Recruitment Near Worksites Facilitates Robustness of Foraging E-puck Swarms to Global Positioning Noise*

Lenka Pitonakova[1,3], Alan Winfield[2] and Richard Crowder[1]

***Abstract*— We compare the ability of two different robot controllers for collective foraging to cope with noise in robot global positioning data and show how recruitment, in the form of broadcast messages near worksites, can make swarms more robust. Swarms of five e-puck robots are used in a semi-virtual environment, facilitated by the VICON positioning system. This setup allows us to control the amount of noise in the robot positioning data and to generate pseudo-random environments, while retaining important physical aspects of the experiment. The effect of inherent noise in the robot infra-red sensors, used for obstacle avoidance, is noted and the importance of modelling such noise in agent-based simulations is highlighted.**

I. INTRODUCTION

Multi-robot foraging is a paradigm for the study of a variety of real-world robot tasks, such as object retrieval [1], [2], [3], autonomous transportation [4], collective decision-making [5], event servicing [3], [6], [7] and others. An important desired property of robot swarms is robustness to noise. Since their inherent robustness is limited [8], we must design robot controllers that, when employed on multiple cooperating robots, reduce any negative impacts of incorrect information on swarm performance. In this paper, we compare the ability of two robot controllers to cope with global positioning noise and show how recruitment in the form of broadcast messages can make a swarm more robust.

Various swarm foraging strategies have been studied previously, both in simulation and on real robots. The effect of sensory-motor noise during foraging was explored in the context of odometry-based navigation [9], threshold-based task allocation [10], nutritional gradient following [11], [12] and dynamic area coverage [13]. Here we concentrate on a central-place foraging task where robots need to find *worksites* in the environment and deliver virtual resources from them to a designated location (as in [9], [10]). Two foraging strategies are explored and their relative performance is thoroughly analysed. *Solitary* robots do not share information about where worksites are located, while *Broadcasters* recruit nearby robots to their worksites.

Swarms of five e-puck robots are used in a semi-virtual environment, facilitated by the VICON positioning system [14]. Gaussian noise is introduced into the VICON positioning data that the robots utilise for navigation. Using such a semi-virtual environment allows us to precisely control and analyse the effects of the noise, as well as to easily generate pseudo-random environments. It is shown that positional noise causes robots to loose track of where worksites are located, but that Broadcasters are able to cope with this problem by repeatedly recruiting each other to the vicinity of worksites. The effect of inherent noise in robot infra-red sensors, utilised for obstacle avoidance, is also noted and the importance of modeling such noise in agent-based simulations is highlighted. Finally, the results are compared to those from numerous simulation-based studies and it is proposed that a robot control strategy does not only affect *whether* a swarm can exhibit robustness to noise, but, more precisely, *what kind of* noise the swarm can be robust to.

II. METHODS

The robots were tasked with finding worksites in the experimental arena and with delivering resource from the worksites to a designated location, *the base* (Figure 1). The base was a quadrant with radius $r_B = 40$ cm, placed in one of the arena corners. The environment was characterised by the number of worksites, $N_W \in \{1, 3, 12\}$ and the minimal worksite distance from the base edge, $D \in \{0.7, 1.4\}$ m, resulting in six separate experimental scenarios. Virtual worksites with radius $r_W = 10$ cm were placed randomly at a distance between D and $D + 0.5$ m from the base at the beginning of each experiment run. The total amount of reward in the environment was $R'_T = 48$ units and each worksite had R'_T/N_W resource units at the beginning of a run. Each experiment consisted of ten runs.

All experiments were performed at the Bristol Robotics Laboratory (BRL) using five e-puck robots with a Linux extension board developed at the BRL [15]. The arena was approximately 2×1.5 m large. The experimental environment was semi-virtual - the robots physically interacted with each other and with the arena boundaries, but received information about their absolute position and the location of the base and worksites from an external *Server*. The Server maintained a virtual representation of the base, worksite, and robot positions, and kept track of the amount of resources left in the worksites. Robots communicated with the Server via an on-board Wi-Fi module, utilising a ROS interface. The Server also facilitated robot-to-robot communication by receiving communication messages from robots and sending them back to other relevant robots.

