Simple Swarm Foraging Algorithm Based on Gradient Computation

Simon O. Obute
School of Computing
University of Leeds
Leeds, UK, LS2 9JT
Email: scoo@leeds.ac.uk

Mehmet R. Dogar
School of Computing
University of Leeds
Leeds, UK, LS2 9JT
Email: M.R.Dogar@leeds.ac.uk

Jordan H. Boyle
Department of Mechanical Engineering
University of Leeds
Leeds, UK, LS2 9JT
Email: J.H.Boyle@leeds.ac.uk

Abstract—Swarm foraging is a common test case application for multi-robot systems. In this paper we present a novel algorithm for controlling swarm robots with limited communication range and storage capacity to efficiently search for and retrieve targets within an unknown environment. In our approach, robots search using random walk and adjust their turn probability based on attraction and repulsion signals they sense from other robots. We compared our algorithm with five different variations reflecting absence or presence of attractive and/or repulsive communication signals. Our results show that best performance is achieved when both signals are used by robots for communication. Furthermore, we show through hardware experiments how the communication model we used in the simulation could be realized on real robots.
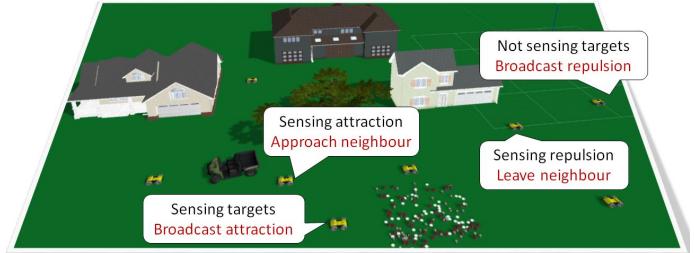
Fig. 1: In direct communication, robots exchange information with neighbours based on their local environment. The neighbouring robots use this information to adapt their search/foraging strategy. In the present work, communication is limited to selectively broadcasting simple attractive and respulsive signals.

I. Introduction

Swarm robotics applies swarm intelligence behaviours observed in natural swarms to multi-robot applications [3]. Swarms in nature have the impressive ability of accomplishing complex tasks through the interaction of simple agents with each other and/or their environments. For example, ants are able to forage food from distant locations, beyond the sensory capabilities of an individual swarm agent, by following pheromone trails which other ants have laid to connect food sources and the swarm nest. Natural swarms are made up of relatively simple individuals acting cooperatively to achieve swarm objectives in robust, flexible and scalable ways [4]. They serve as the main drive for research in swarm robotics. An individual agent in the group does not have access to global knowledge of the world and relies only on interaction with its immediate environment (and sometimes memory of previous experience) to make autonomous control decisions.

Foraging is a canonical test case for swarm robotics applications, which involves collective search and transport of objects (or food) to a specific deposit site (or nest) [22]. It integrates research in robotic exploration, navigation, object identification, manipulation and transport within a single relatively simple (in comparison to the task) robot platform. Foraging also has diverse real-world applications such as cleaning, harvesting, search and rescue, hazardous waste clean-up, landmine clearance and planetary exploration [18]. Swarm robot foraging, like other swarm robotics implementations, draws inspiration from biological parallels such as foraging approaches observed in ants and bees. A key means of achieving cooperation among swarm members during foraging is through communication. This has mostly been realized through a shared memory, communication through the environment or direct communication [3].

In shared memory implementations, all robots have access to a shared medium to write and read information, which gives swarm robots a global means of communication. In [1], robots that locate attractants (objects to forage) use the shared medium to notify all swarm members of the target's location. Yifan Cai and Yang [19] used the global knowledge of percentage of targets found and environment covered by the swarm to adapt the foraging strategy. Major drawbacks of this approach are issues related to scalability, increased complexity of individual robots and inconsistency with the swarm paradigm of decentralized control, local sensing and communication. A closely related approach is the use of a central nest as means of exchanging information among the swarm, where only robots within a limited range of the nest are able to communicate with it [7, 10]. However, this makes the central nest a single point of failure for the swarm.

Achieving cooperation using the environment as a medium of communication involves modification of the search space using "markers" or "beacons" that provide information to guide foraging robots. The work in [8] used robots to form stationary beacon networks that broadcast hop counts of their

proximity to the nest and target locations, thereby forming a gradient to be used by foraging robots to locate and transport targets to the nest. In [21], robots made use of the Cooperative-Color Marking Foraging Agents algorithm to mark out areas that have been covered and lay pheromone trails when returning to the nest if they successfully located a food source. When searching, robots were repulsed from explored areas and attracted to pheromone trails. The repulsive pheromone model was also used in [9]. In their work, robots also kept short-term memory of visited locations to prevent returning to recently explored sites. In [6], pre-deployed flying robots were used to form beacon networks by attaching to ceilings. They communicated instructions to foraging ground robots about paths connecting target locations to the nest. A major challenge for this communication model is finding an effective means of "marking" the environment without sacrificing some robots as stationary beacons, which would affect the scalability of the swarm.

In direct communication (Fig. 1), robots adapt their behaviour to improve foraging efficiency based on information exchanged with neighbouring robots. This information could be measuring a robot's position relative to their neighbours, which was used in [8] to maintain a formation of foraging robots that performed a circular sweep around the nest. Robots in [2] used their relative position to neighbours to avoid nearby robots as they searched for attractors (targets) to forage. In [15], robots used light on top of their bodies to attract neighbouring robots when they located targets. In [5], robots communicated their foraging success to neighbouring robots they encountered in the environment. This information was used to adjust the foraging threshold of a robot, which represented its likelihood of going out to search for food whenever it returned to the nest. In [14], robots mimicked the foraging behaviour of honeybees. Robots returning from scouting for targets communicated the target locations to other robots at the nest. The listening robots probabilistically decided whether to forage from the advertised food source. Robots exchanged foraging success and neurocontroller weights with other robots they encountered in [13]. They then used this information to adapt their neurocontroller weights to improve foraging efficiency. Direct communication approaches overcome the scalability and complexity issues of shared memory, and the implementation challenges of environmental communication. However, there are challenges regarding what type of information should be exchanged, handling interactions with multiple neighbours simultaneously and robustness/reliability of communication media. Our approach shows that analogue sound signals can be used as a simple but effective means of direct communication among a swarm of foraging robots.

