



Hybrid Control of Swarms for Resource Selection

Marco Trabattoni¹  , Gabriele Valentini² , and Marco Dorigo¹ 

¹ IRIDIA, Université Libre de Bruxelles, Brussels, Belgium
{mtrabatt,mdorigo}@ulb.ac.be

² School of Earth and Space Exploration, Arizona State University, Tempe, AZ, USA
gvalentini@asu.edu

Abstract. The design and control of swarm robotics systems generally relies on either a fully self-organizing approach or a completely centralized one. Self-organization is leveraged to obtain systems that are scalable, flexible and fault-tolerant at the cost of reduced controllability and performance. Centralized systems, instead, are easier to design and generally perform better than self-organizing ones but come with the risks associated with a single point of failure. We investigate a hybrid approach to the control of robot swarms in which a part of the swarm acts as a control entity, estimating global information, to influence the remaining robots in the swarm and increase performance. We investigate this concept by implementing a consensus achievement system tasked with choosing the best of two resource locations. We show (i) how estimating and leveraging global information impacts the decision-making process and (ii) how the proposed hybrid approach improves performance over a fully self-organizing approach.

1 Introduction

Swarm robotics is a promising approach to the design and control of systems composed of large numbers of embodied agents [9]. Robot swarms have shown potential for solving tasks which are deemed too dangerous or too demanding for humans, such as search and rescue, de-mining, underwater surveillance or environment patrolling. Inspired by nature [3,5], robot swarms are generally designed and controlled through the principles of self-organization with the aim to obtain systems that are flexible, fault-tolerant and scalable [4,9]. Typically, robot swarms do not have a leader, do not use global information, and are highly redundant thanks to a large number of constituent robots. Robots in a swarm rely on local sensing and communication to solve the tasks they are given. Having a large number of robots acting in an unsupervised manner, however, often results in a system that is hard to control and/or to predict and whose performance can vary greatly over a same task.

Centralized control, on the opposite, relies on a control entity with access to global information and with the authority to correct the behavior of the system to reach the desired goal. In general, centralized systems are easier to manage and

predict than self-organizing ones and often achieve better performances. Centralized approaches to the control of large groups of robots rely on a central entity, for example to provide the robots with directives regarding the task to execute, the motions required, or information about the position of objects of interest in the environment [15, 17, 40]. While centralized control provides us with more manageability over the system, as well as a more stable and trusted performance, the presence of a centralized entity in charge of controlling the functioning of the whole system reduces parallelism and scalability and introduces a single point of failure.

In this paper we investigate a different approach to the control of a robot swarm that we refer to as Swarm Hybrid Control System (SHCS). SHCS combines localized elements of centralized control with self-organizing behaviors performed by the remaining elements of the system with the aim to obtain the best of both design approaches. In our approach, the control authority is not an entity external to the swarm; rather, it consists of a group of robots of the swarm which cooperate in a self-organizing way to provide services akin to those of a central authority. In this way, we are able to exploit the advantages associated to a central authority without introducing a single point of failure into the system. The control entity is thus a formation of robots, created through a self-organizing process, that exchanges information locally to obtain an estimate of the global state of the system and that uses this information to influence the future behavior of the swarm. We investigate this idea by implementing a SHCS for a problem of consensus achievement.

Consensus achievement is a common problem that robot swarms are required to solve in many different application scenarios (e.g., to choose which area to explore in a de-mining scenario or which target requires the most attention in a search and rescue situation). Also known as the best-of- n problem [38], it requires the swarm to choose the best option over a set of n available possibilities which (generally) differ in their quality and cost. The problem of consensus achievement for a robot swarm has been studied in many different application scenarios and modeled with a variety of mathematical tools (e.g., ODE [19], chemical master equations [39]). Additionally, various decision-making strategies have been proposed to address this problem, most of which take inspiration from nature [30]. We consider a binary resource-selection scenario, in which the swarm is foraging between a central location (the *nest*) and two locations (*sources*) containing resources that have the same quality but different costs in terms of time necessary to collect/extract them. That is, the cost of a resource location corresponds to the time required by a robot to collect resources from that specific location. For example, robots might be collecting minerals buried underground and the cost may represent how deep the robot needs to dig to reach the minerals. The scenario we have chosen is a binary consensus achievement problem with indirect modulation of robots opinions resulting from the different cost associated to each resource location [35], in which robots alternate between foraging from their preferred source and disseminating their preference in the nest. Before returning to forage, robots pool the opinions of their neighbors and

apply a decision rule (either the majority rule or the voter model) to decide whether or not to change their current preference.¹