*This work was supported by EPSRC grants EP/G03690X/1, EP/N509747/1 and EP/R004757/1 and by Thales Group. A video accompanying this paper is available on the IROS 2018 proceedings website. The source code and data are available on https://doi.org/10.5258/SOTON/D0388. We would like to thank Wenguo Liu for providing his robot and computer programs that served as a basis for the programs used here.

[1]Dep. of Electronics and Computer Science, Univ. of Southampton, UK
[2]Bristol Robotics Laboratory, Univ. of the West of England, Bristol, UK
[3]contact@lenkaspace.net

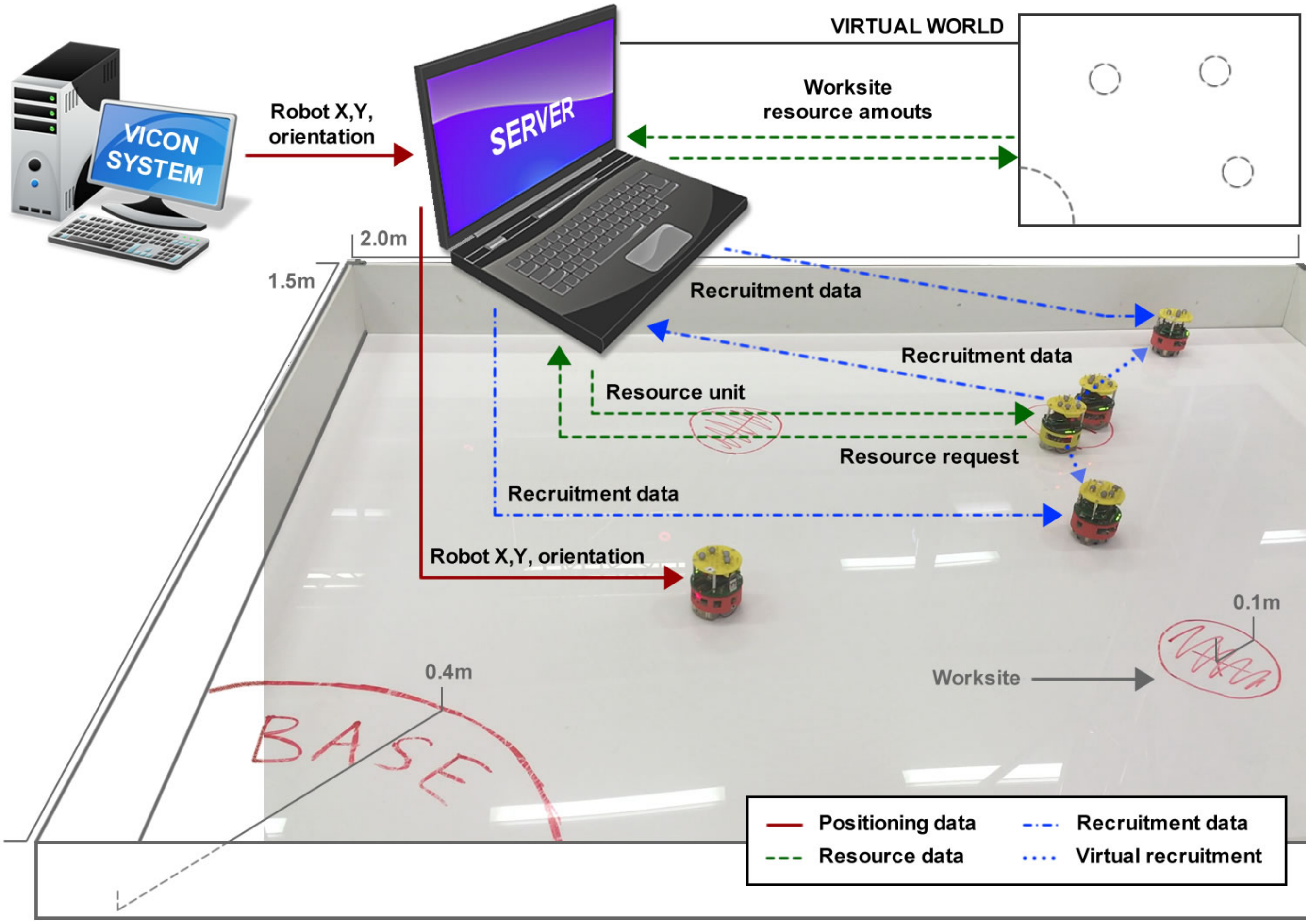

Fig. 1. The experimental setup consisting of the e-puck arena, a Server and a VICON system. Worksites are drawn on the arena floor for visualisation purposes only. The robots communicated with the Server in order to obtain positional, worksite and recruitment data. The Server maintained a virtual representation of the world, keeping track of base, worksite and robot locations and the amount of resource units in each worksite.