The biological foundation for our approach is the chemotaxis behaviour observed in micro-organisms such as the *Escherichia coli* bacterium [12] and *Caenorhabditis elegans* nematode [17], whose motion are characterized by near-linear runs ('swimming' mode) with occasional turns ('turning' mode) that randomize the direction of the next run. The probability that an individual *E. coli* or *C. elegans* will perform
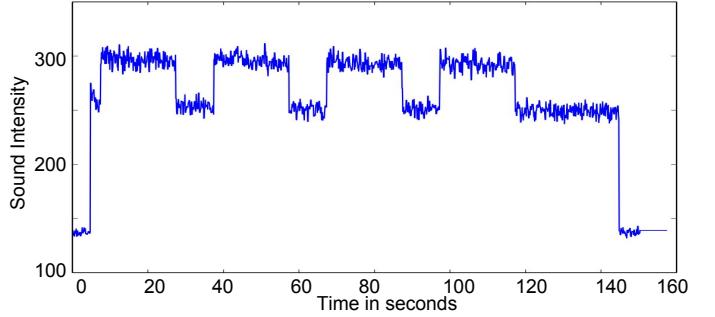

Fig. 2: Simple experiment using white noise from two speakers that demonstrates increased intensity when multiple sound sources are active (appearing as intermittent peaks in the plot).

a turn at any given moment depends on the temporal concentration gradient of a sensed chemical attractant (or repellent) in the environment. If conditions are improving (positive attractant gradient or negative repellant gradient) the organism reduces its turn probability, whereas worsening conditions lead to an increased probability. The simple but elegant approach of responding to the change over time means that a 'single pixel' non-directional analogue sensor is sufficient.

In our proposed swarm algorithm, the robots themselves serve as signal sources that create 'concentration gradients' (repellent and attractant) which are used by neighbouring swarm members to improve their target search. Furthermore, the robots sense and/or broadcast these signals selectively depending on which state they are in, thereby creating a dynamic sensory landscape. These properties make our approach distinct from the biological foundations of our algorithm and its previous robotic impmlementations [11]. While it is primarily a direct communication method, this approach also shares some properties with environmental communication in that it exploits natural properties of the environment. By using sound as the communication signal, gradient formation and superposition of multiple signal sources happens 'for free' through environmental physics (as shown in Fig. 2).

Our approach emphasizes extreme algorithmic simplicity and demonstrates the power of minimalist bio-inspired navigation algorithms. By foregoing complex communication systems , the algorithm lends itself to simple, low cost hardware implementations. To the best of our knowledge, this is the first time this approach has been applied to swarm foraging.

We present this repulsion-attraction (Rep-Att) algorithm in Section II and simulation results based on an idealised communication model in Section III. In Section IV we present our work on implementing the communication model in hardware platforms, and demonstrate that the algorithm still works with a realistic and noisy sound model. We also validate the basic 'chemotaxis' behaviour on a real robot. Finally, concluding remarks and future directions are presented in Section V.

II. REPULSION-ATTRACTION ALGORITHM (REP-ATT)

Rep-Att is based on the use of a communication mechanism whose intensity decreases smoothly with increasing distance

as represented by Equation 1, where A_{ij}^k is signal strength of type k sensed by robot i, located d_{ij} metres away from signal source j and C_{max} is a robot's maximum communication range. Total signal strength sensed by a robot at a particular time, $I_i^k(t)$, at any location in the world is the sum of all signals of the same type at that location (Equation 2), where n is the total number of robots and k is the signal type. We consider only two signal types that robots can sense and broadcast - repulsion ($k = 0$) and attraction ($k = 1$) signals - within a limited range (C_{max}). The robots then sense gradients of individual signals (ΔI_i^k) by computing the difference in signal intensity between two time steps (Equation 3) to use as a tool for improving swarm foraging efficiency. The influence of these signals on a robot depends on whether it is in the searching, acquiring, homing or obstacle avoidance states.

$$A_{ij}^k = \begin{cases} \dfrac{C_{max} - d_{ij}}{C_{max}} & \text{if} \quad d_{ij} \leq C_{max} \\[2ex] 0, & \text{Otherwise} \end{cases} \quad (1)$$

$$I_i^k(t) = \sum_{j=1, j \neq i}^{n} A_{ij}^k \quad (2)$$

$$\Delta I_i^k = I_i^k(t) - I_i^k(t-1) \quad (3)$$

Searching State is when a robot does not sense any target (item to be foraged) within its visual range (V_r). It broadcasts a repulsion signal (A^0) to its neighbours while using a random walk (runs of forwards motion interspersed by random turns) to search for targets. Its goal in this state is to minimize the repulsion (I^0) and maximize the attraction (I^1) signals it senses. This is achieved by detecting the change in intensity of these signals between two time steps. A robot increases its turning probability when moving in the wrong direction, i.e. when $\Delta I^0 > 0$ or $\Delta I^1 < 0$. Doing this increases a robot's likelihood of reorienting itself in the desired direction. On the other hand, when the robot senses a positive gradient for attraction ($\Delta I^1 > 0$) or a negative repulsion gradient ($\Delta I^0 < 0$), it reduces its turning probability, which in turn helps the robot to maintain its current direction for a longer period of time and consequently approach a region that increases its likelihood of finding a target or exploring regions with less searching robots.