A well-mixed state of robots' opinions is generally assumed to be one of the conditions necessary to address distributed decision-making problems [24]. Well-mixed systems are systems in which each robot in the swarm has the same probability to interact with any other robot in the swarm. The necessity for the robots to be well-mixed when disseminating is due to their limited interaction range which limits the information they can perceive about the opinions of other robots. Poor mixing of robots' opinions may result in the fragmentation of the system in parties with contrasting opinions and prevent the achievement of consensus. While robots, when disseminating their opinion, are usually programmed to move randomly in the environment for an amount of time sufficiently long to properly mix inside the swarm [39], random motion does not guarantee that the resulting system will be well-mixed. Moreover, increasing the amount of time that the robots spend disseminating (and thus mixing) their opinions increases the overall duration of the decision-making process as well. In our implementation, the SHCS collects information about the opinions in the swarm through local interactions, and merges them in order to obtain an estimate of the global state of the system in the form of a database of robot opinions. By giving the rest of the swarm access to this information, the SHCS tries to approximate the information that robots would have access to in a well-mixed system. We show the potential of this idea by comparing the SHCS with a fully self-organizing approach over the same task.

The remaining of this paper is organized as follows. In Sect. 2, we discuss related work. In Sect. 3, we describe the chosen decision-making scenario and the controllers of the robots for both the self-organizing approach and the SHCS one. In Sect. 4, we present the results of our experiments performed in simulation. In Sect. 5, we discuss the effect of the SHCS based on our experimental results. Finally, in Sect. 6 we draw our conclusions and discuss our future directions of research.

2 Related Work

2.1 Control of Robot Swarms

Brambilla et al. [4] reviewed the literature of swarm robotics focusing on self-organizing approaches and proposed a taxonomy summarizing different design and analysis methodologies adopted in the field. Most of these design methods are bottom-up approaches in which the controller of each single robot is iteratively refined in order to obtain a desired behavior of the swarm as a whole. Recently, different design methods have been proposed to automatically derive the robot controllers for a given task. Trianni et al. [34] use a generational evolutionary algorithm to evolve robot controllers for a clustering behavior. Francesca et al. [11] proposed AutoMoDe, an approach to automatically generate modular

¹ In the paper we use the terms 'robot opinion' and 'robot preference' interchangeably.

control software in the form of probabilistic finite state machines, starting from a set of predefined atomic behaviors and conditional state transitions through an optimization process. Bottom-up approaches have been used to program a number of different robot swarm behaviors: pattern formation behaviors, aimed at distributing the swarm in space according to desired properties [21–23, 31, 32]; navigation behaviors, aimed at coordinating the movement of the swarm in the environment [10]; and collective decision-making behaviors, in which the swarm has to take a decision about how to distribute its components (i.e., the robots) among different tasks [6] or which option to unanimously choose [30].

Centralized methods for the control of multi robot systems have also been proposed, in particular for navigation problems, such as deployment of robots in cooperative surveillance [33], target tracking [14], path planning [1, 28], or formation control [8]. The purpose of central control can vary between different tasks, but generally it includes calculating the motion plans for the single robots, allowing the robots to localize themselves by sensing and providing global information, or simply providing updated mission goals [41]. Some approaches can be found in which a distributed swarm behavior also relies on an external control entity to initiate or correct its functioning, such as in the work of Berman et al. [2], where a central unit broadcasts updated transition parameters for task allocation.

One notable exception to the above-mentioned approaches where the control is either fully self-organizing or centralized, is the recent work by Mathews et al. [18]. In this work, robots in a swarm are able to physically merge into a single entity, named a ‘mergeable nervous system robot’ (MNS-robot for short), comprising one single brain robot which acts as central controller for the robot aggregate. While both our work and the one of Mathews et al. share the idea of a centralized form of control internal to the swarm, the MNS aims at obtaining swarms able to morphologically adapt to the task of interest, while our focus is on designing a swarm able to monitor and influence its own behavior so as to increase its performance.

2.2 Consensus Achievement

Consensus achievement is one of the two branches of collective decision-making, the other being task allocation [4], and refers to the problem of having a robot swarm select a single option among different alternatives to maximize the benefits of the swarm [35]. Many scenarios have been proposed by the community, mostly inspired by biological systems such as ants choosing the shortest path connecting a pair of locations [7], or honeybees collectively selecting the best site for relocation of the swarm [25]. Montes de Oca et al. [19] proposed a collective decision-making strategy based on the majority rule and the concept of latent voters (i.e. after updating their opinions, agents do not take part in the decision making process for a stochastic amount of time) first described by Lambiotte et al. [16]. We utilize a similar concept in our scenario: after updating their opinion, agents enter a latent phase during which first they forage from the source indicated by their opinion and then disseminate their opinion to other

robots. Valentini et al. [38] reviewed the best-of- n problem for robot swarms in all of its variations, proposing two taxonomies to classify the literature, one based on the relation between cost and quality of each option, and one based on the design approaches. Despite the variety of methods proposed for consensus achievement problems in robot swarms [12, 13, 29], to our knowledge, our work is the first one that proposes to use an emerging control entity to estimate and leverage global information to influence the collective decision-making process.