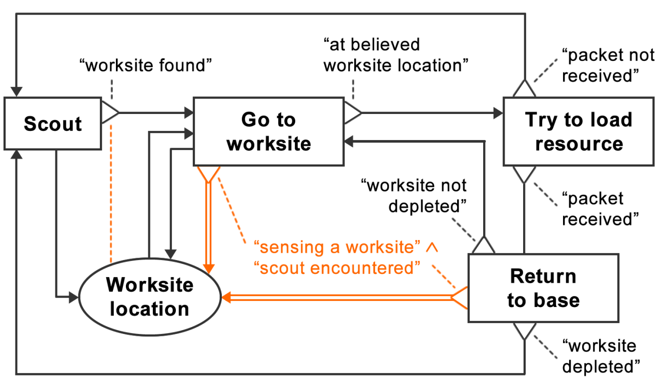

Fig. 2. A BDRML [16] representation of the robot controllers. Primitives and relations in black apply to both controllers. Additional primitives and relations, unique to the Broadcaster controller, are shown in orange.

Two robot control strategies were explored: Solitary and Broadcaster (Figure 2). Both strategies were implemented as finite-state machines. At the beginning of an experiment, all robots were placed at a random position and with a random orientation in the base. Each robot left the base immediately to perform scouting, i.e., to search for worksites using random motion. The robots moved at maximum speed of 8 cm/s and utilised a ring of infra-red sensors to avoid each other and the arena walls. The robots received their x and y positional coordinates and their absolute orientation from the Server approximately ten times per second. When a robot perceived that it was more than $D+0.5$ m away from the base, it turned back towards the base in order to remain in the area were worksites were supposed to be located.

The robots were also equipped with a virtual worksite sensor with range $r_W = 25$ cm. Once a robot was at least r_W cm away from a worksite, the Server informed it about the absolute worksite location. The robot then calculated a vector towards the worksite and travelled to it. Once at the worksite, the robot sent a resource request to the Server and the Server returned a virtual resource unit, provided that the robot was truly located at a worksite and that the worksite had remaining resource. The Server then decreased the number of resource units in the worksite by one and sent the remaining number of units to the robot. After the robot deposited its resource in the base, it returned to the worksite utilising its remembered location, provided that the worksite was not depleted during the robot's last visit. Otherwise, the robot resumed scouting.

Additionally, robots in the Broadcaster swarm were equipped with a virtual communication module with communication range $r_C = 1.25$ m, that they could use to recruit nearby robots to a worksite (as in, e.g. [4], [7], [9], [17]). When a robot was up to 25 cm away from a worksite, i.e., when it was sensing it, it sent the worksite location to the Server. The Server then probabilistically sent this information to all robots within the communication range of the recruiter. The probability of communication decreased exponentially with an increasing distance between robots. Once a scouting robot received the recruitment signal, it travelled to the advertised worksite and started foraging from it. Robots that were already foraging ignored the signal.

The experiment and robot controller parameters are listed in Table I. The parameters were selected based on our previous simulation experiments with MarXBots [3], and scaled with respect to the arena size, allowing for an approximate performance comparison to be made later on. In the simulation, the arena and the base were circular. The real-world arena represented a quarter of the simulated space, resulting in a scaling factor of 0.25. For example, the range at which simulated robots could sense worksites was 1 m, corresponding to sense range of 0.25 m in the real world.

The robots were subject to sensory-motor noise intrinsic to the robot hardware. However, by default, their navigation information was not severely affected by noise, since the VICON positioning system had a resolution of 0.1 mm and the Wi-Fi communication was fairly robust. In order to study the effect of positioning noise, independent experiments were performed in each scenario, where Gaussian noise with the variance $\sigma = 20cm$ was added separately to the x and y position coordinates that were sent to the robots. Because of this noise, the robots could arrive to incorrect worksite locations, unable to extract any resource. On such occasions, the robots abandoned their worksites and resumed scouting.