Acquiring State is when a robot detects target(s) within its visual range (V_r). It switches from searching to the acquiring state, ignores all communication signals and collects the nearest target. In cases where the robot detects multiple targets exceeding its carrying capacity, it broadcasts a range-limited attraction signal (A^1), which searching robots within communication range can sense to adjust their turning probabilities accordingly.

Homing State is used when the robot's capacity is full. In this state, the robot heads directly towards the central deposit site (it is assumed that the direction is known) in order to deposit the collected targets. In this state, the robot ignores
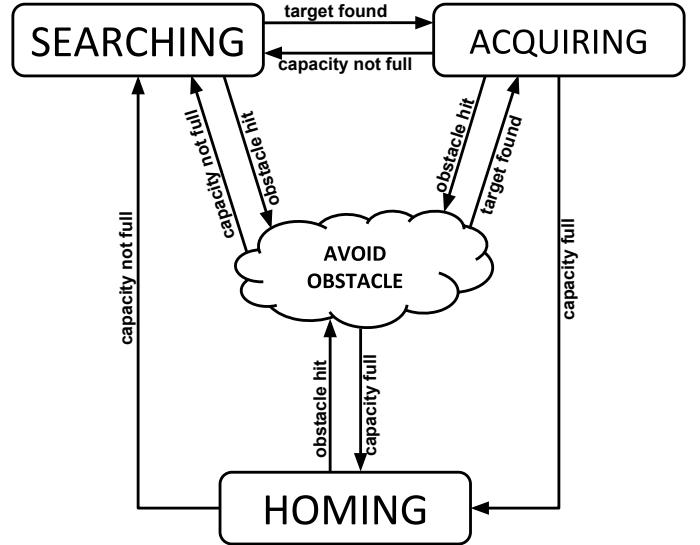

Fig. 3: Finite state machine representation of foraging task, with the consideration of obstacle avoidance behaviour.

attraction and repulsion signals from nearby robots until it has successfully offloaded all foraged targets at the deposit site after which it switches to the searching state.

Obstacle Avoidance State is used by robots to avoid static (nest and walls) and dynamic (other robots) obstacles. When a robot bumps into an obstacle, it avoids it by turning away from the obstacle and making a random linear motion between 0 and 1m. The robot then switches to either the searching, acquiring or homing state, depending on its local environment and availability of space to carry more targets. The interaction between robot states is shown in Fig. 3, while Algorithm 1 illustrates the swarm foraging algorithm described in this section in psuedo code.

From Algorithm 1, five variants can be derived based on the values of the $repel$, $attract$ and $repeller$ Boolean variables:

I. **Random Walk:** In this algorithm, robots do not listen to repulsion and attraction signals. They search with a constant turning probability and view other swarm robots as obstacles upon colliding with them. Thus, $repel = false$, $attract = false$ and $repeller = false$.

II. **Repeller:** This is based on the functionality of the algorithm presented in [2], where robots that cannot see targets are constantly repelled by their neighbours (regardless of what the neighbours sense). To achieve this behaviour from Algorithm 1, $repel$ and $repeller$ are set to $true$ while attract is set to $false$.

III. **Selective Repulsion:** Here, robots communicate repulsion signals when in the searching state and refrain from communication in other states, which is different from Repeller algorithm that repels in all states. The $repel$ variable is $true$, while $attract$ and $repeller$ are both $false$.

IV. **Selective Attraction:** For this variant, $repel$ and $repeller$ are set to $false$ while $attract$ is set to $true$.

Algorithm 1 Swarm Foraging Algorithm
1: Initialize Algorithm
2: **while true do**
3:   **if** capacity is full **then**
4:     Go home and drop collected targets
5:   **else**
6:     $P_t = P_b$
7:     **if** $repel$ **then**
8:       **if** targets seen $== 0$ **or** $repeller$ **then**
9:         Broadcast Repulsion A_i^0
10:      **if** $\Delta I_i^0 < 0$ **then**
11:        $P_t = P_b \div G_r$
12:      **else if** $\Delta I_i^0 > 0$ **then**
13:        $P_t = P_b \times G_r$
14:     **if** $attract$ **then**
15:       **if** targets seen $> c_m$ **then**
16:         Broadcast Attraction A_i^1
17:       **if** $\Delta I_i^1 < 0$ **then**
18:         $P_t = P_b \times G_a$
19:       **else if** $\Delta I_i^1 > 0$ **then**
20:         $P_t = P_b \div G_a$
21:     **if** targets seen > 0 **then**
22:       Go and pick up closest target
23:     **else**
24:       Random walk using P_t as probability of turn

This means robots do not repel each other, but attract neighbours when they locate more targets than they can carry.

V. **Rep-Att:** This is the full version of our algorithm, in which the selective attraction and selective repulsion communication mechanisms are active. $repel = true$, $attract = true$ and $repeller = false$. It has been described in detail at the beginning of this section.

III. SIMULATION AND RESULTS (IDEALISED COMMUNICATION)

A. Simulation Setup

The Gazebo Simulator [16], with maximum step size set to 0.025 second, was used for the simulation experiments to compare the effects of presence/absence of repulsion and/or attraction signals as described in Section II.

To thoroughly investigate the performance of the algorithms, five different distributions were used to initialize locations of 200 targets within a 50m x 50m world. The target distributions are one, two or four clusters, a uniform distribution, and a uniform distribution overlayed with a single cluster (HalfCluster in the figures). The goal in each world scenario is for 36 robots having limited carrying capacity and sensing range to locate the targets and return at least 90% of them to the central deposit site (nest). Each robot in the swarm has a velocity of 0.605 m/s and spends 5 seconds processing each target

TABLE I: Parameters used in simulation and robot experiments. The same values were used for all algorithms (where the algorithm makes use of the parameter) except stated otherwise in the text.

Parameter	Symbol	Value
Base turning probability per time step	P_b	0.0025
Attraction Gain	G_a	10
Repulsion Gain	G_r	10
Robot targets-carrying capacity	c_m	5
Maximum communication range	C_r	15m
Target detection distance	V_r	3m
Random turn distribution		$\mathcal{N}(180^0, 90^0)$

it touches (i.e. to remove a target from the ground). Table I outlines other parameters for the simulation experiments.

