Adaptive Transmission Range for Decentralised Foraging Robots Using Autonomic Pulse Communications

Liam McGuigan, Roy Sterritt, Glenn Hawe
School of Computing, Faculty of Computing, Engineering and the Built Environment
Ulster University
Jordanstown, N. Ireland
email: mcguigan-l8@ulster.ac.uk, r.sterritt@ulster.ac.uk, gi.hawe@ulster.ac.uk

***Abstract*— A robot swarm which is to be deployed without the need for regular human input is required to be autonomous, capable of the self-management needed for operation in distant, complex, or changing environments. Communication between the individual robots is an essential facet of the swarm's ability to cooperate and adapt, and use of a fixed transmission range may result in issues with connectivity, inefficiency, or lead to constraints on robot movement. In this research, an Autonomic Pulse Communications system is developed for a simulated robot swarm, adaptively selecting a suitable transmission range based on local measurements of swarm density. The system is able to successfully share data around the swarm within a fixed time period, even with low density swarms and with a high robustness to communications loss. Further, the APC system is used in a simulated foraging task, performing as well as a previous decentralised autonomic system, but without the need for prior selection of a suitable transmission range.**

Keywords- Swarm robotics; Self-adaptation; Autonomic Computing; Swarm communication; Simulation.

I. INTRODUCTION

This paper is an extended version of the work published in [1], extending those results and presenting further research.

Swarm robotics, the study of how individual behaviours within a group of robots may combine through local interactions to create a more complex set of behaviours [2], has potential applications in fields such as space exploration [3], precision agriculture [4], and disaster response [5], where many small, simple robots can cover a much larger area than a single monolithic craft.

The size of the swarm, its decentralised nature, and the conditions in which it may potentially operate mean that a swarm should be able to act on its own, adjusting its behaviour according to a changing situation without the need for any external guidance [6]. Autonomic Computing concepts [7][8] can assist in achieving swarm self-adaptation, making use of a Monitor, Analyse, Plan and Execute loop, with a shared Knowledge base, known as MAPE-K, as described in [7] to assess the situation, identify any changes necessary, and implement them.

As swarms are decentralised, their ability to adapt depends on their cooperation through sharing information on which to base decisions and come to an agreement on actions to be taken. When the swarms are reliant on local communication with neighbouring robots, the effective range of that communication matters. Too small, and robot behaviour may need to be constrained to maintain communication links with other members of the swarm. Too large, and it may be an inefficient use of battery power, lead to communication interference, or even be detrimental to overall performance.

In previous work, a decentralised swarm made use of an autonomic system to help adjust a range over which robots would broadcast for help in a foraging task [9]. This worked by using a fixed range pulse message between robots to help estimate the density, but it was found that the range of this pulse message needed to be set for differing swarm densities. If this is not initially known, performance would be degraded.

The objective of this work is to implement an adaptive system for setting the range over which a robot broadcasts information, according to the local density of the swarm, detected at run-time. This will then be used in a simulation of foraging robots to resolve the requirement for a pre-set pulse range.

The rest of this paper is structured as follows. Section II discusses related work in swarm self-adaptation and autonomic systems used to develop the Autonomic Pulse Communication (APC) system presented. Section III discusses the design of the APC system and how it estimates local density in order to determine a suitable broadcast range. Section IV describes the data sharing task designed to test the APC's ability to maintain communication in the swarm, Section V introduces the test scenarios used, and Section VI presents the results. Section VII puts the APC system to work in a simulation of foraging robots, comparing the results against the performance of the previous decentralized system. Section VIII discusses the results, and Section IX concludes the paper with a summary, and directions for future research.

II. RELATED WORK

In the context of a robot swarm, a distinction can be made between the adaptation of individual robots, and that of the swarm as a whole. This can be related to the idea of *self-expression* [10][11], in which the swarm at large can be reconfigured. Such swarm-level adaptation can then take advantage of wider knowledge to make changes to swarm composition [12], or cooperative strategies [13].

To achieve swarm-level adaptation, however, cooperation and communication becomes essential. Individuals must share data in order to collectively recognize the need to adapt, and then to decide on the new course of action. Consensus problems, typified in swarm research as the best-of-*n* problem [14], in turn require some means of communicating the currently held opinion of any one robot to neighbours.

Direct communication between neighbours requires a degree of connectivity between the robots in the swarm. All-time connectivity uses approaches such as control laws to balance both the task at hand and the need for connectivity [15][16]. Such approaches necessarily restrict the movement of individual robots, and may be detrimental to performance [17]. Relay approaches may help with this, by delegating the job of providing connectivity to only some portion of the swarm [18][19].

Relaxing the need for all-time connectivity, path planning approaches [17] or ferries [20] may allow for an intermittent approach, but add complexity to swarm behaviour and require some or all robots to halt their task periodically.

The absence of explicit attempts to maintain communications links may be described as opportunistic, with robots transferring data to others in range when their paths happen to cross. This is the least restrictive approach and does not require dedicated roles or periodic rendezvous, but at the expense of guaranteed connectivity.

A crucial factor, regardless of the approach taken, is the communication range. The further apart any two robots may be when maintaining a communication link between them, the freer the robots are to move, and the fewer the number of robots that may be critical to network connectivity. As higher ranges may require more power and result in network interference [21], and lower ranges may decrease connectivity, finding a suitable broadcast range becomes desirable.

The mechanism for achieving this, described in the next section, is based on the existing concept of Pulse Monitoring (abbreviated to PBM due to its extension of Heart Beat Monitoring, HBM) [22], in which a periodic heartbeat message has a pulse encoded within it, allowing a component in a system to indicate its current health status. The concept has been explored in applications such as personal computers [23], telecommunications [24], and cluster management [25]. In order to support a reflexive reaction by minimising the processing required by a recipient, health-related data may be included in the message [24].