The robots also suffered random failures, mostly due to battery life time. When a robot stopped working, it remained at its location and created an obstacle for other robots. In order to maintain data consistency, one robot was allowed to fail in around 30% of runs in each experimental scenario. If more than one robot failed during an experimental run, or when the number of runs with a robot failure already reached 30%, the run was repeated.

III. Results

The swarms were first tested in experiments without positioning noise. In these experiments, only the inherent sensory-motor noise and occasional communication delays with the Server affected the robots. Nevertheless, the robots were always able to reach their worksites, although on some occasions, a robot took a longer route rather following a straight line to its destination.

TABLE I
THE EXPERIMENTAL PARAMETERS

Description	**Symbol**	**Value**
Number of robots	N_R	5
Arena size	-	2×1.5 m
Base radius	r_B	40 cm
Worksite radius	r_W	10 cm
Number of worksites	N_W	{1,3,12}
Min. worksite distance from base edge	D	{0.7,1.4} m
Total reward	R'_T	48
Robot worksite sensor range	r_W	25 cm
Robot communication range	r_C	1.25 m

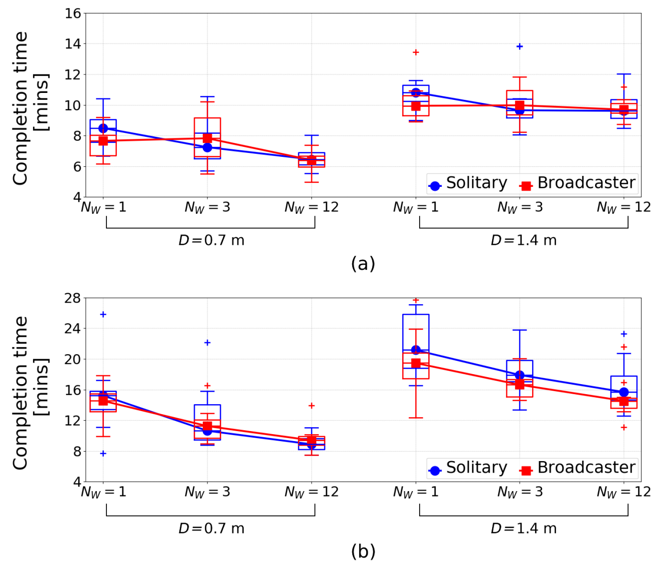

Fig. 3. Task completion time in various environments characterised by the number of worksites, N_W, and the minimum worksite distance from the base, D, in experiments (a) without and (b) with positional noise. Note the y-axis range in (b) is twice that of (a).

The Solitary and the Broadcaster swarms usually performed similarly when the same experimental scenario was considered (Figure 3a). There was some indication that Broadcasters outperformed Solitary swarms in environments with one worksite, although the difference in performance was not statistically significant. The task took longer to complete when worksites were further away from the base due to longer travel times and a larger arena that needed to be explored. Additionally, in small environments ($D = 0.7$ m), the completion time was on average about two minutes shorter when twelve worksites were placed in the arena, compared to when one or three worksites existed. In this experimental scenario, the swarm usually distributed itself among multiple worksites, since they were easy to find. This had two positive effects. Firstly, there was a smaller amount of congestion around each worksite. Secondly, there was a smaller chance that a worksite that a robot was returning to would be meanwhile depleted by other robots. On the other hand, in large environments ($D = 1.4$ m), the number of

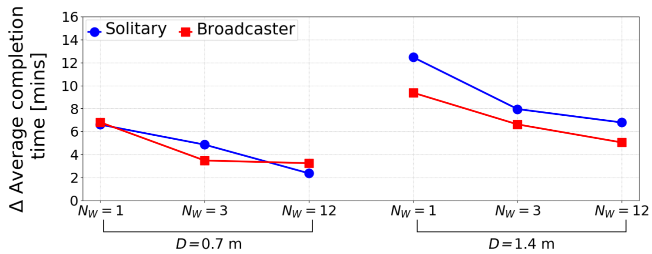

Fig. 4. The difference in the average task completion time in experiments with positional noise, compared to experiments without noise.