B. Results

Simulation snapshots for Rep-Att algorithm for the Two Clusters target distribution are shown in Fig. 4 (supplementary videos are also provided). Here, we see excellent collaboration among swarm members when being controlled with the Rep-Att algorithm. Repulsion signals at the start of the simulation causes the robots to quickly leave the nest in search of targets. When robots find target clusters, they activate attraction signal to invite searching robots to cluster regions for increased exploitation of that area.

The simulation results for the five target distributions are summarized in Fig. 5 as the average time (normalized with the time taken by Random Walk for that distribution) taken by the swarm to successfully return 90% of targets to the nest. Each experiment was repeated 30 times. We have also included results for two unrealistic, but superior, foraging algorithms for benchmarking purposes: Global Detector and Idealized Foraging. Global Detector is when robots in the swarm have infinite target detection range i.e. $V_r = \infty$, effectively eliminating the searching state described in Section II. In Idealized Foraging, we apply a statistical approach to estimate the minimum possible collection time T_{min} assuming perfect task allocation and ignoring robot collisions, according to Equation 4

$$T_{min} = \frac{1}{N_R} \times \left[\frac{1}{V_R} \times \left(\frac{N_T}{R_C} \times 2\overline{D_{TB}} + \right. \right.$$
$$\left. \left. (R_C - 1) \times \overline{D_{TN}} \right) + N_T \times T_P \right] \tag{4}$$

Where N_R = swarm size, V_R = robot velocity, N_T = number of targets, R_C = robot capacity, $\overline{D_{TB}}$ = mean distance of all targets from nest, $\overline{D_{TN}}$ = mean distance between a target and nearest $R_C - 1$ targets, and T_P = target processing time.

The bar plot of Fig. 5 shows that Rep-Att algorithm outperforms both Random Walk and Repeller algorithms on all
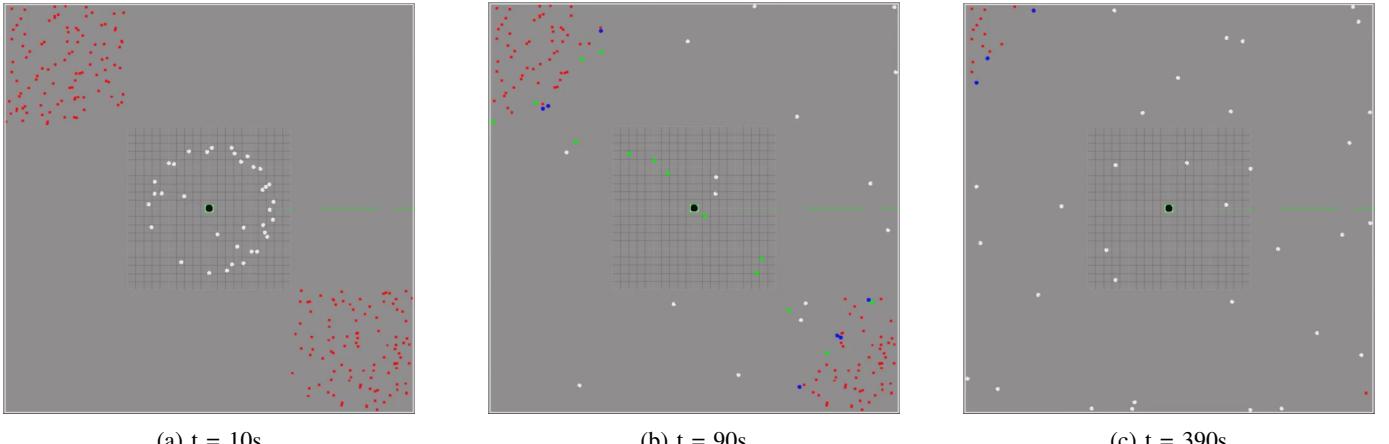
(a) t = 10s   (b) t = 90s   (c) t = 390s

Fig. 4: Simulation snapshots of Rep-Att when foraging in Two Clusters target distribution. Targets are represented in red, searching, acquiring and homing robots are white, blue and green respectively, while central nest is black. (Best viewed in colour)
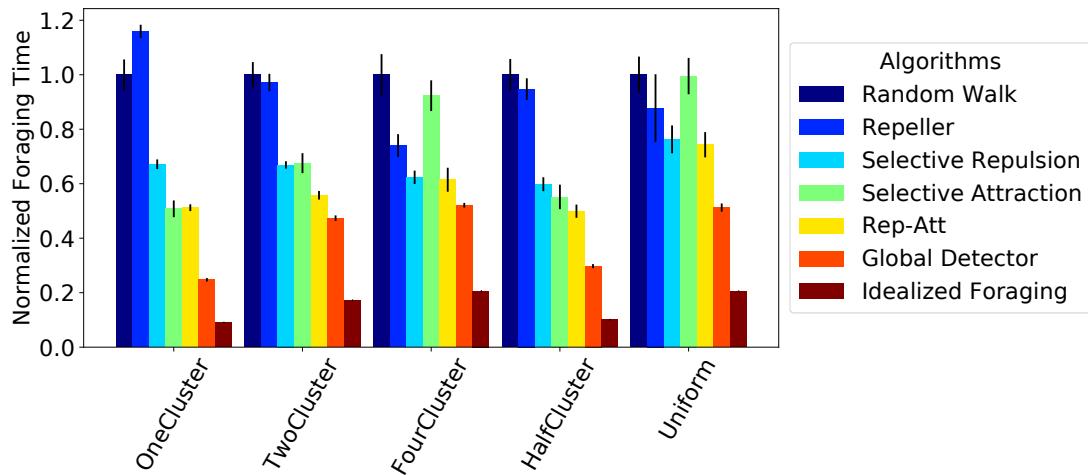

Fig. 5: Normalized time taken to return 90% of targets to nest in 5 different world scenarios. Each bar represents the mean of 30 simulation repetitions. The error bars represent 95% confidence interval. Random walk foraging time was used as baseline for computing normalized times for the respective target distributions

world scenarios, giving statistically significant lower foraging times in all cases. T-test analysis conducted between Rep-Att and these two algorithms yielded a maximum p-value of 0.028 on the uniform target distribution world, while other world scenarios gave lower p-values. This shows that our Rep-Att algorithm is able to significantly improve foraging efficiency by simple adaptation of robot turning probability based on repulsion and attraction gradients sensed by individual robots.