Pulse monitoring may be applied to a robot swarm, such as in [26], where it may be a means for a ruler craft during the Prospecting Asteroid Mission to monitor the health of workers under their control. However, another perspective may be used. In a dynamic swarm, where there is a need for scalability, it may be undesirable for one robot to track another's health over a significant period of time, and it cannot be expected that any one robot would rely upon another *specific* robot to assist in a task. Instead, pulses received during a small interval may represent the health of the local neighbourhood, allowing a robot to determine if its own status is abnormal, or provide early-warning of danger by noting problems developing in neighbouring robots.

Pulse monitoring is typically concerned with reporting on the health of whatever aspect is being monitored, as a form of failure management. In this paper, the concept is adapted to allow an individual robot to measure the local density of the swarm through the receipt of pulse messages from neighbouring robots that contain information about the source robots' positions. In this way, the "I am healthy" signal is replaced with one saying, "I am here". The design of the APC system is described in the next section.

III. AUTONOMIC PULSE COMMUNICATIONS

The goal of the APC system described in this paper is to provide a mechanism for the adaptive adjustment of the transmission range used for inter-robot communication, in order to avoid the pitfalls that come with needing to set the range used at the start of the mission.

To achieve this, the concept of PBM described in the previous section is adapted to repurpose the regular signal sent by each robot. In the Decentralised Autonomic Manager (DAM) described in [9], robots used periodic pulses to determine the local density of the swarm, but the pulse required a fixed transmission range used by each robot. If different transmission ranges were to be used, the density could not be easily calculated.

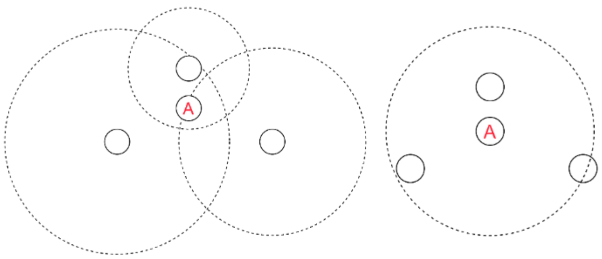
This problem is resolved by having each pulse also contain the position of the sending robot, allowing the distance from the pulse origin to the receiving robot to be calculated. Alternatively, situated communication [27] may be used to derive distance information from the received signal. Whichever approach is taken, the distance may be used to estimate the local density.

Fig. 1 (a) shows a case in which Robot A has a number of neighbours, all broadcasting pulse messages at different ranges, each of which is transmitted far enough to reach the robot. To simplify the example, all robots are shown to be sending their messages simultaneously, but the same process applies as long as all messages are received within the same short period of time. Each pulse contains the position of its sending robot.

By totalling the measured ranges of the received pulses, the APC system is able to calculate the average distance of pulse messages received. The local density, ρ, is then calculated as:

$$\rho = n / \pi \bar{d}^2, \tag{1}$$

where n is the number of received pulses in the time period, and $\bar{d}$ is their mean distance.



(a) (b)

Figure 1. A robot receives pulse messages from neighbours (a), and uses the encoded distance information to calculate a suitable range for its own pulse message.

Given a density, the APC system may then use a density-pulse range relationship provided in its knowledge base. This is tailored for the assigned task such that the ideal range for this task can be determined. In Fig. 1 (b), Robot A sends out its own pulse, with the range determined by that relationship, enabling its pulse message to reach its neighbours.

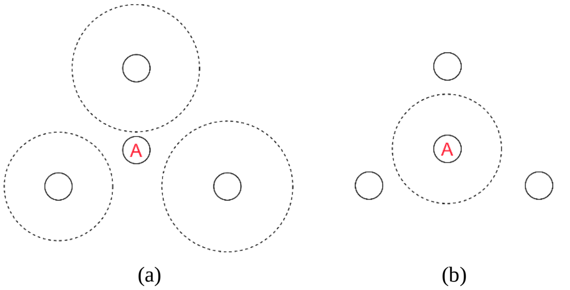
The APC system is only able to calculate a suitable local density if it receives pulse messages during the period between sending its own pulses. In Fig. 2 (a), the nearby robots are not sending pulse messages with sufficient range to reach Robot A. If none are received, the robot is considered to be isolated from the rest of its swarm. Its current pulse range may not be sufficient to reach its own neighbours, as in Fig. 2 (b), and so it gradually increases its broadcast range on subsequent pulses. This increases the chance that the robot will later reconnect with the other robots, in turn influencing future selections of the transmission range.

In addition to the distance information required by the APC system, pulse messages may also share arbitrary data, sent on each broadcast, for the purpose of spreading information throughout the swarm. In this work, the data packet is small and does not grow with size, so a simple strategy of sharing data with neighbouring robots is used, in which no individual robot needs to care about which robots receive a broadcast. This approach scales with the swarm size, as the underlying behaviour of the robots does not need to change for larger swarms.

IV. DATA SHARING FROM A SINGLE ROBOT

This research employs a time-stepped simulation of a homogeneous swarm of robots, tasked with sharing a piece of data throughout the swarm. In this simulation, elapsed time is measured in simulation ticks, while distances are in arbitrary units defining the simulation space, hereafter referred to simply as “units”. The robots are represented by a position only, with no physical size or robot-robot collisions. The purpose of this task is to determine how well a swarm of robots may share a single piece of information, initially held by only one robot in the swarm, with the rest of the members.

The swarm of robots, each using an APC system configured with a pulse period of 10 simulation ticks, and a fixed pulse range of 10 units, is placed in a circular map. Each robot stores a Boolean flag, initially set to false. At the start of each run, a robot is selected at random from the swarm and their flag is set to true. Any robot whose flag is true will share this data via the APC system. Receipt of the flag will cause a robot to set its own flag to true, and commence its own sharing.



(a) (b)

Figure 2. Nearby robot pulses are not strong enough to reach Robot A (a), which must gradually increase its own range to compensate (b).

During the test, the robots may wander freely throughout the map. Each tick of the simulation, a robot picks a random direction in two dimensions. If the robot is able to move forward one unit distance without leaving the map, the robot moves to that location, otherwise it will not move in this simulation update.

The test is left to run for 250 simulation ticks, and at the end, the success of the swarm in sharing the data is scored by the percentage of robots with their flag set to true. The test duration used will impact the density-range calculation, as the ideal range data used will be that which enables the swarm to reliably share the data with all members within 250 ticks.