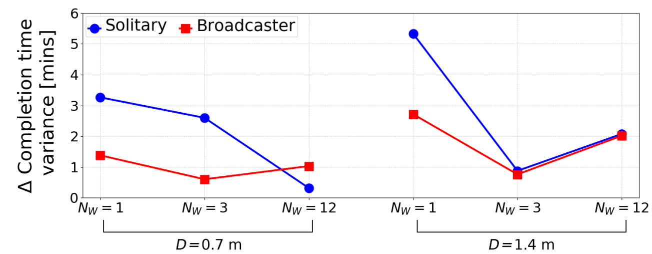

Fig. 5. The difference in the task completion time variance in experiments with positional noise, compared to experiments without noise.

worksites did not affect the completion time by a significant amount. Even when $N_W = 12$, the last few worksites were difficult to find, which made the opportunity of robots to distribute their effort across multiple sites less relevant.

When positional noise was added to the environment, robots often arrived to incorrect locations when navigating towards worksites, resulting in longer task completion times (Figure 3b). A robot that was unable to receive resource from an assumed worksite location resumed scouting and had to re-discover the worksite. The effect of N_W on the task completion time became more prominent, since re-discovery was more difficult when the worksite density was lower. Secondly, Broadcasters usually completed the task faster than Solitary swarms in larger arenas ($D = 1.4$ m). This result was statistically significant (ANOVA, $p = 0.05$) when the three data points, $N_W \in \{1, 3, 12\}, D = 1.4$ m were considered together for each swarm. Moreover, in each of the scenarios with $D = 1.4$ m, the average completion time of Broadcasters increased by a smaller amount compared to the Solitary swarms when noise was added (Figure 4). Finally, the negative effect of noise on the swarm performance variance was smaller in Broadcaster swarms, especially when only a single worksite existed in the environment (Figure 5). These results signify that Broadcaster swarms were, through recruitment, able to retain information about the location of worksites better than Solitary swarms. When a Broadcaster robot abandoned a worksite due to not being able to find it, there was a chance for it to be recruited to the site by a robot that was still foraging from it. This effectively increased the range at which robots were able to find worksites. On the other hand, Solitary robots were unable to help each other to locate worksites, since they did not share information.

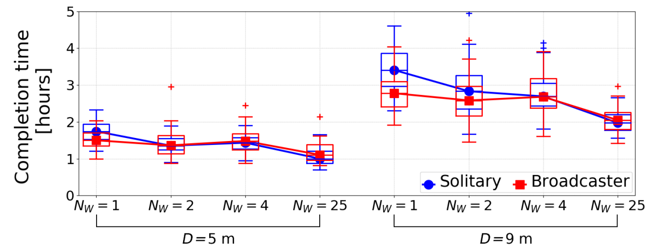

Fig. 6. Task completion time in various environments characterised by the number of worksites, N_W and the minimum worksite distance from the base, D, in simulated experiments with MarXBot robots [3].

IV. Related work and discussion

In many real-world swarm foraging experiments, items that need to be foraged from are often physically placed in the arena [2], [6], [10]. However, in the experiments presented here, a semi-virtual environment, similar to that in, e.g., [9], was used. Using a virtual representation of worksites allowed us to create pseudo-randomly-generated foraging worlds that were parametrised by D and N_W, which would be difficult to achieve repeatedly in a purely physical setup. Similarly, using virtual sensors made it possible to precisely control and explore the effect of noise in the modality of our interest. At the same time, we were able to incorporate physical aspects of the environment, such as walls and robot bodies, as well as the inherent sensory-motor noise of robots, that would be difficult to replicate precisely in a model.

We have previously studied the same robot controllers in the ARGoS simulator using 10, 25 and 50 MarXBot robots [3]. The robots were similar to e-pucks in shape and capabilities, but were about 2.5 times larger than the e-pucks. The simulated arena was also larger and it was circular, and the smallest worksite distance from the base was $D = 5$ m. A total of 25 environments were studied, with $D \in \{5, 9, 13, 17, 21\}$ m and $N_W \in \{1, 2, 4, 25\}$. Despite the differences in the experimental setups, there were similarities in swarm performance between the simulated and the real-world experiments. Firstly, the real-world study confirmed that swarms with different robot controllers do not perform significantly differently in the same environment when small swarms and small arenas are considered (compare Figures 3a and 6). This is because the positive impact of information sharing, such as the ability of robots to find worksites faster, as well its negative effects, such as a higher amount of congestion, are not significant in small experiments.