Fig. 5 also investigates the individual effects of attraction and repulsion signals under the different target distributions. What we see is that repulsion is most useful for the swarm when targets are less clustered in the environment. This is most evident in the four clusters and uniform targets distributions. Attraction on the other hand is most effective when resources/targets are densely clustered within few regions in the world as is evident in the one cluster case. Our Rep-

Att algorithm is, therefore, able to forage efficiently across all the simulated target distributions because it integrates the contrasting strengths of repulsion and attraction within a single algorithm.

In comparison to the unrealistic Global Detector and Idealized Foraging algorithms, Rep-Att, unsurprisingly, performs less well. However, it is interesting to note that Rep-Att is somewhat competitive when compared to the Global Detector in the Two Clusters and Four Clusters target distributions, where Rep-Att only took 17.77% and 17.96% more time respectively. This shows that when an environment has multiple clustered targets at various locations, repulsive signals will quickly drive multiple robots to locate target clusters. Other robots are then more likely to come within communication range of one or more attracting robot(s), thereby reducing the time they would have spent searching for targets. The

difference in performance between Rep-Att and the unrealistic algorithms is more pronounced in other world scenarios because: robots spend more time searching for targets (One Cluster and HalfCluster distributions case); and robots did not sense enough targets to attract neighbours to their foraging locations (Uniform distribution case).

IV. SWARM COMMUNICATION EXPERIMENTS AND RESULTS (REALISTIC COMMUNICATION)

A. Communication Modelling

The successful deployment of the Rep-Att algorithm on real robots hinges on development of suitable communication hardware, a good visual system for detecting targets and a number of navigation, transport and manipulation challenges. We show in this section our progress in realising the communication mechanism for the robots on hardware. We show that gradient computation can be realized on real robot platforms using sound, and that the algorithm continues to work well with a realistic, noisy sound model. The first step involves collection of sound data using a robot equipped with an omnidirectional microphone to measure sound intensity from a sound source (Fig. 6a). In our experiment, the robot started 15m away from a speaker broadcasting white noise signal and was made to move toward the speaker at a speed of 0.1m/s. The robot logged distance travelled (using odometry) and sound intensity as it journeyed toward the speaker. This experiment was repeated five times in order to collect reliable data for developing the model equation.

In the second step, the sound data was used to model its degradation as distance of propagation increases, represented as Equation 5 [20]. The A_e term was added to account for the mean ambient noise in the environment. A_0 is the sound intensity from the source, α is the attenuation factor of the environment and d is the distance between the sound source and microphone. The parameters of Equation 5 were computed by finding the least square fit between the collected data and the model equation using a simple MATLAB script. The computed parameters were $A_0 = 140.5193$, $\alpha = 0.1193$ and $A_e = 48.1824$.

$$A_{ij}^k = A_0 e^{-\alpha d_{ij}} + A_e \qquad (5)$$

The third step involved quantifying the noise in the recorded sound signal (shown in Fig. 6b). To do this, the logged data were broken into 1m segments; a line was fitted to each segment; and the deviation of recorded data on each segment from the fitted line was computed. We observed a direct proportional relationship between sound intensity and the computed deviation. However, there was minimal variation in the ratio of the deviation to mean intensity for each segment (Fig. 6c). Thus, we modelled the noise in the signal as the mean of this ratio (0.06 of sound intensity). The final sound intensity model that takes into consideration the noise is as shown in Equation 6, where $A_{pure} = A_{ij}^k$ computed using Equation 5 and $\mathcal{N}(0, 0.06)$ is a normal distribution with mean
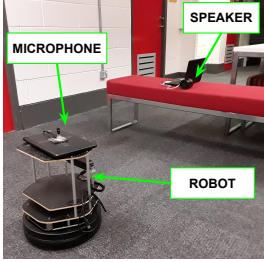
of 0 and deviation of 0.06.

$$A_{noisy} = A_{pure}(1 - \mathcal{N}(0, 0.06)) \qquad (6)$$

After obtaining this realistic sound model we introduced it to the simulator in place of Equation 1 and repeated the original simulations. This had an extremely detrimental effect as can be seen by comparing Rep-Att-Q1 to Random Walk in Fig. 7. Indeed, the Rep-Att algorithm was now five times slower than random walk. We examined the sensory data from these simulations (not shown) and arrived at the following explanation. With a simulation update frequency of 40Hz, the change in the underlying sound intensity from timestep to timestep is very small compared to the magnitude of the noise. As such, the chances that the signal will have increased or decreased is very nearly 50:50 regardless of the underlying signal. Not only does this mask any useful gradient information, but it also leads to a significant increase in overall turn probability because the effect of multiplying and dividing by $G_{a/r}$ is not symmetric in terms of the expected number of turns.
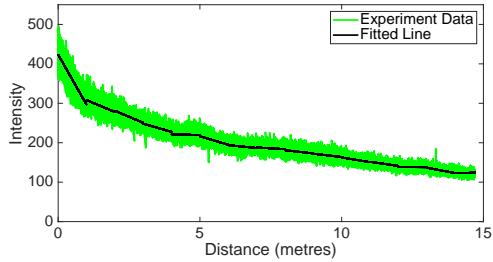
To mitigate the effect of noise on the Rep-Att algorithm, we introduced an averaging filter to be used by the robots when sensing attraction and repulsion signals. This simple filter computes the average of a queue of multiple instantaneous signal value measurements and the result is used to compute gradient information by comparing it to the mean of the previous queue. The effect of this filter in a foraging scenario is shown in Fig. 7 for the two clusters target distribution, comparing the random walk, noiseless communication signal and noisy communication signal with queue from 1 to 80. This shows that the filter is effective, and an average of 40 samples gives best foraging performance.