All tests were run with the APC system set to stagger pulse times, rather than having all robots pulse simultaneously. This removes any requirement of the APC system to synchronise robot behaviour, while also avoiding flooding the available bandwidth with messages sent simultaneously.

V. TEST SCENARIOS

The following subsections describe the particular test scenarios run. Each test was run 50 times, and the results averaged across all runs.

A. *Density-Pulse Range Relationship*

To determine the relationship between the swarm density and the ideal pulse range to use, a set of simulations was run, for swarm sizes of 50, 100, 200, 500 and 1,000 robots, and maps with radii of 25, 50, 75 and 100 units.

The ideal pulse range for a given combination was determined by taking the lowest pulse range for which over 99.5% of the swarm, on average, received the data.

B. *Pulse Period*

This test explores how the APC pulse period affects the ability of the swarm to share the data. A map with a radius of 100 units was used, with the pulse range fixed at 10 units. The test was repeated with the five swarm sizes from the previous test, and pulse periods of 2, 5, 10, 15, 20 and 25 ticks. Each combination of swarm size and pulse period was tested, and the scores from each scenario are compared to evaluate the effects.

C. *Test Duration*

This test explores how the APC pulse period affects the ability of the swarm to share the data. A map with a radius of 100 units was used, with the pulse range fixed at 10 units. The test was repeated with the five swarm sizes from the previous test, and pulse periods of 2, 5, 10, 15, 20 and 25 ticks. Each combination of swarm size and pulse period was tested, and the scores from each scenario are compared to evaluate the effects.

D. *Adaptive Pulse Range*

The equation relating density and pulse range derived from the previous test is now used in the APC system to adaptively adjust the pulse range, based on the local swarm density. This test looks at the ability of this adaptive APC system to set an appropriate pulse range, and therefore share the data throughout the swarm.

The maps and robot counts are the same as those listed from the Density – Pulse Range tests. Each APC system starts with a pulse range of one unit, and uses a period of 10 ticks. The score for each combination of map and swarm size is measured, and compared against the best performing fixed range communication established in the previous test.

E. *Communications Loss*

To explore the impact of communications no longer being guaranteed to arrive, a swarm of 200 robots is tested on a map with a radius of 100 units. The simulation is configured with a probability of any robot receiving a broadcast range, and the test is run with probabilities of 20%, 15%, 10%, 5%, 4%, 3%, 2% and 1%, together with a test of the fixed range communications with a probability of communication success set to 5%. Every 10 ticks, the number of robots that have the flag set to true are recorded, and the results compared.

VI. RESULTS

The following subsections discuss the results of the tests described above.

A. *Density-Pulse Range Relationship*

Table I shows the best performing ranges and their respective scores for each combination of map radius and swarm size, while Fig. 3 shows the relationship between swarm density and best performing pulse range.

Fitting a trend line to the plot leads to an equation for determining the pulse range to use, given the density of the swarm:

$$r = 0.5884 \times \rho^{-0.652}, \tag{2}$$

where r is the pulse range, and ρ is the swarm density.

B. *Pulse Period*

Fig. 4 shows the performance for each size of swarm, as the pulse period is increased. Increasing the period results in a drop in the score achieved, which is less prominent in the largest swarms, and is most clearly seen with a swarm of 200 robots.

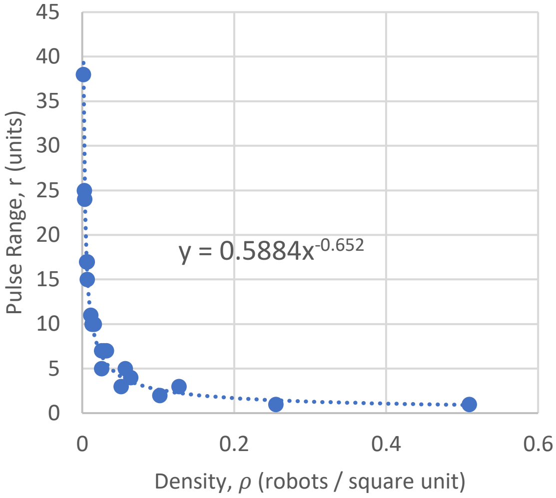


Figure 3. Plot of ideal pulse range against swarm density, for which ≥ 99.5% of the swarm received data shared starting with a single robot.

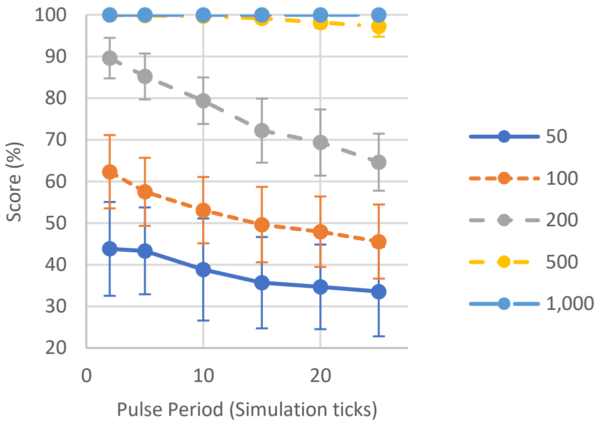


Figure 4. Score achieved by the swarm in sharing a data starting with a single robot, for the given pulse periods.