On the other hand, there were some differences between the simulation and real-world in terms of the correlation between the number of worksites and the task completion time. In simulation, environments with a smaller number of worksites always took a longer amount of time to complete, especially to Solitary swarms. In the real world, this trend was only observed when positional noise was added into the environment (Figure 3b). This result can be explained by considering the two forces that affect foraging swarm performance, namely the ability of robots to discover worksites and their ability to collect resource from them [3]. In the larger simulated experiments, the difficulty of swarms in environments with a small number of worksites resulted from a very small worksite density [3]. However, this was not the case in the real-world experiments without positional noise, where worksites were on average discovered more quickly when there was a smaller number of them (Figure 7).

Instead, the difficulty that the real-word swarms faced was severe congestion due to smaller arena size relatively to the robot size, as well as due to the presence of arena walls that created additional obstacles for the robot movement. The travel time between worksites and the base inversely correlated with the number of worksites and was the longest when only a single worksite existed in the environment (Figure 8a), suggesting the highest amount of congestion. The nature of the interplay between the effect of congestion and the effect of average worksite discovery time resulted in task completion times that were mostly similar regardless of N_W. Although the robots discovered all worksites quicker when there was a smaller number of worksites to discover (Figure 7), they could not distribute their foraging effort between multiple worksites, increasing congestion. On the other hand, where many worksites were placed in the arena, the congestion was less severe but it took the swarm longer to discover all worksites. Secondly, it was also observed that robots found it more difficult to deal with congestion once it occurred in the real-world, compared to in the simulation. The previous simulation study [3] did not take inherent infra-red sensor noise into account, allowing robots to avoid each other fairly quickly and mostly without touching. In the real world, each infra-red sensor had a different range, where some robot sensors could only detect obstacles up to about 2 cm away from the robot. This made the avoidance behaviour much less efficient than in the simulation, increasing the negative effect of overcrowding.

The balance between worksite discovery and congestion changed when positional noise was introduced. The foraging trip time decreased, especially when there was a small number of worksites and when worksites were closer to the base (Figure 8b), suggesting a smaller amount of congestion. On the other hand, worksite discovery had to be repeated multiple times, increasing the impact of worksite density. The relationship between the number of worksites and task completion time resembled that found in the larger simulated experiments, as environments with a small number of worksites became very difficult to forage in.

Other research has also shown that recruitment to worksites can increase swarm robustness during foraging [10]. Similarly, "social odometry" can be utilised by robots to cope with the negative effect of noise through exchanging confidence levels about the information that they share [9]. When using odometry, a robot maintains a relative vector towards a worksite by integrating information about its wheel movement at each time step. Robots that have travelled a larger distance, and are therefore likely to have more incorrect positional information as a result of accumulated noise, can learn worksite locations from robots that have visited them more recently. However, social odometry would not be able cope with the noise introduced in the experiments presented in this paper, since the robots here used a global positioning system, making their errors non-accumulative. Nevertheless, it would be interesting to explore the idea of taking into account some measure of confidence during recruitment. For example, a robot in a larger swarm could follow a recruitment signal with a greater probability if multiple robots were recruiting to a similar area.

Utilising social information can also help robots with noisy sensors to navigate the environment more efficiently during other collective tasks such as flocking and aggregation [11], [12]. Social information can help individuals to avoid mistakes, such as getting attracted to local maxima of a reward gradient, since the effect of individual choices and errors is averaged over the robot population. However, sensory-motor noise can also have a positive effect on swarm performance. For example, during multi-robot dispersion tasks, such as area coverage, noise can decrease redundancy, helping the swarm to spread through the environment better [13].

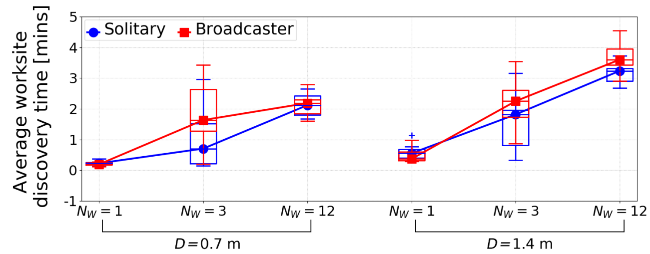

Fig. 7. The average worksite discovery time in various environments characterised by the number of worksites, N_W, and the minimum worksite distance from the base, D, in experiments without positional noise.

