. P_{Leave} is the probability to leave the site as described in Eq. (2). (b) The simulated Kilogrid with a swarm of 50 Kilobots, an aggregation site in white, an aggregation site in black, the obstacle area in red and a neutral area in blue. (c) The Kilobot robot.

tion of virtual sensors measuring different values in each cell (e.g. to measure the humidity or the temperature of a virtual environment). The Kilobots are also able to modify the environment by sending messages to the current cell that will be processed as actions.

Experiments were performed using the ARGoS simulator [21] with the ARGoS-Kilobot plugin to model the Kilobots [20], and the ARGoS-Kilogrid plugin [1] which allow the use of identical code in simulation and on the real Kilobots and Kilogrid. In our previous experimental scenario [26], the arena was a circle with two circular aggregation sites. Here, we test a different environment with a squared arena of $100 \times 100\,\mathrm{cm}^2$ (Figure 1b) allowing experimentation with swarms of 50 Kilobots. The two aggregation sites are stripes positioned symmetrically at the edges of the arena. One stripe stands for the white site and the other one for the black site. Each stripe is composed of 16×3 Kilogrid cells, corresponding to $80 \times 15\,\mathrm{cm}^2$. We added a red zone with a width of 2 Kilogrid cells (10 cm) around the perimeter of the arena which acts as an obstacle for the Kilobots. Once in the red area, the robots enter a state of obstacle avoidance until they get back inside the arena. The use of the Kilogrid enhances the capabilities of the Kilobots with virtual ground sensors: each cell broadcasts continuously its colour which is received by the robots present at the location.

We reproduced the controller from [26] which is a probabilistic finite-state machine composed of three states and we added a state of obstacle avoidance for the walls of the arena (Figure 1a). The robot can be in the state: ($\mathcal{RW}$) random walk during which it explores the environment, ($\mathcal{S}$) stay during which it aggregates on a site, ($\mathcal{L}$) leave during which it gets out of a site, and ($\mathcal{OA}$) obstacle avoidance during which it gets out of the obstacle area. Different from [26], in our experimental setup, the Kilobots are not equipped with physical sensors capable of detecting the position of other close robots or obstacles. Therefore, the controller used in this paper does not implement a collision avoidance scheme between the robots. However, to keep the robots from getting stuck on the walls of the arena, we added the $\mathcal{OA}$ state which triggers as soon as the robots enter the red area surrounding the environment and allows the robot to get out of it. This differs from [26] where robots can directly detect close obstacles with their physical sensors and avoid them while in any state of the probabilistic finite-state machine.

The initial state is random walk ($\mathcal{RW}$); in this state, the robot explores the environment following an isotropic random walk with turning angles obtained from a wrapped Cauchy distribution [18] with the following density function:

$$f_\omega(\theta, \mu, \rho) = \frac{1}{2\pi} \frac{1 - \rho^2}{1 + \rho^2 - 2\rho\cos(\theta - \mu)}, 0 < \rho < 1, \qquad (1)$$

with μ the average value of the distribution and ρ the skewness. At $\rho = 0$, the distribution becomes uniform and the turning angles of the robot are not correlated. At $\rho = 1$, the distribution becomes a Dirac distribution and the robot's path is a straight line. In this paper, we use $\rho = 0.5$. We define a fixed step length by taking a step duration of 10 s (the Kilobot average speed is 1 cm/s). If the robot enters a valid aggregation site (any site if it is non-informed, the white site if it is white-informed or the black site if it is black-informed), it transitions from state $\mathcal{RW}$ to state $\mathcal{S}$. During this transition, it continues to move straight during 10 s to avoid aggregating on the border of the site.

While in state $\mathcal{S}$, the robot stops moving and signals its presence to other robots by broadcasting infrared messages containing its unique ID. At the same time, it receives messages from other robots resting in the nearby neighbourhood (the maximum communication range of the Kilobot is 10 cm). Each time the robot receives a message with a new ID, it stores this ID in an array. Periodically, every update period of length T_{update}, it estimates n, the local number of robots, by counting the total number of unique messages received. This number is then used to compute the probability to leave the site:

$$P_{Leave} = \begin{cases} \alpha e^{-\beta n} & \text{for non-informed robots} \\ 0 & \text{for informed robots} \end{cases} \qquad (2)$$

where $\alpha = 0.5$ and $\beta = 2.25$. The higher is the number of robots in the local neighbourhood, the higher is the probability to stay on the site. Informed robots do not leave their site after finding it. After sampling the probability, the robot stays in state $\mathcal{S}$ or transitions to state $\mathcal{L}$.