The realistic communication model with an averaging filter of queue size 40 was tested on the five target distributions investigated in Section III-B. Figure 8a shows the relative foraging performance of the Rep-Att algorithm with noiseless (queue size 1) and noisy (queue size 40) communication, compared to the baseline Random Walk algorithm. The noisy communication resulted in a maximum of 39.30% decrease in performance when compared with the noiseless communication (this was observed in the one cluster distribution). Notwithstanding, Rep-Att still outperforms the baseline algorithm with a minimum 10.38% improvement in foraging time (based on comparisons on uniform target distribution), with the maximum improvement being 52.27% (based on comparisons on one cluster target distribution).
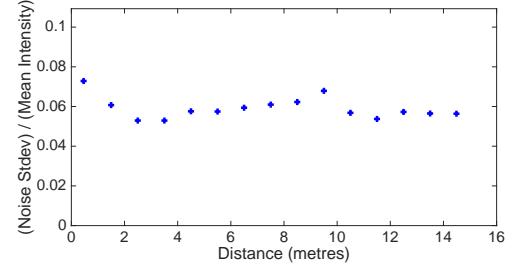
A final test on the algorithm's performance was conducted on a larger world, with 100m x 100m dimension. The target locations remained unchanged from the 50m by 50m worlds, so there was an area of empty space between the targets and the bounding wall. The bar chart in Fig. 8b shows the normalized bar plots. It is evident that with increase in world dimensions, Rep-Att's superiority over the baseline increased substantially in all cases except the uniform world. Surprisingly, the version with noisy communication and a queue size of 40 outperforms

(a) Experiment setup for collecting sound data

(b) Modelling sound degradation with distance

(c) Ratio of deviation to mean intensity within each 1-metre segment of line fit

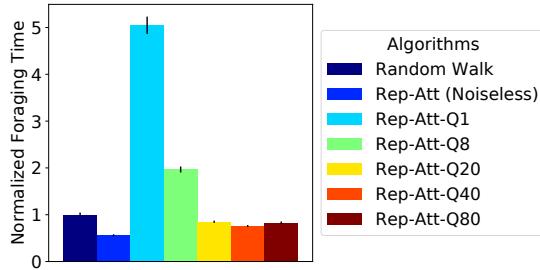Fig. 6: Modelling the degradation of sound with distance

Fig. 7: Investigating the effect of average filter on the performance of Rep-Att using realistic noise values. The filtering window was increased in steps from 1 to 80 to visualize its effect.

the noiseless verions with queue size 1 in the four cluster and uniform environments. The relative contributions of the noise and longer queue size to this phenomenon are unclear and will be investigated in future.

B. Communication Validation Experiment

To validate that our communication model is a valid representation of the real hardware we initially characterised, we set up an experiment with a single Turtlebot2 in a 5m x 3m arena. The robot had a omnidirectional microphone and was programmed to use the attraction behaviour from our algorithm to search for a single sound source. The setup in Fig. 9 was used to reflect the sound from a speaker in all directions, effectively making it omnidirectional. To the best of our knowledge, this is the first such experiments to be conducted in real hardware using bio-inspired temporal gradient computation from a sound signal.

The robot velocity used was 0.1 m/s and a filter queue size of 40 samples was used. Other parameters were left unchanged (Table I). The sound source was kept at (-1.5,0), while the Turtlebot2 starting location was (0,0) facing away from the sound source. The Turtlebot2's odometry was used to keep track of where it is in the arena. Each experiment lasted for 1000 seconds and was repeated 10 times. We investigate our algorithm's ability to find the sound source and provide results for pure random walk, as a baseline. We conducted equivalent experiments in simulation and using the real hardware.
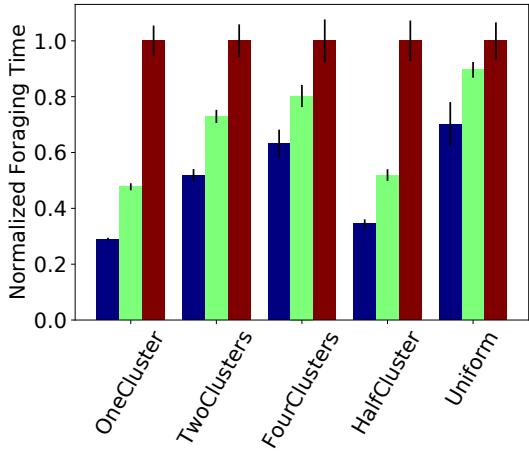
Fig. 10 shows snapshots of the experiment, where the Turtlebot2, when controlled with Rep-Att, was able to find the sound source and remain within that region for the remainder of the experiment's duration.