3 Methods

3.1 Experimental Setup

We consider a binary resource selection problem for a robot swarm performing a foraging task. We define an environment consisting of an arena of size $200 \times 100 \text{ cm}^2$ divided into three areas: a nest ($80 \times 100 \text{ cm}^2$) positioned in the center of the arena, and two resource locations ($60 \times 100 \text{ cm}^2$ each) on each side of the nest. These locations, called source A and source B, have different costs σ_A and σ_B , with $\sigma_A < \sigma_B$ in our experiments. The cost of a resource location reflects the time required to collect resources from that source, representing features such as how deep a robot would have to dig for minerals, or how far the source is from the central nest. Two light sources are positioned on one side of the arena in order to provide the robots with a light gradient and to enable them to navigate the environment. The robots are initially placed in the nest and have an initial opinion for a preferred source when the experiment starts. Initially, robots in the swarm are equally split among the two options. Robots perform the foraging task by collecting resources from their currently preferred source, and then returning to the nest. Robots in the nest can change their opinion based on the opinions of neighboring robots by applying a decision mechanism. The goal of the swarm is to achieve consensus on the best source (which is always source A in our experiments).

We implemented this scenario using the ARGoS3 simulator [27] and the ARGoS3 Kilobot plug-in [26]. Figure 1a shows a view of the environment and of the swarm of Kilobots implemented inside the simulation, where source A and source B are represented, respectively, by the blue and red areas. The Kilobot [31] is a low cost and small size (3.3 cm diameter) autonomous robot. It is able to communicate with other Kilobots at a distance of up to 10 cm via infrared communication, to sense ambient light, and to move by means of 2 vibrating motors and 3 rigid legs. By means of an ARGoS loop function, we provide the Kilobots with the ability to detect whether or not they are in close proximity of a wall, in which area of the environment they currently are (i.e., nest, source A, source B), and, in case they are in one of the two sources, the source quality.

3.2 Self-organizing Behavior

We implement the self-organizing behavior with indirect modulation of the latent phase in the decision-making process [35]. In this phase the robots alternate

between dissemination and exploration. During the exploration phase, robots forage from their preferred resource location for a time drawn from a normal distribution with mean $g \cdot \sigma_i$ and standard deviation $g/10$, where $\sigma_A = 1$ and $\sigma_B \geq 1$ are the costs, respectively, of source A and source B. In the dissemination phase, robots broadcast their current opinion inside the nest and listen to the opinions of neighboring robots for a time drawn from an exponential distribution with mean q ; differently from the exploration time, the dissemination time is not modulated. At the end of the dissemination phase, the robots apply a decision rule on a set of opinions containing the last G opinions received from their neighbors with the aim to decide whether or not to switch their current opinion. After that, robots enter the exploration phase. We implemented two decision rules: the voter model, where a robot changes its opinion to the one of a randomly selected neighbor, and the majority rule, where a robot selects its opinion to be the one of the majority of its neighbors.

During both the dissemination and the exploration phases, robots move randomly, by alternating periods of straight motion with periods of rotating motion. Forward motion lasts for an amount of time drawn from a normal distribution with mean 20 s and standard deviation 5 s while the rotation motion lasts for an amount of time drawn from a normal distribution with mean 3 s and standard deviation 0.5 s. Additionally, when robots move closer than 5 cm to the edges of the arena, they perform wall avoidance by turning on the spot in a random direction and then moving forward. Between dissemination and exploration, robots have to move from the nest to the foraging sites and vice-versa. To do so, robots perform a gradient-following routine, by sensing the light intensity received from the light sources. Robots following the light gradient move forward while keeping track of the minimum and maximum light intensities sampled in intervals of 5 s. If a robot detects that it is not following the light gradient in the desired direction, it turns on the spot (using the same parameters as the random walk rotation) and then moves forward again, until it finds the correct direction of motion. Robots always show their current opinion by switching their on-board LED to the color of their preferred source.

Because of the shorter time required to forage from the source with lower cost, robots foraging from the best source will return to the nest more frequently and have more chances to disseminate their opinion in the nest: this results in a higher chance for their opinion to be observed from other robots as they apply the decision rule which biases the swarm towards consensus for the best option. The swarm is thus able to slowly achieve consensus on the best source as robots repeat the exploration-dissemination-decision rule cycle. In the following, we will refer to robots performing the behavior described in this section as SO robots.