In state $\mathcal{L}$, the robot tries to leave the aggregation site by moving forward. Once out, it transitions to state $\mathcal{RW}$. In some cases where the robot is encircled by a cluster of other robots, it will not be able to get out of the site. To avoid a situation where the robot tries to go forward indefinitely but cannot get out of the cluster, we implemented a timer limiting the time spent in state $\mathcal{L}$. If after 30 s, the robot is not outside the site, it transitions back to state $\mathcal{S}$.

The robot can also transition from state $\mathcal{RW}$ to state $\mathcal{OA}$ by entering the obstacle area near the surrounding walls. In state $\mathcal{OA}$, the robot first turns left for 4 s, then moves straight for 10 s. Once the robot is out of the obstacle area, it transitions to state $\mathcal{RW}$. If not, it repeats the operation. Our simulation experiments show that this simple method is sufficient to allow the Kilobots to exit the obstacle area.

3 Results

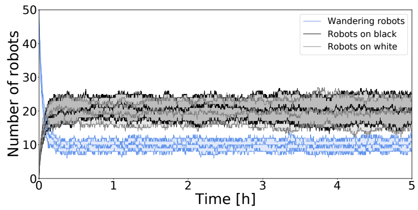
This section discusses the impact of the update time on the dynamics of the swarm for all configurations of the parameters found in Table 1. We run 50 trials for each combination of these parameters. The swarm size is set to $N = 50$ robots and the proportion of informed robots in the swarm to $\rho_I = 0.3$. We vary the update time $T_{update} = \{2\,\text{s}, 8\,\text{s}\}$ for different proportions of black/white informed robots: $\rho_{sb} = \{0.5, 0.7, 1\}$, and $\rho_{sw} = 1 - \rho_{sb}$. The robots are randomly placed at the start of the simulation in the neutral area (the blue area in Figure 1b) following a uniform distribution. The time length of the simulation experiments is set to 5 h to capture the slow convergence of the system occurring in some cases, even if the battery autonomy of the real robots is limited to 2.5 h. After discussing the obtained results, we also propose a simple method to drive the convergence of the swarm towards the target final distribution in a shorter time while limiting the number of wandering robots in the arena.

Table 1. Parameters values

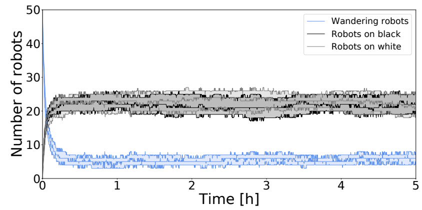
Experiment parameters	Values
Swarm size (N)	$\{50\}$
Proportion of informed robots (ρ_I)	$\{0.3\}$
Proportion of black informed robots (ρ_{sb})	$\{0.5, 0.7, 1\}$
Update time (T_{update})	$\{2, 8\}$ s

Figure 2 shows the evolution of the numbers of robots aggregated on the white site and the black site as well as the number of robots wandering in the arena in function of the time. In all graphs, the solid black lines are the median and the interquartile range of the number of robots aggregated on the black site over the 50 trials. The same applies for the number of robots aggregated on the white site in grey and the number of wandering robots in blue. We set T_{update} to 2 s in Figures 2a, 2c, 2e and to 8 s in Figures 2b, 2d, 2f. The proportion of informed robots staying on the black site is set to $\rho_{sb} = 0.5$ in Figures 2a and 2b, to $\rho_{sb} = 0.7$ in Figures 2c and 2d and to $\rho_{sb} = 1$ in Figures 2e and 2f. The target final distributions of the swarm at the end of the experiment for these proportions are, respectively, 25 robots on each site; 35 robots on the black site and 15 on the white site; 50 robots on the black site.