C. *Test Duration*

Fig. 5 shows the performance of each swarm size over time. It can be seen that denser swarms more quickly reach

TABLE I. PERFORMANCE OF SWARM IN SHARING DATA USING IDEAL PULSE RANGES FOR EACH COMBINATION OF MAP AND SWARM SIZE

Swarm Size	**Map Radius**							
	25		**50**		**75**		**100**	
	Range	Score	Range	Score	Range	Score	Range	Score
50	5	99.84%	15	99.72%	25	99.67%	38	99.88%
100	3	99.82%	10	99.62%	17	99.54%	24	99.60%
200	2	99.97%	7	99.94%	11	99.57%	17	99.53%
500	1	99.96%	4	99.94%	7	99.90%	10	99.74%
1,000	1	100%	3	100%	5	99.95%	7	99.87%

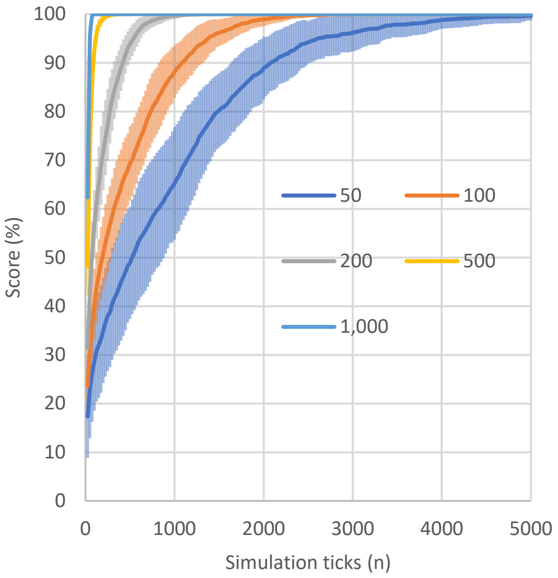


Figure 5. Score improvement during the data sharing scenario for swarms of different sizes in a 100-unit radius map. Shaded areas indicate one standard deviation.

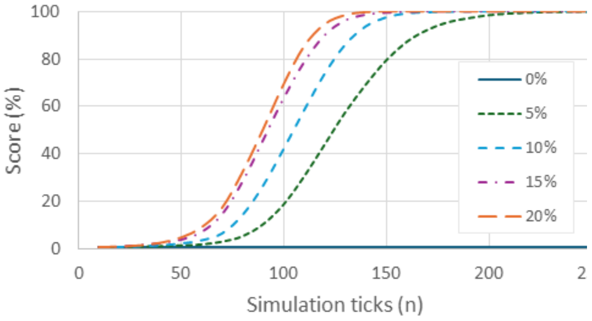
the point where all robots have received the information, but less dense swarms may require much more than the 250 ticks used as a standard in other tests.

D. *Adaptive Pulse Range*

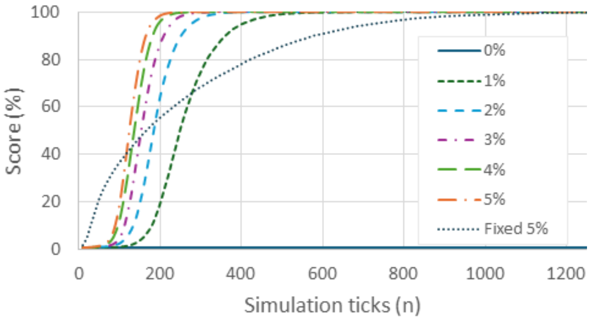
Table II shows the performance of the swarm, and average pulse range used, for each combination of map radius and swarm size. All scenarios achieved greater than the 99.5% score used as a benchmark in the fixed range tests, and all but three of the scenarios received a perfect score. The average pulse range used by the swarm can be compared against the ideal fixed ranges shown in Table I, and shows that higher density swarms make use of shorter-range pulses on average.

E. *Communications Loss*

Fig. 6 shows the performance of the swarm of 200 robots on a map with a 100-unit radius, in scenarios where the probability of a communications broadcast being received by a robot was 20% or lower. In addition, the chart shows the performance of the APC system running with a fixed pulse range, where communications have a 5% probability of succeeding.



(a) up to 20%



(b) up to 5%, with fixed pulse range

Figure 6. Performance of swarm in sharing data originating with a single robot, for given chances of a successful communication. At 0%, no messages are successfully received.

VII. FORAGING ROBOTS WITH APC

The APC system was inspired by the results of previous work on foraging robots [9], showcasing the need for an adaptive pulse range. That scenario is revisited here, employing the APC system to fulfil that requirement.

This research makes use of a time-stepped simulation of a heterogeneous swarm of agents, tasked with foraging for items within a square arena, as presented in previous work [9][28]. In this task, robots and items are placed within a grid at random, as shown in Fig. 7. Each robot and item may be either of two possible types, denoted by their colour. A single cell contains only one item, but may contain any number of

TABLE II. AVERAGE PULSE RANGES AND PERFORMANCE FOR SWARM SHARING DATA USING AUTONOMIC PULSE COMMUNICATION

Swarm Size	**Map Radius**							
	25		**50**		**75**		**100**	
	Range	Score	Range	Score	Range	Score	Range	Score
50	7.28 ± 0.32	100%	16.27 ± 0.33	100%	23.34 ± 0.47	99.96%	29.42 ± 0.51	99.64%
100	4.61 ± 0.14	100%	10.68 ± 0.21	100%	16.50 ± 0.18	100%	21.63 ± 0.26	99.98%
200	3.04 ± 0.07	100%	7.14 ± 0.13	100%	11.06 ± 0.17	100%	14.87 ± 0.19	100%
500	1.89 ± 0.02	100%	4.02 ± 0.08	100%	6.35 ± 0.07	100%	8.98 ± 0.11	100%
1,000	1.38 ± 0.00	100%	2.65 ± 0.03	100%	4.18 × 0.05	100%	5.82 ± 0.06	100%

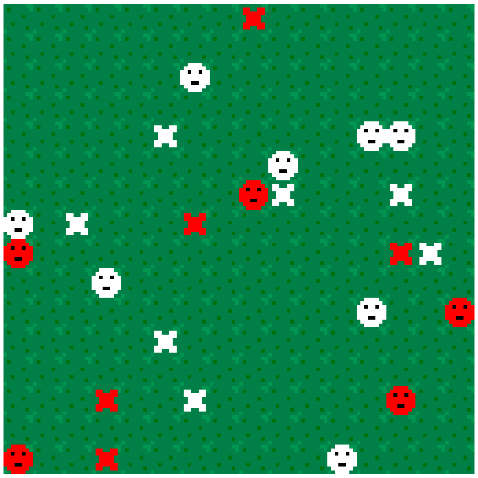


Figure 7. A portion of the world state during a simulation. The colour of a robot (face) or item (cross) indicates its type.

robots, with potential collisions between robots ignored by the simulation as each cell may be considered much larger than any one robot.

