An Analysis of Emergent Taxis in a Wireless Connected Swarm of Mobile Robots

Jan Dyre Bjerknes, Alan FT Winfield, Chris Melhuish
Bristol Robotics Laboratories
Coldharbour Lane
Bristol UK BS16 1QY
Email: {jan.dyre.bjerknes, chris.melhuish}@brl.ac.uk, alan.winfield@uwe.ac.uk

Abstract— **In swarm robotic systems emergent swarm properties are particularly difficult to analyse and model. This paper describes a simple but effective algorithm for emergent swarm taxis (swarm motion toward a beacon) in a 2D or 3D wireless connected swarm of minimalist mobile robots. The paper then undertakes a deep analysis of the swarm taxis by identifying both first and second order micro-level robot interactions and quantifying the contribution of each such interaction to the macro-level swarm behaviour. From the analysis we develop a simple quantitative model that is able to predict swarm velocity with reasonable accuracy. Although the analysis is specific to the swarm algorithm in question, we believe that the methodology presented has generic value to swarm modellers.**

I. Background.

The fields of robotics and artificial intelligence have seen increased interest in swarm systems in recent years. Drawing inspiration from biology and in particular social insects, researchers in swarm robotics aim to build systems that comprise large numbers of autonomous robots that self-organise to achieve useful work, without centralised or heirarchical control.

The idea of de-centralised control is essential. Almost all traditional technologies and social structures (governments and companies, for instance) employ centralised control. One agent is the leader that organizes the other agents and instructs them what to do and when to do it. In swarm systems no such leader exists. The organization is completely 'flat'; there is no hierarchy, and all the agents in the system have the same amount of influence. Furthermore, robots in a swarm robotic system typically cannot communicate with all other robots. Rather, each robot can only communicate with a few of its nearest neighbours. In other words, communication is local.

Typically a robot might only communicate with five to ten of its closest neighbours, even though a swarm can comprise hundreds of robots. The challenge for swarm roboticists is to design the communication and behavioural rules for the individual agents so that the overall desired swarm properties and functions will emerge from the interactions between individual robots: Each other and their environment.

One of the significant advantages of such a decentralised approach is robustness. Since there is no common-mode failure point or vulnerability in the swarm, overall swarm behaviours should operate even if one or a number of individual robots fail [1]. Due to the locality of sensing and communication swarm robotic systems also have the potential to scale to large numbers [2].

Work in swarm robotics to date has fallen into two relatively distinct categories. In the first of these categories, the swarm performs the task faster than an individual robot could do on its own, but adds no new qualitative ability. This will typically be for swarm tasks such as foraging, area mapping and clustering [3], [4], [5]. A single robot may solve the problem alone given enough time, but applying a swarm greatly enhances the performance time and robustness. The swarm level exhibit a new *quantitative* property compared to the individual robots.

In the second category swarm task completion is dependent on the collective. Such tasks are dependent on cooperation between multiple robots; one single robot could not complete the task, even given infinite time. In such tasks, the swarm level behaviour is *qualitatively* different compared to the individual robots. One example is collective manipulation (stick pulling) [6]. A further example is our work on wireless connected swarms that exhibit interesting and potentially useful emergent swarm properties, including swarm aggregation, ad-hoc networking, beacon-taxis and beacon containment [7]. All of these properties emerge from the interaction of multiple robots. This paper focuses on emergent swarm taxis, and develops the algorithm developed by Nembrini et al [7], [8]. We then undertake a detailed analysis of the low-level interactions which give rise to the swarm taxis behaviour (hitherto, the mechanisms of this swarm taxis behaviour have not been fully understood), and armed with this analysis we propose a simple model with which we can predict swarm velocity from low-level robot parameters.

Modelling and analysis of swarm robotic systems poses a significant challenge to researchers in the field. Developing models with predictive power, such that macro-swarm properties can be predicted from changes in micro-robot parameters is difficult, not least because robotic swarms are examples of non-linear, dynamical, stochastic systems.

There are two main motivations for making models. Models are important for proving safety properties of swarm systems. There are some great promises in swarm robotics, e.g for huge swarms of nano sized robots to solve certain medical problems [9] [10]. If swarm robotics is to deliver on these promises, formal proofs of swarm properties and swarm models are essential. We also believe that good models will facilitate the

1-4244-0708-7/07/$20.00 ©2007 IEEE

design of swarm systems. If we want to design a swarm system with a given property, models that relate individual robot properties to swarm level properties avoid the need to find solutions by trial and error. For a recent survey of modelling approaches see [11].

Although the analysis developed in this paper is specific to the emergent swarm taxis algorithm, we believe that the methodology presented has generic value in extending the tools and methods available to swarm modelers.