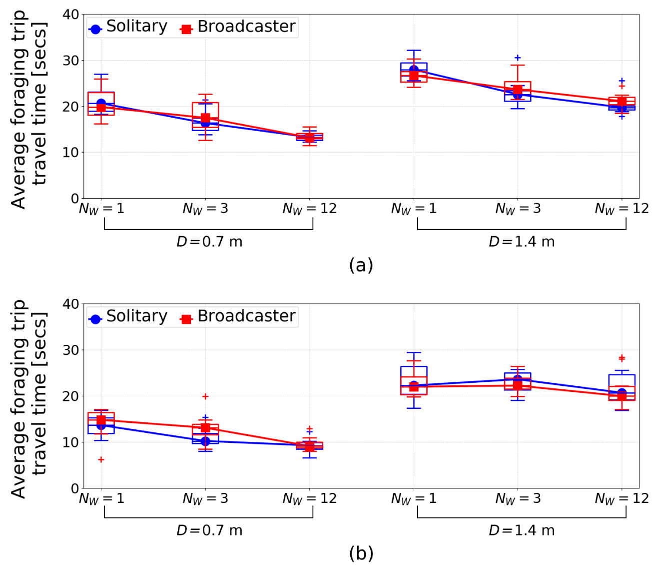

Fig. 8. The average travel time between the base and a worksite in various environments characterised by the number of worksites, N_W, and the minimum worksite distance from the base, D, in experiments (a) without and (b) with positional noise.

Finally, it is also interesting to consider how swarms with different communication strategies can deal with sensory-motor noise during foraging. In bee-inspired swarms, robots exchange information about worksites in the base. When there are worksites of different qualities in the environment, bee-inspired robots can adjust their recruitment effort based on the perceived worksite quality (for example, based on the net amount of reward available from a single resource unit) [5]. Such recruitment is highly robust to noise in perception of worksite quality that individuals might experience, making it possible for the majority of the swarm to forage from the most profitable worksite. On the other hand, bee-inspired swarms find it difficult to cope with odometry noise that affects their estimate of worksite *locations* [1]. Because bee-inspired robots travel to the base to recruit, the amount of accumulated error is fairly large at the time of recruitment, compared to robots that recruit near worksites, such as Broadcasters. Moreover, the error propagates to many robots as a direct result of recruiting in a small designated area. This suggests that a control strategy of robots does not only affect *whether* swarms are robust to sensory-motor noise, but also *what kind of* noise they can be robust to.

V. Conclusion and future work

In this paper, we have explored two robot swarm control strategies in the context of central-place foraging in a semi-virtual environment. The robots navigated using a virtual global positioning sensor that was subject to noise. The Broadcaster swarm was more robust to errors in the positional data, since robots that arrived to incorrect worksite locations had a chance to receive new worksite locations from other members of the swarm. This allowed the swarm to maintain information about approximate worksite locations better than the Solitary swarm, where robots did not recruit.

This study has confirmed our previous simulation result that a choice of robot control strategy plays a small role in small swarms and in small environments, i.e., when worksites are relatively easy to find and when the effect of information sharing cannot accumulate due to a small robot population size. However, adding positional noise increased the difficulty of returning to worksites during foraging, making worksite density, as well as the choice of a robot control strategy, more important factors that affected the swarm performance. Secondly, it was found that real-world swarms experienced congestion more severely compared to their simulated counterparts, not only due to the small arena size, but also due to the inherent noise in the infra-red sensors that the robots used for obstacle avoidance. Modelling this noise, or its effects on robot navigation, in future simulation studies would be worthwhile. Finally, a comparison with studies of bee-inspired swarms suggests that the choice of a robot control strategy does not only impact whether a swarm is robust to sensory-motor noise, but, more importantly, what kind of noise it can be robust to.