3.3 SHCS Implementation

In our hybrid implementation, we introduce a second behavior in addition to the self-organizing one described in the previous section. Robots of the swarm can be either part of the control entity (SHCS robots) or be SO robots. Moreover,

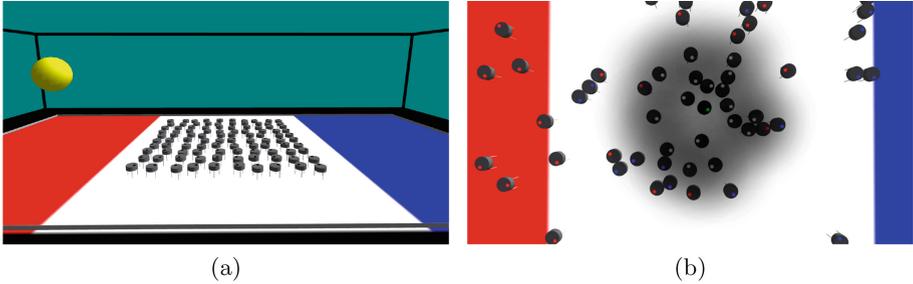


Fig. 1. View of the environment and Kilobot swarm implemented with the ARGoS3 simulator (a) and of the SHCS during a simulated experiment (b). The shaded area shows the communication range of the SHCS considered as a whole. SHCS robots show their LED in green (seed robot) or white (remaining SHCS robots); robots showing blue and red LEDs are SO robots, with the color representing their current opinion. (Color figure online)

robots can switch between these two modalities. At the beginning of the experiment, the swarm allocates its workforce between SHCS robots and SO robots. To do so, the robots select a seed robot around which they start an aggregation process to form the SHCS entity. The seed robot is selected through a self-exclusion process starting with a connected swarm² placed in the nest. The connectivity requirement strongly reduces the probability of selecting multiple robot seeds. Each robot spends the first 10 s of the experiment turning on the spot and sampling light values. Then, for the next 10 s, robots broadcast the minimum and maximum light measurement perceived in the swarm, initially set to their own perceived value and later updated according to the received messages. Additionally, robots also broadcast a randomly generated number between 0 and 255. Robots who find themselves outside of a 10% range from the mean value of the light perceived by the swarm (based on information received from neighbors) exclude themselves from the selection process and become SO robots. The purpose of this initial procedure is to obtain a selection of candidate seed robots that is positioned at an intermediate distance from the light source. Then, these candidate seed robots compare their own randomly generated number with those received from their neighbors and, if they receive a lower value, they exclude themselves from the process and become SO robots. After an additional 10 s, all remaining robots in the process become SHCS robots. The aim of this final part of the procedure is to maximize the probability to select a single seed robot. The total procedure to select the seed robot requires around 30 s.

SHCS robots, initially represented by the sole seed robot selected with the above procedure, maintain a representation of their position h inside the aggregate in a manner similar to that of Nagpal et al. [20], and share this value with their neighbors as part of a *heartbeat* protocol. The seed robot has a posi-

² A swarm is connected if a path of communicating robots can be found between any two robots in the swarm.

tion $h = 0$. All other SHCS robots in the aggregate set their position h to $h = h'_{min} + 1$ where h'_{min} is the minimum position received from neighboring SHCS robots. In our experiments, we limit the size of the SHCS aggregate by imposing a maximum position $h = 3$, that is, 3 levels of SHCS robots surrounding the seed robot. SO robots that perceive SHCS robots join the SHCS aggregate with probability $p = \frac{0.1}{h+1}$ if $h \in \{0, 1, 2\}$, where h is the position of the SHCS robot broadcasting the message. If the perceived position of the SHCS robot broadcasting the message is $h = 3$, SO robots do not join the aggregate. Once joined the SHCS aggregate, robots estimate their distance from neighboring SHCS robots by measuring the intensity of the infrared signal of received messages. If a SHCS robot with position h is too close (i.e., distance < 40 mm) or too far (i.e., distance > 70 mm) from his neighbors at position $h' = h - 1$, the SHCS robot will try to reposition itself at a favorable distance by moving in a random direction while it does not move otherwise. SHCS robots may lose connectivity from the aggregate during repositioning or due to collisions with other robots. If an SHCS robot loses connectivity for more than 10 s, it becomes an SO robot. This process allows the SHCS aggregate to initially form around the seed robot in a distributed manner and to maintain a stable dimension robust to connectivity failures. Figure 1b shows a top-view of the SHCS aggregate and its communication range during a simulated experiment.