The simulation is updated in a time step manner, with each robot updated in turn for each tick of the simulation. The behaviour of the robots is based on the particular cooperation strategy they are using, as presented in [28]. In this work, the Help Recruitment and Blackboard strategies are used.

In the Help Recruitment strategy, as shown in Fig. 8, a robot begins in the Explore state. In this state, the robot moves to an adjacent cell in search of an item every tick of the simulation. If it finds an item, it moves to the Forage state, otherwise it will continue to Explore in the next tick.

In the Forage state, the robot determines the type of the item at that location. If the robot and item share a type, the robot is able to successfully forage the item, and so returns to the Explore state. However, if the robot and item are of different types, the robot must cooperate with neighbours. To do so, it broadcasts a recruitment message at a given range, containing the location and type of the item to be foraged, then moves to the Wait for Offers state.

Nearby robots in the Explore state that receive a help request will inspect the item's type, and if they are able to help, will send a response offering help, before moving to the Wait for Assignment state.

The original robot requesting help will wait for a short period and receive any offers. If found, the nearest responding robot to the item is selected and assigned the task, and the original robot can resume its previous behaviour. Robots which have offered help remain in the Wait for Assignment state for a short period before returning to Explore, however if they receive an assignment, they'll enter the Respond state in which they move directly towards the item to forage.

As can be seen in Fig. 8, a robot in the Respond state may find items en route that they are unable to forage, and they will send out help messages of their own before resuming their journey to their assigned item. When they reach the location, they will forage the item if found. If the item has been foraged by another robot in the intervening period, the responding robot will resume exploration.

The APC's ability to share data throughout the swarm presents an opportunity to employ the Blackboard strategy in which each robot maintains a list of known items while following the behaviour shown in Fig. 9. When in the Explore state, before a robot moves to a random adjacent cell, it first checks its knowledge base to see if there is a nearby item of the same type that it may move towards. If so, the robot will enter the Respond state in order to forage that item. During exploration or responding, if a robot finds an item it cannot forage, it will add it to its knowledge base.

To facilitate cooperation, knowledge is periodically broadcast to neighbouring robots, which synchronise the incoming data with their own knowledge. Each item is recorded with its position, type and forage status. By storing the status of an item, a robot is able to inform neighbours when an item has been foraged, thus spreading that information through the swarm, and preventing robots from moving to forage items which no longer exist. To balance

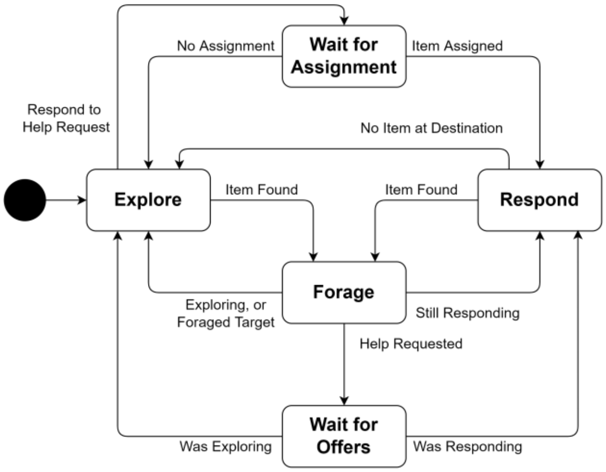


Figure 8. State Machine for the Help Recruitment cooperation strategy for use in the foraging task.

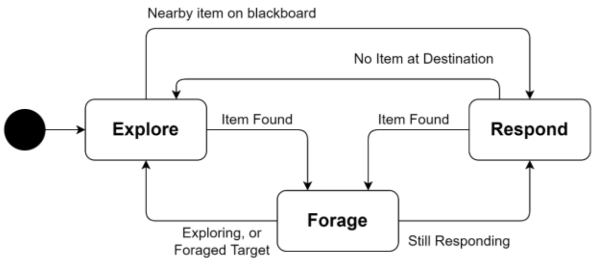


Figure 9. State Machine for the Blackboard strategy for use in the foraging task.

robot behaviour between exploration and responding to known items, a maximum response distance is used, under the assumption that the swarm is better served by ignoring distant items as other swarm members may be better placed.

As reported in [28], engaging in cooperative behaviour allows the swarm to find and remove all items more quickly. Further, for the Help Recruitment strategy to perform at its best, it was found in [9] that an autonomic system that manages the range of each help broadcast can improve performance. The Decentralised Autonomic Manager presented there works by using a fixed pulse range between robots for the purpose of estimating the local density, with the proviso that the appropriate pulse range must be set ahead of time.

In this paper, each robot makes use of the APC system to share their robot type and current position alongside the APC pulse messages. By receiving messages from neighbouring robots, each robot may deduce the numbers of each robot type in their neighbourhood, and subsequently calculate an appropriate density and select a suitable range. As in Section IV, the robots must make use of a relationship between the density of the swarm and the desired pulse range.

When using the DAM, every robot uses the same pulse range, and so it was possible to define the area from which pulses were received by treating it as a circle using the pulse range as the radius. With the APC, each robot may use a different pulse range, so the average distance of all received pulses is used instead.

A. *Methods*

To test the performance of the APC system, first it is necessary to determine the density-pulse range relationship for the task. To achieve this, swarms of between 16 and 320 robots, rising in 16-robot increments, were tested with fixed help broadcast ranges of 4 units, and 8-64 units, rising in 8-unit increments, and no APC system active.

Following that, swarm sizes of 32, 64, 128 and 256 robots, equally split between the two types, are deployed. Each configuration is run 50 times with a different initial position of robots and items, and the performance is measured as the number of simulation ticks taken to forage all items.

For the Help Recruitment strategy, these tests are carried out with the DAM set with fixed pulses of 8-64 units, rising in 8-unit increments, and again with the APC system, both making use of the density-pulse range relationship to determine a suitable range for help broadcasts. The performance of the APC is then compared against the best performing DAM configuration.