Our next step will be to describe in detail the forces that play a role in the swarm behaviour reported here by utilising the Information-Cost-Reward framework [3]. Furthermore, creating an agent-based simulation that includes the inherent noise experienced by the robots is a possible future direction of this research. Such a model would make it possible to explore more kinds of noise, such as motor noise, infra-red sensory noise and communication noise, separately and in a greater detail than real-world experimentation allows.

References

[1] L. Pitonakova, R. Crowder, and S. Bullock, "Understanding the role of recruitment in collective robot foraging," in *Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems (ALIFE 14)*, H. Lipson, H. Sayama, J. Rieffel, S. Risi, and R. Doursat, Eds. Cambridge, MA: MIT Press, 2014, pp. 264–271.

[2] E. Castello, T. Yamamoto, F. D. Libera, W. Liu, A. F. T. Winfield, Y. Nakamura, and H. Ishiguro, "Adaptive foraging for simulated and real robotic swarms: the dynamical response threshold approach," *Swarm Intelligence*, vol. 10, no. 1, pp. 1–31, 2016.

[3] L. Pitonakova, R. Crowder, and S. Bullock, "The Information-Cost-Reward framework for understanding robot swarm foraging," *Swarm Intelligence*, vol. 12, no. 1, pp. 71–96, 2017.

[4] J. Wawerla and R. T. Vaughan, "A fast and frugal method for team-task allocation in a multi-robot transportation system," in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA 2010)*. Piscataway, NJ: IEEE Press, 2010, pp. 1432–1437.

[5] G. Valentini, H. Hamann, and M. Dorigo, "Self-organized collective decision making: The weighted voter model," in *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014)*. New York: ACM, 2014, pp. 45–52.

[6] M. O. F. Sarker and T. S. Dahl, "Bio-Inspired communication for self-regulated multi-robot systems," in *Multi-Robot Systems, Trends and Development*, T. Yasuda, Ed. InTech, 2011, pp. 367–392.

[7] F. Ducatelle, G. A. Di Caro, A. Forster, M. Bonani, M. Dorigo, S. Magnenat, F. Mondada, R. O'Grady, C. Pinciroli, P. Retornaz, V. Trianni, and L. M. Gambardella, "Cooperative navigation in robotic swarms," *Swarm Intelligence*, vol. 8, no. 1, pp. 1–33, 2014.

[8] A. F. T. Winfield and J. Nembrini, "Safety in Numbers: Fault Tolerance in Robot Swarms," *International Journal on Modelling Identification and Control*, vol. 1, no. 1, pp. 30–37, 2006.

[9] Á. Gutiérrez, A. Campo, F. Monasterio-Huelin, L. Magdalena, and M. Dorigo, "Collective decision-making based on social odometry," *Neural Computing and Applications*, vol. 19, no. 6, pp. 807–823, 2010.

[10] M. J. B. Krieger and J.-B. Billeter, "The call of duty: Self-organised task allocation in a population of up to twelve mobile robots," *Robotics and Autonomous Systems*, vol. 30, no. 1-2, pp. 65–84, 2000.

[11] V. Gazi and K. M. Passino, "Stability analysis of social foraging swarms," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 539–557, 2004.

[12] Y. Liu and K. M. Passino, "Stable social foraging swarms in a noisy environment," *IEEE Transactions on Automatic Control*, vol. 49, no. 1, pp. 30–44, 2004.

[13] C. Ke and P. Dasgupta, "Dynamic area coverage using faulty multi-agent swarms," *Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2007)*, pp. 17–23, 2007.

[14] VICON, https://www.vicon.com [Accessed on 07 Feb 2019].

[15] W. Liu and A. F. T. Winfield, "Open-hardware e-puck Linux extension board for experimental swarm robotics research," *Microprocessors and Microsystems*, vol. 35, no. 1, pp. 60–67, 2011.

[16] L. Pitonakova, R. Crowder, and S. Bullock, "Behaviour-Data Relations Modelling Language for multi-robot control algorithms," in *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2017)*. Piscataway, NJ: IEEE Press, 2017, pp. 727–732.

[17] C. Amato, G. D. Konidaris, G. Cruz, C. A. Maynor, J. P. How, and L. P. Kaelbling, "Planning for decentralized control of multiple robots under uncertainty," in *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA 2015)*. IEEE Press, 2015, pp. 1241–1248.