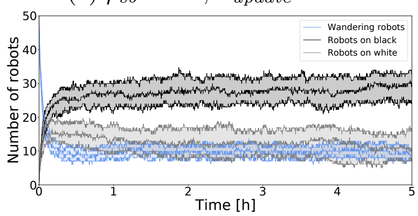
The dynamics of the system are the following. The numbers of robots aggregated on the sites increase rapidly in the first phase of the simulation. Afterwards, they vary slowly before stabilising around a certain value. In the graphs of the



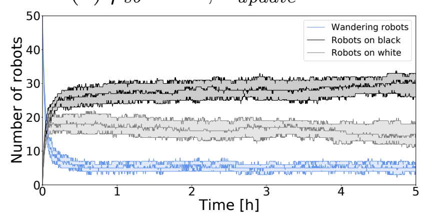
(a) $\rho_{sb} = 0.5$; $T_{update} = 2$ s



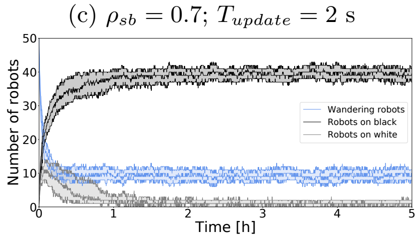
(b) $\rho_{sb} = 0.5$; $T_{update} = 8$ s



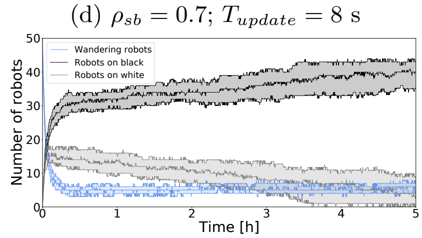
(c) $\rho_{sb} = 0.7$; $T_{update} = 2$ s



(d) $\rho_{sb} = 0.7$; $T_{update} = 8$ s



(e) $\rho_{sb} = 1$; $T_{update} = 2$ s



(f) $\rho_{sb} = 1$; $T_{update} = 8$ s

Fig. 2. Evolution over time of the number of robots over 50 trials for $\rho_I = 0.3$. $T_{update} = 2$ s on the graphs of the first column and $T_{update} = 8$ s on the graphs of the second column. $\rho_{sb} = 0.5$ in the first row, $\rho_{sb} = 0.7$ in the second row and $\rho_{sb} = 1$ in the third row. The solid grey lines represent the median and the interquartile range of the number of robots aggregated on the white site. The same applies to the black lines corresponding to the number of robots aggregated on the black site and the blue lines corresponding to the number of wandering robots.

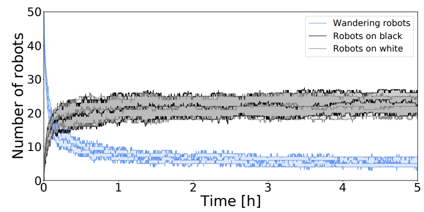
left column of Figure 2, the median of the number of robots not aggregated on any of the two sites and wandering in the environment decreases quickly (as a consequence of robots aggregating on the two sites), and then remains stable throughout the simulation around a value of 10 robots. The same occurs for the graphs of the right column of Figure 2 but the number of wandering robots stabilises around 5. This effect is due to the longer update time of 8 s which improves the total number of robots staying on the sites. A robot with a short update time has more occasions to sample the probability to leave the site in a given time frame, compared to a robot having a longer update time. Thus, the proportion of robots staying at a site that choose to leave is higher for shorter update time, resulting in a higher quantity of robots wandering the arena. This is well illustrated by comparing Figure 2a where the number of robots on the black or white site converges around 20 robots and Figure 2b where the distribution of robots stabilises around 22 robots on each site.

We can also see that, in Figure 2f, the numbers of robots staying on the sites converge more slowly towards their stabilised value compared to Figure 2e. These slower dynamics are another effect of a longer update time inducing a lower tendency to get out of the sites. This slowdown is not visible in Figures 2a, 2b, 2c and 2d for other values of ρ_{sb} and ρ_{sw}. These values are used to obtain target final distributions of robots between the two sites that are similar to the distribution initially formed at the start of the aggregation process, where the robots distribute themselves randomly between the two sites, hence in a distribution with approximately half of the swarm on the black site and the other half on the white site. Thus, the effect of a longer update time in these cases is less evident but manifests more clearly when the swarm should attain a more extreme final distribution (e.g. for $\rho_{sb} = 1$ and $\rho_{sw} = 0$ in Figure 2f).