The Blackboard strategy is employed using the APC only, with performance compared against the performance of the Help Recruitment strategy in both DAM and APC configurations. As the Blackboard strategy requires a parameter dictating the maximum range at which a robot responds to a nearby item, this strategy is tested with maximum ranges of 8, 16, 24 and 32 units.

B. *Results*

Fig. 10 shows the relationship between swarm density and help broadcast range, with a trend line fitted. The resulting density-pulse range relationship for the foraging robots task is:

$$r_{help} = 1.4615 \times \rho^{-0.501}, \tag{3}$$

where r_{help} is the help broadcast range, and ρ is the swarm density calculated using (1).

Table III shows the performance of the two cooperation strategies implemented using the APC system, compared against the best-performing DAM configuration as determined by how quickly each configuration completes the foraging task.

For the Help Recruitment strategy, there is no statistical difference between the results of the DAM and the APC system at $p < 0.05$. The Blackboard strategy, on the other hand, shows a statistical difference with some values for the response range parameter, performing worse than the DAM in those cases. An exception is the case with 256 robots and a response range of 8 units, in which the APC system outperforms the DAM.

VIII. DISCUSSION

The results show that a relationship may be established between the performance of the swarm and the pulse range used for transmitting the data, as seen in Fig. 3. This relationship is specific to the task employed, in this case the sharing of data to at least 99.5% of the swarm within 250 ticks. Different tasks, with different requirements for success,

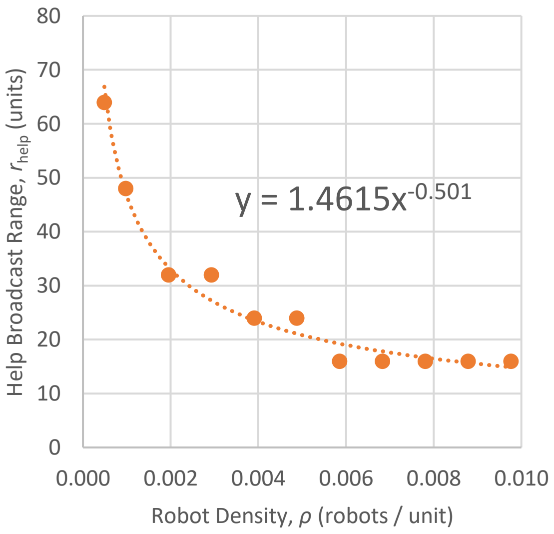


Figure 10. Plot of ideal broadcast range against swarm density, with a best fit trend line, for communications within a swarm of robots engaged in a foraging task.

TABLE III. Comparison of swarm strategy performance in a foraging task using the Decentralised Autonomic Manager and Autonomic Pulse Communication

Swarm Size	DAM Ticks			APC Ticks			t-statistic	p-value
	Range	Mean	SD	Strategy	Mean	SD		
32	24	9022.78	2692.76	Help	8863.28	2406.95	0.312	0.755
				Board 8	10164.52	2561.04	-2.172	0.032
				Board 16	9291.96	2710.00	-0.498	0.619
				Board 24	9452.50	2780.54	-0.785	0.434
				Board 32	9440.04	3786.79	-0.635	0.527
64	24	4331.44	1121.37	Help	4728.30	1604.44	-1.434	0.155
				Board 8	4799.04	1430.61	-1.819	0.072
				Board 16	4052.30	1278.41	1.161	0.249
				Board 24	4240.56	1130.31	0.404	0.687
				Board 32	4801.76	1429.02	-1.831	0.070
128	16	2073.22	667.62	Help	2127.40	577.34	-0.434	0.665
				Board 8	2354.30	869.60	-1.813	0.073
				Board 16	2076.66	481.86	-0.030	0.976
				Board 24	2333.58	810.23	-1.754	0.083
				Board 32	2414.86	640.94	-2.610	0.010
256	8	876.34	180.73	Help	1059.82	676.66	-1.852	0.067
				Board 8	696.38	239.71	-2.191	0.031
				Board 16	1130.18	371.97	-4.340	< 0.001
				Board 24	1182.20	315.82	-5.944	< 0.001
				Board 32	1477.02	445.47	-8.835	< 0.001

will necessarily result in a different relationship being established.

Increasing the pulse period has a detrimental effect on swarm performance, although it would reduce the energy used as fewer pulses would be sent. Balancing the performance needs of the swarm with the energy cost is an important factor, so a pulse period of 10 ticks was chosen for the adaptive APC and communications loss tests. Halving the period to 5 ticks would double the expected energy usage for only a small gain in performance, as seen in Fig. 4. Any performance decrease from using a longer period can be balanced through pulse range selection in the adaptive APC system.

Fig. 5, showing how the knowledge of the swarm improves with time, indicates that the larger the swarm, the faster the data is shared. As the test uses a fixed pulse range, the results are explained by noting that a larger swarm has a greater density, and so more robots are likely to be reached with each pulse. At the smallest size, some pulses may not be received by any robot. When the swarm contains 200 or fewer robots, it was not able to share data with all members within 250 ticks.

It is the responsibility of the adaptive APC system to address this problem, and the results in Table II show that the system, when starting with an initial pulse range of just one unit, is able to determine an appropriate range for a robot to broadcast at and enable the sharing of the data throughout the swarm within the allotted 250 ticks. This is a large improvement over the performance seen in Fig. 5, where the swarm of 50 robots still hasn't reached that knowledge level after 5,000 ticks.

When comparing the average pulse range in Table II to the best fixed ranges in Table I, the adaptive APC system is found to have a slightly higher range on average in lower density swarms, but in higher density swarms it can reduce the average pulse range, allowing the swarm to expend less energy. In the denser swarms, not every robot will detect the same local density, so the APC system enables the robots to reduce their pulse range while in higher density areas.