In Quantitative terms, Fig. 11 shows the proportion of the experiment time a robot spent in regions around the sound source. Each real robot experiment was repeated 10 times, while simulation experiments were repeated 30 times. The results show that both real and simulated robots effectively navigated to the sound source when using the taxis (Rep-Att) behaviour and spent significantly more time within 1m of the sound source in comparison to random walk. The relatively good agreement between simulated and physical experiments suggests that the sound model we used is a meaningful reflection of real hardware. The noticeable difference between simulation and hardware experiment times for both algorithms is due to slight differences between the two platform implementations. For example, in the hardware experiment, the Turtlebot2's obstacle avoidance behaviour entailed its reversing for 10cm before turning a random angle, which meant it spent more time within the obstacle's region (wall or sound source) when compared to the time taken by the simulated robot (see obstacle avoidance description in Section II)
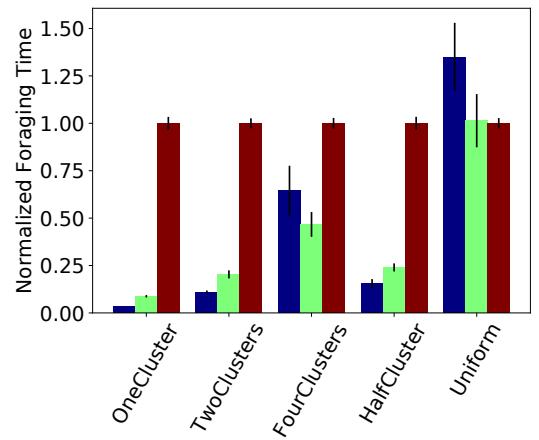
V. CONCLUSION

We have presented Rep-Att, a simple yet effective, swarm foraging algorithm that works surprisingly well - even when given a realistic and far from ideal communication model strongly grounded in real hardware experiment. Rep-Att is based on a concept of selective broadcast of repulsion and attraction signals among swarm agents, which was used to adapt robot's turning probability while searching. The end result was a significant improvement in the swarm's foraging efficiency. In addition to extensive simulation studies on how our algorithm performs under different target distributions, we have also validated our communication model by comparing hardware to simulated results for a single robot.

Future work will involve a full demonstration of our foraging algorithm on a swarm of real robots. We also intend to extend the capability of our robots to use odometry in conjunction with memory of previous places targets were last

(a) 50m x 50m worlds

(b) 100m x 100m worlds

Fig. 8: Time taken in seconds to return 90% of targets to nest in 5 different world scenarios. Each bar represents the mean of 30 simulation repetitions. The error bars represent 95% confidence interval.
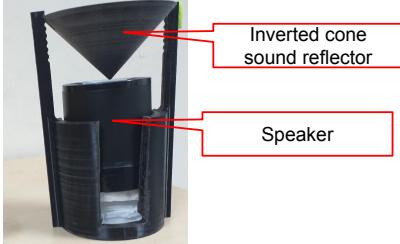

Fig. 9: 3D-printed setup for makes use of an inverted cone to reflect the sound signals omni-directionally round the speaker.

found so that they can weigh the options between returning to last foraged locations or solely search using repulsion-attraction signals. Investigation of the effects of obstacles within the foraging environment will also be looked into.

REFERENCES

[1] R.C. Arkin, T. Balch, and E. Nitz. Communication of behavorial state in multi-agent retrieval tasks. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 588–594. IEEE Comput. Soc. Press, 1993. ISBN 0-8186-3450-2. doi: 10.1109/ROBOT. 1993.291841. URL http://ieeexplore.ieee.org/document/291841/.

[2] Ronald C. Arkin. Cooperation without Communication: Multi-Agent Schema-Based Robot Navigation. *Journal of Robotic Systems*, 9(3):351–364, 1992. doi: 10. 1002/rob.4620090304. URL https://www.cc.gatech.edu/ai/robot-lab/online-publications/ieee/coop92.pdf.

[3] Levent Bayindir. A review of swarm robotics tasks. *Neurocomputing*, 172:292–321, 2016. ISSN 18728286. doi: 10.1016/j.neucom.2015.05.116.

[4] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: A review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 2013. ISSN 19353812. doi: 10.1007/s11721-012-0075-2.

[5] Hanyi Dai. *Adaptive Control in Swarm Robotic Systems*, volume 3. Western Michigan University, Graduate Student Advisory Committee, sep 2009. URL http://scholarworks.wmich.edu/hilltopreview/vol3/iss1/7.

[6] Frederick Ducatelle, Gianni A Di Caro, Carlo Pinciroli, Luca M Gambardella, F Ducatelle, Ga Di Caro, LM Gambardella, and C Pinciroli. Self-organized cooperation between robotic swarms. *Swarm Intell*, 5: 73–96, 2011. doi: 10.1007/s11721-011-0053-0. URL http://www.swarmanoid.org.

[7] Joshua P. Hecker, Justin Craig Carmichael, and Melanie E. Moses. Exploiting clusters for complete resource collection in biologically-inspired robot swarms. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 434–440. IEEE, sep 2015. ISBN 978-1-4799-9994-1. doi: 10.1109/IROS. 2015.7353409. URL http://ieeexplore.ieee.org/document/7353409/.

[8] Nicholas Hoff, Robert Wood, and Radhika Nagpal. Distributed colony-level algorithm switching for robot swarm foraging. In *Springer Tracts in Advanced Robotics*, volume 83 STAR, pages 417–430. Springer Berlin Heidelberg, 2012. ISBN 9783642327223. doi: 10.1007/978-3-642-32723-0_30.

[9] Danielli A Lima and Gina MB Oliveira. A cellular automata ant memory model of foraging in a swarm of robots. *Applied Mathematical Modelling*, 47:551–572, 2017.

[10] Qi Lu, Joshua P. Hecker, and Melanie E. Moses. Multiple-place swarm foraging with dynamic depots. *Autonomous Robots*, 42(4):909–926, 2018. ISSN 15737527. doi: 10.1007/s10514-017-9693-2.