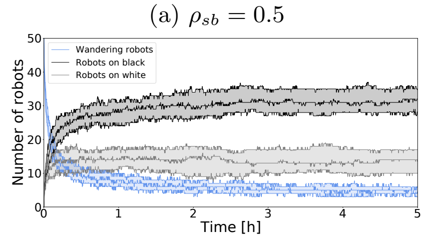
In order to obtain a relatively quick convergence to the target final distribution and to limit the number of wandering robots in the arena at a later stage, we introduce the idea of a dynamic update time. The time T_{update} is initialised at 1 s and increases linearly throughout the aggregation process by a quantity Δ_{update} every minute. Here we selected $\Delta_{update} = 0.125\,\mathrm{s}$ after testing multiple values. Our time-varying strategy ensures a short update time which makes the swarm more dynamic in the first part of the aggregation process where the distribution of robots begins its formation on the sites, and a longer update time afterwards which lowers the probability of robots getting out of the sites and makes the swarm more static. Results for the same parameters described in Table 1 are shown in Figure 3. For all graphs, the median of the number of wandering robots gradually decreases to attain a value of 5. Figure 3a shows a convergence of the median of the number of robots on both sites around a value of 22. In Figure 3b, the median of the number of robots aggregated on the black site stabilises around 31 and the one for the white site around 14. In Figure 3c, the median of the number of robots aggregated on the black site converges to 45 and the median of the number of robots aggregated on the white one to 0.

These results show that the performance of the swarm at attaining the target final distributions is improved by the use of our strategy of dynamic update time. In all studied cases, the number of wandering robots remain close to 5, as it was the case for a update time of 8 s. In addition, the use of a dynamic update time cancels the slow dynamics observed with a longer update time when the target final distribution was far from the one initially formed in the first phase of the aggregation process. The numbers of robots aggregated on the two sites stay stable after 2 h which will allow us to test our controller on the physical robots with an experiment time that do not exceed the Kilobot's battery autonomy.

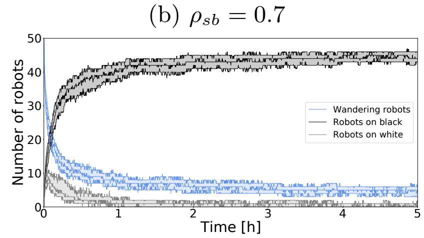
In all the graphs, there is a small number of robots that do not aggregate on the sites. Visual inspection of the simulations show that this could be due to the nature of the robots used and the shape of the arena. No collision avoidance scheme is implemented on the Kilobots. When the aggregation has occurred and the numbers of robots aggregated on the sites are high, there is a higher



(a) $\rho_{sb} = 0.5$



(b) $\rho_{sb} = 0.7$



(c) $\rho_{sb} = 1$

Fig. 3. Evolution over time of the number of robots over 50 trials for $\rho_I = 0.3$, and different proportions of informed black robots: (a) $\rho_{sb} = 0.5$, (b) $\rho_{sb} = 0.7$ and (c) $\rho_{sb} = 1$ with a linear increase of the update time. The solid grey lines represent the median and the interquartile range of the number of robots aggregated on the white site. The same applies to the black lines corresponding to the number of robots aggregated on the black site and the blue lines corresponding to the number of wandering robots.

probability for the remaining robots trying to enter a site to push other robots out of the site while entering. Furthermore, robots choosing to leave an already heavily populated site also have a higher probability to push other robots out of the site while getting out of it. This could explain the constant remaining number of wandering robots at the end of the experiment: even if new robots manage to aggregate on the sites, a small amount of aggregated robots are constantly being pushed out of the sites. Another cause could also be that the entrance of the few remaining robots into the already populated sites is hindered by the physical barrier formed by clusters of robots near the borders of the sites. Nevertheless, while we recognise that a small number of wandering robots always subsist at the end of the aggregation process, the influence of the informed robots on the dynamics is clearly visible as well as the impact of the update time.