The APC system was also found to be extremely robust to communications loss, being able to successfully share the data within 250 ticks even when the probability of a successful message is as low as 5%, and it performs much better than the fixed pulse range at that level. A lower number of pulses being successfully received will result in a lower density estimate being made by the APC system, and a corresponding increase in the pulse range to reach more robots. While this system balances, increasing pulse ranges will increase energy usage.

It may be preferable for the swarm in cases of extremely high message loss to recognise the problem and find an alternative solution, perhaps contracting the swarm or temporarily increasing the period between pulses. Adaptive adjustment of the pulse period may help reduce energy usage overall, and this may be a topic for future work.

With the APC shown to be capable of allowing robots to adjust their pulse range in reaction to the perceived local density of the swarm, the next scenarios investigated the system's use in a foraging task. The results here show that the APC is capable of matching the performance of the DAM when used for the Help Recruitment strategy in the foraging task. By adaptively adjusting the pulse range based on the density of the swarm, the system does not require prior knowledge of the swarm size, making it useful in situations where the swarm may change due to robot loss, or the addition of reserves.

However, it is not perfect. While not statistically significant, the swarm of 256 robots appears to take longer with the APC system than with the DAM. This may be down to the difficulty in calculating the area around the robot from which the local density is derived. In the DAM, the fixed pulse range may be used as a radius. In the APC system, an average distance approach may be used, changing the density calculation.

The Blackboard, when implemented using the APC system, is also capable of matching performance in some cases, and in one case exceeding it. However, it requires an appropriately configured response range in order to do so, and an incorrect setting may negatively impact performance. This may itself be a candidate for adaptive adjustment based on the environment, using information such as the number of known items and the composition of the swarm.

Further, the Blackboard strategy has much higher data transfer requirements, increasing with every item known rather than the fixed size used for the Help Recruitment strategy. This may be mitigated by limiting the data sent in some way, perhaps using timestamps to favour recent data, or only sharing items nearby. Any advantage conferred by the Blackboard strategy should be balanced against the strategy's requirements.

IX. CONCLUSION AND FUTURE WORK

This research presented a system for adaptively adjusting the range of communications between robots based on the density of the swarm, by adapting the existing concept of Pulse Monitoring. By replacing the "I am healthy" message with one saying, "I am here", a receiving robot can use the aggregate data presented by multiple received pulses to estimate the local density of the swarm.

In a task to share a piece of data with the rest of the swarm, the Autonomic Pulse Communications system was able to adaptively determine the pulse range to use to achieve excellent results, ensuring that 100% of the swarm received the data within the allotted time in all but three scenarios. The results show the system selecting shorter pulse ranges when the swarms are denser, and compare favourably with the best performing fixed pulse ranges used to establish the relationship between density and pulse range that the system uses. Further, the APC system was shown to be extremely robust to communications loss, as the system adapts to a decrease in the number of received messages by increasing the pulse range, thus increasing the chances of the message being received by some robots.

The APC system was then used to implement both the Help Recruitment and Blackboard strategies for a swarm of foraging robots. The performance was shown to be comparable to that of the best performing DAM which required a pulse range to be set prior to the mission. The APC lifts that restriction, successfully enabling the swarm to adapt the pulse range according to the measured swarm density.

The APC system therefore shows promise, allowing a swarm to maintain communication links between its members while imposing fewer restrictions on the behaviour of the robots. Should the swarm suffer loss of robots over the course of the mission, the resulting lower density of the swarm may be compensated for automatically by the system.

This work was carried out exclusively using simulation, which may suffer from what is termed the "reality gap" [29], where results obtained in simulation are not replicated when the same experiment is run in reality. The abstract nature of the simulations used here means there are several steps that can be taken to close the gap, however the ideal test environment would use real physical hardware.

Individual pulse messages used in this work were simplified, by considering them to be atomic actions. Larger amounts of data may take longer to broadcast than small packets, and this will impact the ability of a robot to successfully receive all of the data in a single broadcast. The motion of the robots may result in a recipient moving out of range before the transmission is completed. Additionally, communications failure was simply modelled as a random chance of failure, not taking into account the operating conditions or physical obstructions in the path.

Future work may investigate the impact of those aspects on the system, as well as applying the APC system to other tasks such as collective decision-making. Another avenue of interest may be the mechanism by which data is shared. As information grows in complexity, it may be desirable to selectively share only a portion of data in order to minimise the time and energy costs of data transfer, keeping the pulse messages short.

Further work may also investigate the impact of other factors in the ability of the swarm to share data. In this work, the data to be shared was fixed, so a changing data set that requires frequent reporting should be investigated. Also of note is the movement of the swarm, which supports data sharing through changing the set of neighbours receiving a robot's pulse. Different robot speeds, more limited mixing, and the absence of motion altogether may impact the performance of the system.

REFERENCES

[1] L. McGuigan, R. Sterritt, and G. Hawe, 'Autonomic Pulse Communications for Adaptive Transmission Range in Decentralised Robot Swarms', Sep. 2023, pp. 15–21, Accessed: Jan. 22, 2024. [Online]. Available: https://www.thinkmind.org/index.php?view=article&articleid=emerging_2023_1_30_50021.

[2] E. Sahin, 'Swarm robotics: From sources of inspiration to domains of application', in *Swarm Robotics*, vol. 3342, E. Sahin and W. M. Spears, Eds. 2005, pp. 10–20.

[3] M. G. Hinchey, R. Sterritt, and C. Rouff, 'Swarms and Swarm Intelligence', *Computer*, vol. 40, no. 4, pp. 111–113, Apr. 2007.

[4] V. Trianni, J. IJsselmuiden, and R. Haken, 'The SAGA concept: Swarm Robotics for Agricultural Applications', Technical Report, 2016. Accessed: Dec. 21, 2022. [Online].

Available: http://laral.istc.cnr.it/saga/wp-content/uploads/2016/09/saga-dars2016.pdf.

[5] L. Abraham, S. Biju, F. Biju, J. Jose, R. Kalantri, and S. Rajguru, ‘Swarm Robotics in Disaster Management’, in *2019 International Conference on Innovative Sustainable Computational Technologies (CISCT)*, Oct. 2019, pp. 1–5.