[11] Surya G. Nurzaman, Yoshio Matsumoto, Yutaka Naka-

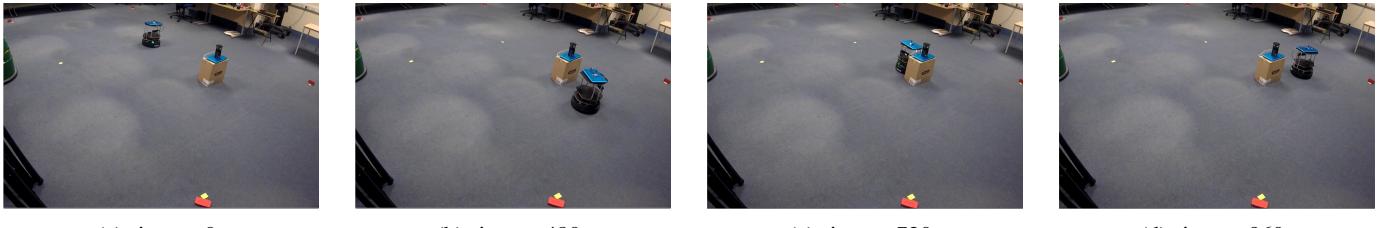| (a) time = 0s | (b) time = 480s | (c) time = 720s | (d) time = 960s |

Fig. 10: Experiment with a Turtlebot2 using a single microphone and performing chemotaxis to be attracted to white noise from an omnidirectional speaker. The turtlebot is restricted to search within a 5m by 3m virtual wall (enabled by odometry). Footage of the full experiment is available in supplementary video file.
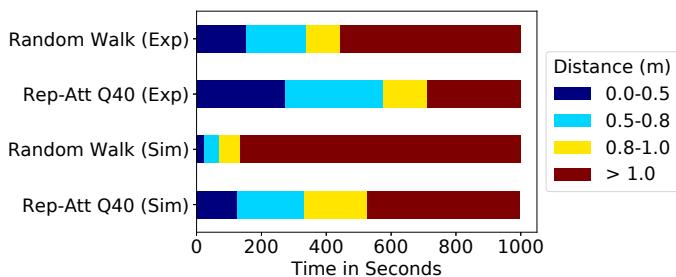

Fig. 11: Proportion of 1000 seconds experiment time spent by Turtlebot2 relative to the sound source. Please note that robot bumps into the sound source setup when it is within at least 0.27m away from it. *Exp* represent data gotten from Turtlebot2 experiment, *Sim* represents data gotten from equivalent Gazebo simulation, *Rep-Att Q40* represents noisy communication with window/queue size of 40 samples.

mura, Satoshi Koizumi, and Hiroshi Ishiguro. Biologically inspired adaptive mobile robot search with and without gradient sensing. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pages 142–147, St. Louis, USA, 2009. ISBN 9781424438044. doi: 10.1109/IROS.2009.5353998.

[12] Surya G. Nurzaman, Yoshio Matsumoto, Yutaka Nakamura, Satoshi Koizumi, and Hiroshi Ishiguro. 'Yuragi'-based adaptive mobile robot search with and without gradient sensing: From bacterial chemotaxis to a Levy walk. *Advanced Robotics*, 25(16):2019–2037, 2011. ISSN 01691864. doi: 10.1163/016918611X590229.

[13] Iñaki Fernández Pérez, Amine Boumaza, and François Charpillet. Learning Collaborative Foraging in a Swarm of Robots using Embodied Evolution. In *ECAL 2017 14th European Conference on Artificial Life*, Lyon, sep 2017. Inria. URL https://hal.archives-ouvertes.fr/hal-01534242/.

[14] Lenka Pitonakova, Richard Crowder, and Seth Bullock. Information flow principles for plasticity in foraging robot swarms. *Swarm Intelligence*, 10(1): 33–63, mar 2016. ISSN 1935-3812. doi: 10.1007/s11721-016-0118-1. URL http://link.springer.com/10.1007/s11721-016-0118-1.

[15] Paul E Rybski, Amy Larson, Harini Veeraraghavan, Monica Anderson, and Maria Gini. Performance Evaluation of a Multi-Robot Search & Retrieval System: Experiences with MinDART. *J Intell Robot Syst*, 52:363–387, 2008. doi: 10.1007/s10846-008-9222-9.

[16] Ying Tan and Zhong-yang Zheng. Research Advance in Swarm Robotics. *Defence Technology*, 9(1):18–39, 2013. ISSN 22149147. doi: 10.1016/j.dt.2013.03. 001. URL http://www.sciencedirect.com/science/article/pii/S221491471300024X.

[17] Samuel Ward. Chemotaxis by the nematode Caenorhabditis elegans: identification of attractants and analysis of the response by use of mutants. *Proceedings of the National Academy of Sciences of the United States of America*, 70(3):817–21, 1973. ISSN 0027-8424. doi: 10.1073/pnas.70.3.817. URL http://www.pnas.org/content/pnas/70/3/817.full.pdf/pmc/articles/PMC433366/?report=abstract.

[18] Alan F T Winfield. Towards an engineering science of robot foraging. *Distributed Autonomous Robotic Systems 8*, (3):185–192, 2009. doi: 10.1007/978-3-642-00644-9-16.

[19] Yifan Cai and Simon X. Yang. A PSO-based approach to cooperative foraging tasks of multi-robots in completely unknown environments. In *2014 World Automation Congress (WAC)*, pages 813–822. IEEE, aug 2014. ISBN 978-1-8893-3549-0. doi: 10.1109/WAC.2014.6936157. URL http://ieeexplore.ieee.org/document/6936157/.

[20] Peng Yu, Rua Yan, and Li Yao. Measurement of Acoustic Attenuation Coefficient of Stored Grain. In *3rd International Conference on Control, Automation and Robotics*, pages 551–554, 2017. ISBN 9781509060887.

[21] Ouarda Zedadra, Hamid Seridi, Nicolas Jouandeau, and Giancarlo Fortino. An energy-aware algorithm for large scale foraging systems. *Scalable Computing*, 16(4):449–466, 2015. ISSN 18951767. doi: 10.12694/scpe.v16i4.1133.

[22] Ouarda Zedadra, Nicolas Jouandeau, Hamid Seridi, and Giancarlo Fortino. Multi-Agent Foraging: state-of-the-art and research challenges. *Complex Adaptive Systems Modeling*, 5(1):3, 2017. ISSN 2194-3206. doi: 10.1186/s40294-016-0041-8.