4 Conclusions

We successfully reproduced the controller defined in [26] in simulation and adapted it to a second robotic platform, the Kilobot [24]. Namely, we introduced an obstacle avoidance state to avoid the edges of the arena and we removed the collision avoidance scheme between the robots. Through simulations, we conducted an empirical study showing the impact of the update time on the dynamics of the aggregation of a swarm of robots on two distinct sites. This was realised for three different ratios of black/white informed robots in the swarm targeting three final distributions of the swarm on the sites. Our results show that a longer update time reduces the number of robots that do not aggregate on any of the two sites and wander in the arena. However, this also induces a slower convergence towards the target final distributions when the swarm needs to attain a distribution that is far from the one initially formed in the first phase of the aggregation process. To solve this problem, we introduced a dynamic update time linearly increasing during the experiment. This resulted in a quick convergence towards the target final distributions and an overall low number of wandering robots. Future work will consist of implementing our controller on the physical robots in order to validate our results and evaluate its ability to cross the reality gap.

Acknowledgements. This work was supported by Service Public de Wallonie Recherche under grant n° 2010235 - ARIAC by DIGITALWALLONIA4.AI; by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 681872); and by Belgium's Wallonia-Brussels Federation through the ARC Advanced Project GbO (Guaranteed by Optimization). AR and MB acknowledge the financial support from the Belgian F.R.S.-FNRS, of which they are Chargé de Recherches and Directeur de Recherches, respectively.

References

1. Aust, T., Talamali, M.S., Dorigo, M., Hamann, H., Reina, A.: The hidden benefits of limited communication and slow sensing in collective monitoring of dynamic environments. In: under review in Swarm Intelligence (ANTS 2022). LNCS, Springer, Cham (2022)
2. Brambilla, M., Ferrante, E., Birattari, M., Dorigo, M.: Swarm robotics: a review from the swarm engineering perspective. Swarm Intelligence **7**(1), 1–41 (2013)
3. Calvo Martín, M., Eeckhout, M., Deneubourg, J.L., Nicolis, S.C.: Consensus driven by a minority in heterogenous groups of the cockroach periplaneta americana. iScience **24**(7) (2021)
4. Çelikkanat, H., Şahin, E.: Steering self-organized robot flocks through externally guided individuals. Neural Computing and Applications **19**(6), 849–865 (2010)
5. Couzin, I.D., Krause, J., Franks, N.R., Levin, S.A.: Effective leadership and decision-making in animal groups on the move. Nature **433**, 513–516 (2005)