SHCS robots continuously broadcast a *heartbeat* message with the aim (i) to maintain a database of the last 30 source preferences received from SO robots and (ii) to use this database to influence the preference of SO robots. A heartbeat message is composed of the id of the sending SHCS robot, its position h , a robot preference taken from its database, and a decision-making outcome. Whenever an SHCS robot receives a new opinion, either from a heartbeat message or from an SO robot, it adds the received opinion to its database (in a first-in first-out manner) and sets this opinion as the robot preference to share in the heartbeat message. SHCS robots generate a new decision-making outcome each time they send a new heartbeat; to do so, they use either the majority rule or the voter model applied to a set of G preferences randomly selected from their database.

SO robots behave as described in Sect. 2.2 except when receiving a heartbeat message. In this case, if a SO robot is in the dissemination phase, it immediately changes its opinion to match that contained in the decision-making outcome of the heartbeat message, terminates the dissemination phase, and returns to the foraging task. This mechanism improves the efficiency of the swarm as SO robots spend more time foraging and less time disseminating their opinions.

4 Experiments

We perform a series of simulation experiments to compare the hybrid control system (SHCS) approach with the fully self-organizing (SO) approach. In our experiments, we keep the cost of source A constant to $\sigma_A = 1$ and vary the cost of source B in $\{1.11, 1.25, 1.43, 1.67, 2\}$. We use a swarm of 100 robots of which 50 have initial opinion A and 50 have initial opinion B. The mean duration of

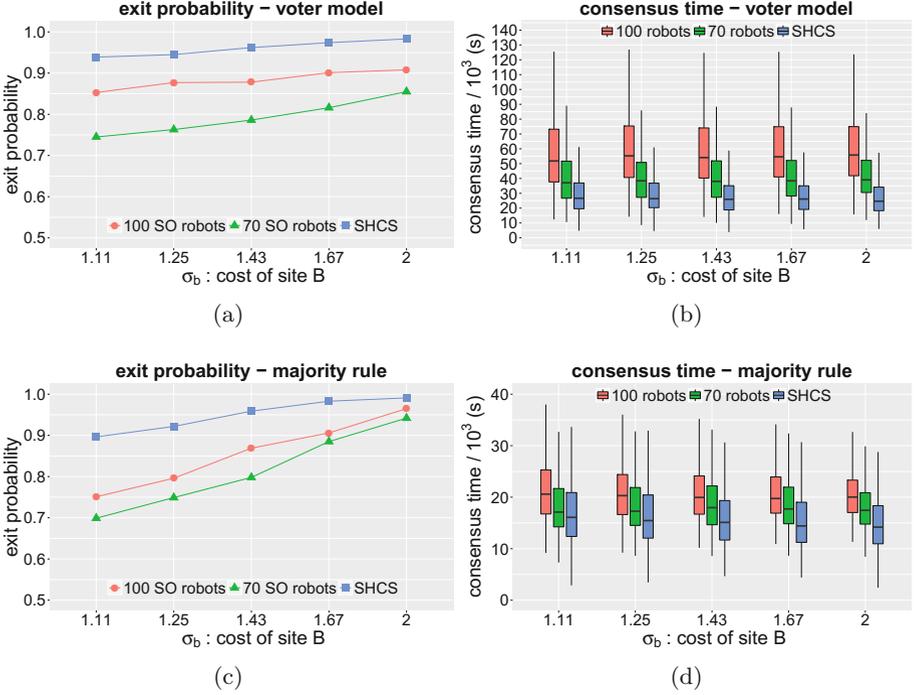


Fig. 2. Simulation results with SHCS, 100 SO robots, and 70 SO robots for varying σ_B : exit probability (a) and consensus time (b) for the voter model; exit probability (c) and consensus time (d) for the majority rule. Results obtained running 1000 simulations for each tested condition.

the dissemination and exploration phases is set, respectively, to $q = 300$ s and $g = 600$ s. We test two decision rules, the majority rule and the voter model, with a group size of $G = 3$ preferences. We perform 1000 simulation runs for each value of σ_B for both the SO approach and the SHCS one. We consider two metrics: exit probability, computed as the proportion of simulations converging to a consensus for source A , and the mean consensus time, computed over all simulations. Since the average SHCS size during experiments was approximately 30 robots, we performed an additional set of experiments implementing the SO approach with a swarm of $100 - 30 = 70$ robots, in order to compare the performance of the SHCS with the SO approach over a similar number of SO robots actively pursuing the foraging task.

Figure 2 shows the exit probability and mean consensus time obtained with the three implementations (SHCS for a swarm of 100 robots, 100 SO robots, 70 SO robots) for the two tested decision rules: voter model and majority rule.