This paper proceeds as follows. In section II we describe the enhanced swarm taxis algorithm, with reference to the finite state machine for individual robots. Section III develops an informal analysis of the taxis behaviour by identifying and analysing micro-level pair-wise interactions between robots. Sections IV, V an VI quantify the contributions to swarm taxis from each of these micro-interactions, and hence build a formal model of swarm taxis, with results comparing predicted and measured swarm velocities.

II. Swarm Taxis Algorithm

We have developed a class of algorithms which make use of local wireless connectivity information alone to achieve swarm aggregation [7], [8]. Wireless connectivity is linked to robot motion so that robots within the swarm are wirelessly ‘glued’ together. This approach has several advantages: firstly the robots need neither absolute or relative positional information; secondly the swarm is able to maintain aggregation (i.e. stay together) even in unbounded space, and thirdly, the connectivity needed for and generated by the algorithm means that the swarm naturally forms an *ad hoc* communications network. Such a network would be a requirement in many swarm robotics applications. The algorithm requires that connectivity information is transmitted only a single hop. The algorithm meets the criteria for swarm robotics, articulated by Şahin, 2005 [12] and Beni, 2005 [13]. We have a highly *robust* and *scalable* swarm of homogeneous and relatively incapable robots with only local sensing and communication capabilities, in which the required swarm behaviours are truly emergent.

The lowest level swarm behaviour is ‘coherence’ which, in summary, works as follows. Each robot has range-limited wireless communication and, while moving, periodically broadcasts an ‘I am here’ message. The message will of course be received only by those robots that are within wireless range, r_w. It is important to recognise that this communication is fully situated [14], i.e. there is no content in the messages, all that is significant is message presence or absence. Robots do not communicate any information on their internal state etc, nor is it possible for a robot to determine its heading relative to the communicating robot. If a robot loses a connection *and* the number of remaining neighbours is less than or equal to the threshold α, then it assumes it is moving out of the swarm and will execute a $180°$ turn. When the number of connections rises (i.e. when the swarm is regained) the robot chooses a new direction at random. We say that the swarm is coherent if any break in its overall connectivity lasts less than a given time constant C. Coherence gives rise to the two basic emergent behaviours of swarm aggregation and a (coherent) connected *ad hoc* wireless network. Each robot also has short-range avoidance sensors and a long-range beacon sensor.

Firstly, the short range collision avoidance sensor. The robots will use this sensor to avoid colliding into each other (or other obstacles in the environment). This sensor provides robots with information about the relative direction towards the obstacle.

Secondly, the robots have an omnidirectional beacon sensor. This sensor can detect if the robot is illuminated by the beacon source. We have assumed that there is no ambient beacon radiation in the environment. Importantly the beacon source is placed on the same level as the robots, so that one robot can occlude the beacon from another.

In our simulation environment[1] the robots have a physical radius of 0.3 units. When they move they cover a distance of 0.05 units per time step, and we refer to this distance as D_{robot}. In experiments, the wireless communication range, r_w, varies from 2.5 units to 4.0 units. This allows us to see how swarm taxis velocity varies with different communication ranges, and how well our model captures this variation. The velocity of the whole swarm is denoted D_{swarm} and refers to the distance the swarm centroid moves per time step. An important detail of the enhanced swarm taxis algorithm is that a robot’s avoid distance is varied dependent on whether the robot’s beacon sensor is illuminated or non-illuminated. If a robot is in the shadow of another robot (or an obstacle) it will have a smaller avoid radius as opposed to when it is illuminated. The avoid radius for shadowed robots is arbitrarily chosen as 0.4 units, and 0.51 for illuminated robots.

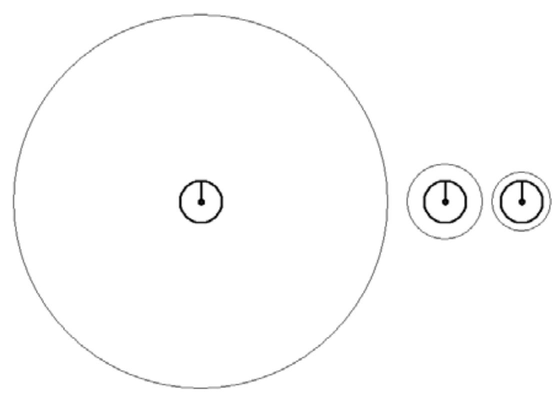

Fig. 1. Three robots drawn to scale with, r_w 2.5 units (left), avoid radius for illuminated robots, 0.51 units (middle), and avoid radius for shadowed robots, 0.4 units (right). The robot radius is 0.3 units.