6. De Masi, G., Prasetyo, J., Zakir, R., Mankovskii, N., Ferrante, E., Tuci, E.: Robot swarm democracy: the importance of informed individuals against zealots. Swarm Intelligence **15**(4), 315–338 (2021)
7. Dorigo, M., Trianni, V., Şahin, E., Groß, R., Labella, T.H., Baldassarre, G., Nolfi, S., Deneubourg, J.L., Mondada, F., Floreano, D., Gambardella, L.M.: Evolving self-organizing behaviors for a swarm-bot. Autonomous Robots **17**(2), 223–245 (2004)
8. Ferrante, E., Turgut, A.E., Huepe, C., Stranieri, A., Pinciroli, C., Dorigo, M.: Self-organized flocking with a mobile robot swarm: a novel motion control method. Adaptive Behavior **20**(6), 460–477 (2012)
9. Ferrante, E., Turgut, A.E., Stranieri, A., Pinciroli, C., Birattari, M., Dorigo, M.: A self-adaptive communication strategy for flocking in stationary and non-stationary environments. Natural Computing **13**(2), 225–245 (2014)
10. Firat, Z., Ferrante, E., Cambier, N., Tuci, E.: Self-organised aggregation in swarms of robots with informed robots. In: Theory and Practice of Natural Computing. pp. 49–60. Springer (2018)
11. Firat, Z., Ferrante, E., Gillet, Y., Tuci, E.: On self-organised aggregation dynamics in swarms of robots with informed robots. Neural Computing and Applications **32**(17), 13825–13841 (2020)
12. Firat, Z., Ferrante, E., Zakir, R., Prasetyo, J., Tuci, E.: Group-size regulation in self-organized aggregation in robot swarms. In: Swarm Intelligence. pp. 315–323. Springer International Publishing, Cham (2020)
13. Francesca, G., Brambilla, M., Trianni, V., Dorigo, M., Birattari, M.: Analysing an evolved robotic behaviour using a biological model of collegial decision making. In: From Animals to Animats 12: 12th International Conference on Simulation of Adaptive Behavior, SAB 2012. Lecture Notes in Computer Science, vol. 7426, pp. 381–390. Springer, Berlin, Germany (2012)
14. Garnier, S., Gautrais, J., Asadpour, M., Jost, C., Theraulaz, G.: Self-organized aggregation triggers collective decision making in a group of cockroach-like robots. Adaptive Behavior **17**(2), 109–133 (2009)
15. Gillet, Y., Ferrante, E., Firat, Z., Tuci, E.: Guiding aggregation dynamics in a swarm of agents via informed individuals: an analytical study. In: The 2019 Conference on Artificial Life: A Hybrid of the European Conference on Artificial Life (ECAL) and the International Conference on the Synthesis and Simulation of Living Systems (ALIFE). pp. 590–597. MIT Press (2019)
16. Hamann, H.: Swarm Robotics: A Formal Approach (2018)
17. Jeanson, R., Rivault, C., Deneubourg, J.L., Blanco, S., Fournier, R., Jost, C., Theraulaz, G.: Self-organized aggregation in cockroaches. Animal Behaviour **69**(1), 169–180 (2005)
18. Kato, S., Jones, M.: An extended family of circular distributions related to wrapped cauchy distributions via brownian motion. Bernoulli **19**(1), 154–171 (2013)
19. Ligot, A., Birattari, M.: Simulation-only experiments to mimic the effects of the reality gap in the automatic design of robot swarms. Swarm Intelligence **14**(1), 1–24 (2020)
20. Pinciroli, C., Talamali, M.S., Reina, A., Marshall, J.A.R., Trianni, V.: Simulating Kilobots within ARGoS: models and experimental validation. In: Swarm Intelligence (ANTS 2018), LNCS, vol. 11172, pp. 176–187. Springer, Cham (2018)
21. Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Di Caro, G., Ducatelle, F., Birattari, M., Gambardella, L.M., Dorigo, M.: ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. Swarm Intelligence **6**(4), 271–295 (2012)

22. Prasetyo, J., De Masi, G., Ferrante, E.: Collective decision making in dynamic environments. Swarm Intelligence **13**(3), 217–243 (2019)
23. Reina, A., Cope, A.J., Nikolaidis, E., Marshall, J.A.R., Sabo, C.: ARK: Augmented reality for Kilobots. IEEE Robotics and Automation Letters **2**(3), 1755–1761 (2017)
24. Rubenstein, M., Ahler, C., Nagpal, R.: Kilobot: A low cost scalable robot system for collective behaviors. In: 2012 IEEE International Conference on Robotics and Automation. pp. 3293–3298 (2012)
25. Schranz, M., Umlauft, M., Sende, M., Elmenreich, W.: Swarm robotic behaviors and current applications. Frontiers in Robotics and AI **7** (2020)
26. Sion, A., Reina, A., Birattari, M., Tuci, E.: Controlling robot swarm aggregation through a minority of informed robots. In: Swarm Intelligence (ANTS 2022). LNCS, vol. 13491. Springer (2022)
27. Soysal, O., Şahin, E.: Probabilistic aggregation strategies in swarm robotic systems. In: Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005. pp. 325–332 (2005)
28. Szopek, M., Schmickl, T., Thenius, R., Radspieler, G., Crailsheim, K.: Dynamics of collective decision making of honeybees in complex temperature fields. PLOS ONE **8**(10), 1–11 (2013)
29. Valentini, G., Antoun, A., Trabattoni, M., Wiandt, B., Tamura, Y., Hocquard, E., Trianni, V., Dorigo, M.: Kilogrid: a novel experimental environment for the kilobot robot. Swarm Intelligence **12**(3), 245–266 (2018)