Figure 2a shows the exit probability for the voter model. The SHCS implementation maintains a value above 0.95 for all of the considered values of σ_B while the SO implementations are considerably worse. The accuracy for all

the three systems increases as the cost of source B increases. This is because the decision-making problem becomes simpler as the difference in cost between source A and source B increases. The SHCS implementation performs similarly for the majority rule (Fig. 2c), where its exit probability maintains values around 0.9 even at lower values of σ_B ; the performances of the SO swarms instead are significantly worse. The 70 SO robot swarm has lower exit probability than the 100 SO robot swarm, and both of them are outperformed by the SHCS approach for all of the considered cases. Overall, the majority rule when compared to the voter model obtains a higher exit probability for the easier cases and a lower exit probability for the more difficult ones, in agreement with what reported in previous works [39].

Figure 2b shows the consensus time for the voter model. The SHCS shows significantly ($p < .001$, Wilcoxon rank-sum test) lower consensus times than both SO implementations. The 70 SO robots swarm shows lower consensus times than the 100 SO robots one, again coherently with previous literature work. Figure 2d shows the consensus time for the majority rule. The SHCS implementation results faster than both SO implementations for lower difficulties; however, the 70 SO robots swarm shows similar (even though statistically different, $p < .001$, Wilcoxon rank-sum test) consensus time at higher difficulties. The consensus time for all the three implementation slowly decreases as the cost σ_B increases, for both decision rules. Overall, the majority rule shows a significantly ($p < .001$, Wilcoxon rank-sum test) lower consensus time than the voter model, resulting in a speed vs accuracy trade-off between the two decision rules [37].

5 Discussion

The results of our experiments show the potential of the SHCS approach which is able to improve the performance of a fully self-organizing robot swarm in a collective decision-making problem. The SHCS approach leverages information regarding the global state of the opinions in the swarm to influence the individual decisions of SO robots. This results in a higher accuracy of the swarm in terms of the probability to choose the best resource location compared to the accuracy of the SO swarm (Figs. 2a and c). Additionally, the SHCS speeds up the decision-making process by allowing robots to terminate the dissemination phase as soon as they get in contact with the SHCS aggregate, since the dissemination of their opinion is performed by the SHCS. The faster convergence to a collective decision shown in Figs. 2b and d derives from a combination of the shorter dissemination phase and the more accurate information provided by the SHCS robots. In future work, we intend to investigate the extent of the contribution of each of the two mechanisms.

One may conjecture that the difference in performance between the SHCS approach and the fully SO approach is due to the fact that the SHCS swarm is actually relying on a smaller swarm size to actively perform the decision-making task. In our experiments, we measured an average of 30 robots composing the SHCS aggregate, leaving 70 SO robots to perform the self-organizing behavior.

However, the results obtained with a swarm of 70 SO robots are significantly different and of lower quality than those obtained with the SHCS approach. These results rule out the above conjecture that the difference in number of SO robots is responsible for the different performances between the SHCS and the SO approach.

It should be noted that in our experiments we use constant probabilities for SO robots to join the SHCS and limit the SHCS to three levels, preventing the SHCS from extending to the entire swarm. However, it would be interesting to extend this approach to include perceived features in the environment, for example by changing the probabilities with which SO robots join the SHCS depending on ambient light values, in order to obtain a more dynamic system.

6 Conclusions and Future Work

In this paper, we proposed a new control strategy for a robot swarm based on a combination of centralized information and self-organized behaviors. We called this control strategy Swarm Hybrid Control System (SHCS) and we investigated this idea with a preliminary implementation of the SHCS approach for a problem of consensus achievement in a binary resource-selection scenario. Our system is characterized by a control entity, having the form of an aggregate of SHCS robots and arising through a self-organizing process, with the purpose to estimate information about the global state of the swarm and to use this information to influence the collective decision-making process. We have shown how, for both the majority rule and the voter model, our system is able to outperform the fully self-organizing approach by achieving a shorter consensus time while providing higher accuracy of the collective decision in terms of exit probability. In the near future, we plan to implement the consensus achievement scenario presented in this paper on a real swarm of 100 Kilobots by leveraging the potential of a 2 m² Kilogrid system [36].

As future work, we are interested in investigating how the proportion of SHCS robots, the shape of the SHCS or the usage of multiple smaller SHCS each controlling a portion of the swarm can impact the performance of the system, as well as how our control approach can be applied to different scenarios, such as task allocation and pattern formation. We also intend to investigate whether automatic design techniques can be used to generate the controllers for the robots of our hybrid swarm.