A. States and behaviours

The robots have five different states and for each state a corresponding behaviour. See figure 2. There are four transition rules that determine transitions between the states. The default state is the Forward state. When the robot is in this state it will move forward by D_{robot} units per time step. The coherence

[1] Netlogo. http://ccl.northwestern.edu/netlogo/

state maintains swarm aggregation, as described above. Given the different avoid sensor radii, depending upon the robot's illuminated status, there are two avoidance states. Finally we have a state for random turns. As we will see later, it is the two avoidance states that enable the swarm to perform taxis.

The default state is the forward state. Dependent on the robots' environment other states can be invoked, but as soon as the corresponding behaviour is performed the robot returns to the forward state. While in the forward state the robots continuously monitor the number of robots within communication range (neighbours) and the avoidance and light sensors. If the number of neighbours falls below a predefined value, α, the robot will enter the coherence state. In this state the robot will perform a 180° turn. Assuming that a robot lost a connection because of moving away from the swarm, a 180° turn will ensure that the robot will reconnect with the swarm again, contributing to maintaining swarm aggregation. As soon as the 180° turn has been performed, the robot re-enters the default forward state.

The random state is entered when the robot notices an increase in number of robots within communication range. Since this number is increasing the robot may be moving closer to the center of the swarm. In this state the robot will make a random turn, to a new direction and then return to the forward state.

There are two avoid states, one which applies when the robot is illuminated, the other when the robot is in shadow. The short range avoidance sensor also provides accurate heading towards the colliding object. When an object is being detected the robot will turn in the opposite direction to the object, and then return to the forward state. The difference between the two avoid states is the range. When a robot is illuminated the avoid radius is 0.51 units, 0.4 when it is in a shadow.

Since our simulation is a discrete system there is a chance of several conditions being triggered at the same time. In such cases, the coherence and random states take precedence over avoid states. Coherence and random states can not have their conditions satisfied at the same time. When the robot enters

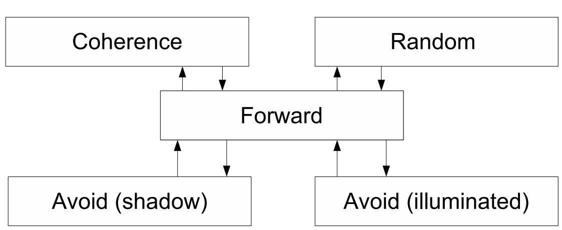

Fig. 2. The state diagram of the robot control.

the avoid state, it will make a turn in the opposite direction, and then immediately return to the forward state. The same holds true for the coherence and random turn states. As a simplifying assumption we have set the turn time to zero.

B. *Symmetry breaking necessary for taxis*

Thus far we have described how the swarm maintains aggregation. One behaviour keeps the swarm together (coherence), and another prevents the robots from colliding into each other (avoidance). But to achieve taxis we need some kind of *symmetry breaking*. Information of the direction towards the beacon must somehow be captured by the swarm. We do this via the avoid states. Consider the case of robots A and B in

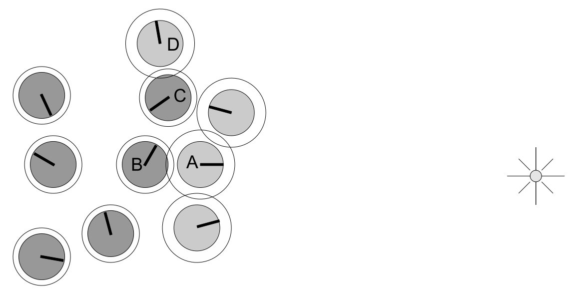

Fig. 3. A swarm of ten robots and a beacon source. Due to the difference in avoid radius the swarm will move towards the beacon. The light gray robots are illuminated, the dark gray robots are non-illuminated, i.e. occluded.

figure 3. We see that robot A, which is illuminated, will try to avoid robot B, since B is within A's avoidance radius. But there will be no similar behaviour on behalf of B since B's avoid radius is smaller, and hence B does not detect A. It is exactly this difference in avoid radius that gives rise to the beacon taxis behaviour.

The position of the swarm is defined as the average X and Y positions of the robots, i.e as the centroid. Swarm movement is then the change in the centroid position. This means that if two robots move in exactly opposite directions with the same distance, they will not change the position of the swarm. The movement of one robot contributes to movement of the swarm with a value that is equivalent to the distance moved by the robot divided by the swarm population N.

In simulation we have varied the wireless communication range, and run 50 simulations for each range. In all cases the swarm successfully reached the beacon. The swarm velocity and communication range are plotted in figure 4. We see that the swarm velocity increases with the communication range. This will be analysed and explained in detail later in this paper.

III. Informal analysis of the taxis behaviour

In order to develop a model of swarm taxis we will in this section informally describe how the different interactions between robots contribute to the overall swarm behaviour.