[6] G. Beni, ‘From Swarm Intelligence to Swarm Robotics’, in *Swarm Robotics*, Berlin, Heidelberg, 2005, pp. 1–9.

[7] J. O. Kephart and D. M. Chess, ‘The vision of autonomic computing’, *Computer*, vol. 36, no. 1, pp. 41–50, Jan. 2003.

[8] E. Vassev, R. Sterritt, C. Rouff, and M. Hinchey, ‘Swarm Technology at NASA: Building Resilient Systems’, *IT Prof.*, vol. 14, no. 2, pp. 36–42, Mar. 2012.

[9] L. McGuigan, R. Sterritt, G. Wilkie, and G. Hawe, ‘Decentralised Autonomic Self-Adaptation in a Foraging Robot Swarm’, *Int. J. Adv. Intell. Syst.*, vol. 15, no. 1 & 2, pp. 12–23, 2022.

[10] M. Puviani, G. Cabri, and L. Leonardi, ‘Enabling Self-Expression: The Use of Roles to Dynamically Change Adaptation Patterns’, in *2014 IEEE Eighth International Conference on Self-Adaptive and Self-Organizing Systems Workshops*, Imperial College, London, United Kingdom, Sep. 2014, pp. 14–19.

[11] F. Zambonelli, N. Bicocchi, G. Cabri, L. Leonardi, and M. Puviani, ‘On Self-Adaptation, Self-Expression, and Self-Awareness in Autonomic Service Component Ensembles’, in *2011 Fifth IEEE Conference on Self-Adaptive and Self-Organizing Systems Workshops*, Ann Arbor, MI, USA, Oct. 2011, pp. 108–113.

[12] J. Zelenka, T. Kasanický, and I. Budinská, ‘A Self-adapting Method for 3D Environment Exploration Inspired by Swarm Behaviour’, in *Advances in Service and Industrial Robotics*, Cham, 2018, pp. 493–502.

[13] C. Saunders, R. Sterritt, and G. Wilkie, ‘Autonomic Cooperation Strategies for Robot Swarms’, in *Adaptive 2016: The Eighth International Conference on Adaptive and Self-Adaptive Systems and Applications*, Rome, Italy, Mar. 2016, pp. 20–27.

[14] G. Valentini, E. Ferrante, and M. Dorigo, ‘The Best-of-n Problem in Robot Swarms: Formalization, State of the Art, and Novel Perspectives’, *Front. Robot. AI*, vol. 4, 2017, Accessed: Jan. 03, 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frobt.2017.00009.

[15] B. Capelli and L. Sabattini, ‘Connectivity Maintenance: Global and Optimized approach through Control Barrier Functions’, in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 5590–5596.

[16] B. Capelli, H. Fouad, G. Beltrame, and L. Sabattini, ‘Decentralized Connectivity Maintenance with Time Delays using Control Barrier Functions’, in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 1586–1592.

[17] Y. Kantaros, M. Guo, and M. M. Zavlanos, ‘Temporal Logic Task Planning and Intermittent Connectivity Control of Mobile Robot Networks’, *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 4105–4120, Oct. 2019.

[18] N. Majcherczyk, A. Jayabalan, G. Beltrame, and C. Pinciroli, ‘Decentralized Connectivity-Preserving Deployment of Large-Scale Robot Swarms’, in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 4295–4302.

[19] V. S. Varadharajan, D. St-Onge, B. Adams, and G. Beltrame, ‘Swarm Relays: Distributed Self-Healing Ground-and-Air Connectivity Chains’, *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5347–5354, Oct. 2020.

[20] P. Smith, R. Hunjet, A. Aleti, and J. Barca, ‘Data Transfer via UAV Swarm Behaviours’, *J. Telecommun. Digit. Econ.*, vol. 6, pp. 35–57, Jun. 2018.

[21] P. Gupta and P. R. Kumar, ‘The capacity of wireless networks’, *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[22] R. Sterritt, ‘Pulse monitoring: extending the health-check for the autonomic grid’, in *IEEE International Conference on Industrial Informatics, 2003. INDIN 2003. Proceedings*, Banff, AB, Canada, 2003, pp. 433–440.

[23] R. Sterritt and S. Chung, ‘Personal autonomic computing self-healing tool’, in *Proceedings. 11th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems, 2004*, May 2004, pp. 513–520.

[24] R. Sterritt, D. Gunning, A. Meban, and P. Henning, ‘Exploring autonomic options in a unified fault management architecture through reflex reactions via pulse monitoring’, in *Proceedings. 11th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems, 2004*, Brno, Czech Republic, 2004, pp. 449–455.

[25] W. Truszkowski, M. Hinchey, and R. Sterritt, ‘Towards an Autonomic Cluster Management System (ACMS) with Reflex Autonomicity: Workshop on Reliability and Autonomic Management in Parallel and Distributed Systems (RAMPDS-05) at ICPADS-2005’, *Unkn. Host Publ.*, pp. 478–482, Jul. 2005.

[26] E. Vassev and M. Hinchey, ‘Self-Awareness in Autonomous Nano-Technology Swarm Missions’, in *2011 Fifth IEEE Conference on Self-Adaptive and Self-Organizing Systems Workshops*, Ann Arbor, MI, USA, Oct. 2011, pp. 133–136.

[27] K. Støy, ‘Using Situated Communication in Distributed Autonomous Mobile Robotics’, *SCAI*, vol. 1, pp. 44–52, Feb. 2001.

[28] L. McGuigan, C. Saunders, R. Sterritt, and G. Wilkie, ‘Cooperation Strategies for Swarms of Collaborating Robots: Analysis of Time-Stepped and Multi-Threaded Simulations’, *Int. J. Adv. Syst. Meas.*, vol. 14, no. 3 & 4, pp. 44–58, 2021.

[29] N. Jakobi, P. Husbands, and I. Harvey, ‘Noise and the reality gap: The use of simulation in evolutionary robotics’, in *Advances in Artificial Life*, vol. 929, F. Morán, A. Moreno, J. J. Merelo, and P. Chacón, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 704–720.