Acknowledgements. Gabriele Valentini acknowledges support from the NSF grant No. PHY-1505048. Marco Dorigo acknowledges support from the Belgian F.R.S.-FNRS, of which he is a Research Director. The work presented in this paper was partially supported by the FLAG-ERA project RoboCom++ and by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement number 681872).

References

1. Antonelli, G., Chiaverini, S.: Kinematic control of platoons of autonomous vehicles. *IEEE Trans. Robot.* **22**(6), 1285–1292 (2006)
2. Berman, S., Halasz, A., Hsieh, M., Kumar, V.: Optimized stochastic policies for task allocation in swarms of robots. *IEEE Trans. Robot.* **25**(4), 927–937 (2009)
3. Bonabeau, E., Dorigo, M., Theraulaz, G.: *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York (1999)
4. Brambilla, M., Ferrante, E., Birattari, M., Dorigo, M.: Swarm robotics: a review from the swarm engineering perspective. *Swarm Intell.* **7**(1), 1–41 (2013)
5. Brutschy, A., Scheidler, A., Ferrante, E., Dorigo, M., Birattari, M.: “Can ants inspire robots?” Self-organized decision making in robotic swarms. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4272–4273. IEEE Press (2012)
6. Brutschy, A., Pini, G., Pinciroli, C., Birattari, M., Dorigo, M.: Self-organized task allocation to sequentially interdependent tasks in swarm robotics. *Auton. Agents Multi-Agent Syst.* **28**(1), 101–125 (2014)
7. Campo, A., Gutiérrez, Á., Nouyan, S., Pinciroli, C., Longchamp, V., Garnier, S., Dorigo, M.: Artificial pheromone for path selection by a foraging swarm of robots. *Biol. Cybern.* **103**(5), 339–352 (2010)
8. De La Cruz, C., Carelli, R.: Dynamic modeling and centralized formation control of mobile robots. In: *IECON 2006–32nd Annual Conference on IEEE Industrial Electronics*, pp. 3880–3885. IEEE (2006)
9. Dorigo, M., Birattari, M., Brambilla, M.: Swarm robotics. *Scholarpedia* **9**(1), 1463 (2014)
10. 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. *Adapt. Behav.* **20**(6), 460–477 (2012)
11. Francesca, G., Brambilla, M., Brutschy, A., Trianni, V., Birattari, M.: AutoMoDe: a novel approach to the automatic design of control software for robot swarms. *Swarm Intell.* **8**(2), 89–112 (2014)
12. Francesca, G., Brambilla, M., Trianni, V., Dorigo, M., Birattari, M.: Analysing an evolved robotic behaviour using a biological model of collegial decision making. In: Ziemke, T., Balkenius, C., Hallam, J. (eds.) *SAB 2012. LNCS (LNAI)*, vol. 7426, pp. 381–390. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33093-3_38
13. Gutiérrez, Á., Campo, A., Monasterio-Huelin, F., Magdalena, L., Dorigo, M.: Collective decision-making based on social odometry. *Neural Comput. Appl.* **19**(6), 807–823 (2010)
14. Hausman, K., Müller, J., Hariharan, A., Ayanian, N., Sukhatme, G.S.: Cooperative multi-robot control for target tracking with onboard sensing. *Int. J. Robot. Res.* **34**(13), 1660–1677 (2015)
15. King, J., Pretty, R.K., Gosine, R.G.: Coordinated execution of tasks in a multiagent environment. *IEEE Trans. Syst. Man Cybern.-Part A: Syst. Hum.* **33**(5), 615–619 (2003)
16. Lambiotte, R., Saramäki, J., Blondel, V.D.: Dynamics of latent voters. *Phys. Rev. E* **79**, 046107 (2009)
17. Lindsey, Q., Mellinger, D., Kumar, V.: Construction with quadrotor teams. *Auton. Robot.* **33**(3), 323–336 (2012)

18. Mathews, N., Christensen, A.L., O'Grady, R., Mondada, F., Dorigo, M.: Mergeable nervous systems for robots. *Nat. Commun.* **8**(1), 439 (2017)
19. Montes de Oca, M.A., Ferrante, E., Scheidler, A., Pinciroli, C., Birattari, M., Dorigo, M.: Majority-rule opinion dynamics with differential latency: a mechanism for self-organized collective decision-making. *Swarm Intell.* **5**(3–4), 305–327 (2011)
20. Nagpal, R., Shrobe, H., Bachrach, J.: Organizing a global coordinate system from local information on an ad hoc sensor network. In: Zhao, F., Guibas, L. (eds.) *IPSN 2003*. LNCS, vol. 2634, pp. 333–348. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36978-3_22
21. Nouyan, S., Campo, A., Dorigo, M.: Path formation in a robot swarm: self-organized strategies to find your way home. *Swarm Intell.* **2**(1), 1–23 (2008)
22. Nouyan, S., Dorigo, M.: Chain based path formation in swarms of robots. In: Dorigo, M., Gambardella, L.M., Birattari, M., Martinoli, A., Poli, R., Stützle, T. (eds.) *ANTS 2006*. LNCS, vol. 4150, pp. 120–131. Springer, Heidelberg (2006). https://doi.org/10.1007/11839088_11
23. Nouyan, S., Groß, R., Bonani, M., Mondada, F., Dorigo, M.: Teamwork in self-organized robot colonies. *IEEE Trans. Evol. Comput.* **13**(4), 695–711 (2009). <https://doi.org/10.1109/TEVC.2008.2011746>
24. Nowak, M.A.: Five rules for the evolution of cooperation. *Science* **314**(5805), 1560–1563 (2006)
25. Parker, C.A.C., Zhang, H.: Cooperative decision-making in decentralized multiple-robot systems: the best-of-N problem. *IEEE/ASME Trans. Mechatron.* **14**(2), 240–251 (2009)
26. Pinciroli, C., Talamali, M.S., Reina, A., Marshall, J.A.R., Trianni, V.: Simulating Kilobots within ARGoS: models and experimental validation. In: Dorigo, M. (ed.) *ANTS 2018*. LNCS, vol. 11172, pp. 176–187. Springer, Heidelberg (2018)
27. Pinciroli, C., et al.: ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intell.* **6**(4), 271–295 (2012)
28. Preiss, J.A., Honig, W., Sukhatme, G.S., Ayanian, N.: Crazyswarm: a large nano-quadcopter swarm. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3299–3304. IEEE (2017)
29. Reina, A., Dorigo, M., Trianni, V.: Towards a cognitive design pattern for collective decision-making. In: Dorigo, M., et al. (eds.) *ANTS 2014*. LNCS, vol. 8667, pp. 194–205. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-09952-1_17
30. Reina, A., Valentini, G., Fernández-Oto, C., Dorigo, M., Trianni, V.: A design pattern for decentralised decision making. *PLoS One* **10**(10), e0140950 (2015)
31. Rubenstein, M., Cornejo, A., Nagpal, R.: Programmable self-assembly in a thousand-robot swarm. *Science* **345**(6198), 795–799 (2014)
32. Şahin, E., et al.: SWARM-BOT: pattern formation in a swarm of self-assembling mobile robots. In: *2002 IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 1–6. IEEE Press, Piscataway (2002)
33. Saska, M., Vonásek, V., Chudoba, J., Thomas, J., Loianno, G., Kumar, V.: Swarm distribution and deployment for cooperative surveillance by micro-aerial vehicles. *J. Intell. Robot. Syst.* **84**(1–4), 469–492 (2016)
34. Trianni, V., Groß, R., Labella, T.H., Şahin, E., Dorigo, M.: Evolving aggregation behaviors in a swarm of robots. In: Banzhaf, W., Ziegler, J., Christaller, T., Ditzrich, P., Kim, J.T. (eds.) *ECAL 2003*. LNCS (LNAI), vol. 2801, pp. 865–874. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39432-7_93

35. Valentini, G.: Achieving Consensus in Robot Swarms: Design and Analysis of Strategies for the Best-of-N Problem. Springer International Publishing, Cham (2017). <https://doi.org/10.1007/978-3-319-53609-5>
36. Valentini, G., et al.: Kilogrid: a novel experimental environment for the kilobot robot. *Swarm Intell.* **12**(3), 245–266 (2018)
37. Valentini, G., Brambilla, D., Hamann, H., Dorigo, M.: Collective perception of environmental features in a robot swarm. In: Dorigo, M., et al. (eds.) ANTS 2016. LNCS, vol. 9882, pp. 65–76. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44427-7_6
38. Valentini, G., Ferrante, E., Dorigo, M.: The best-of-n problem in robot swarms: formalization, state of the art, and novel perspectives. *Front. Robot. AI* **4**, 9 (2017)
39. Valentini, G., Ferrante, E., Hamann, H., Dorigo, M.: Collective decision with 100 Kilobots: speed versus accuracy in binary discrimination problems. *Auton. Agents Multi-Agent Syst.* **30**(3), 553–580 (2016)
40. Weigel, T., Gutmann, J.S., Dietl, M., Kleiner, A., Nebel, B.: CS Freiburg: coordinating robots for successful soccer playing. *IEEE Trans. Robot. Autom.* **18**(5), 685–699 (2002)
41. Winfield, A.F., Holland, O.: The application of wireless local area network technology to the control of mobile robots. *Microprocess. Microsyst.* **23**(10), 597–607 (2000)