A. *Coherence behaviour*

We first make the simplifying assumption of α = population size N. Even though this means that the swarm will be fully connected - or attempt to be fully connected - we have chosen to fix α at this value because the swarm becomes easier to analyze. The rationale is that when $\alpha = N$ the coherence behaviour will always take place in pairs. In other words when a robot loses a connection there will always be another robot, i.e. the one with which the connection was lost, that also loses a connection, see figure 5. Since both robots will then always have less than α connections, both will enter the coherence

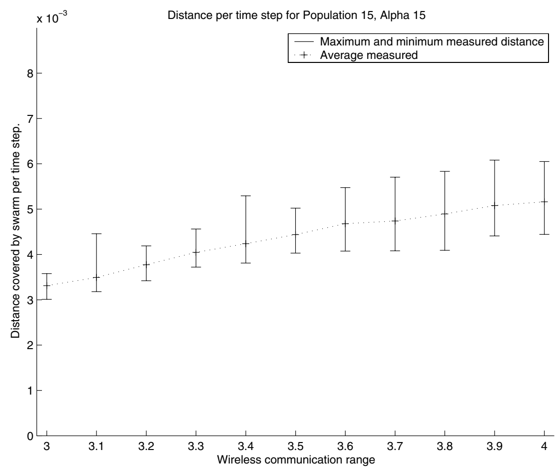

Fig. 4. The D_{swarm} for a swarm with 15 robots for different wireless ranges. Each point is the average of 50 runs. The error bars denote the fastest and slowest in each group.

state. From the perspective of overall swarm movement the two actions will cancel each other out and hence provide no net contribution to swarm taxis.

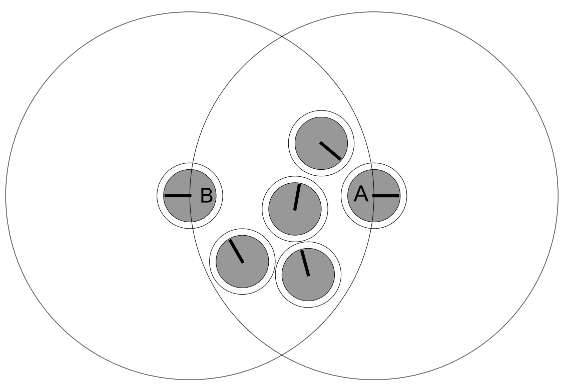

Fig. 5. Coherence movement takes place in pairs. Robot A loses contact with robot B and vice versa at the same time. They will both turn 180° in the next time step.

B. Collision between non-illuminated robots

When two robots that are both in shadow get too close to each other, they will enter the avoid state at the same time. Their avoid sensors have the same range, and they will perceive each other's direction accurately and at the same time. Hence they will move in opposite directions and therefore the avoid behaviour provides no net movement for the swarm. Non-illuminated collisions are thus neutral with respect to taxis.

C. Collision between illuminated robots

An identical argument can be made for the avoidance behaviour between two robots that are both illuminated. Even though their avoidance radii are larger than that of non-illuminated robots, each robot has same radius, ensuring they will avoid each other at the same time. The net movement from the swarm perspective is therefore nil.

D. Collision between robots with different illumination status

The case of particular interest is that of collision between two robots with different illumination status. Because of its larger avoid sensor range the illuminated robot will detect the non-illuminated robot at a greater distance and the illuminated robot will consequently enter the avoid state without being detected by the non-illuminated robot; this is where the symmetry breaking necessary for swarm taxis takes place. The illuminated robot will move away from the non-illuminated robot, and since by definition the illuminated robot is closer to the beacon then we have a contribution to the net swarm movement (taxis) towards the beacon.[2] When an illuminated robot is avoiding a non-illuminated robot as described here, it will move towards the beacon; for simplicity we will refer to a robot in this particular condition as being in a 'progressive' state. The progressive state is really no more than

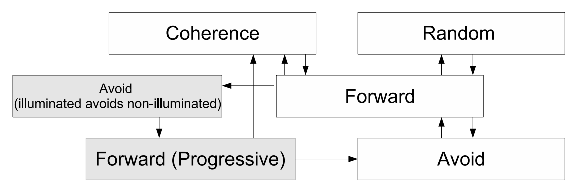

Fig. 6. When a robot moves towards the light, we refer to this as being in a progressive state. This is a quasi-state that exits independently of the control states. See full text for detailed explanation

a special case of the forward state, however there is value in differentiating this quasi-state for the purposes of this analysis. Figure 6 shows the new extended state diagram for the robot, with two quasi-states added for avoid and forward.

E. An exception case

We saw above that due to the difference in collision avoidance radius illuminated robots will differentially avoid non-illuminated robots. If the illuminated robot is the one that generates the shadow for the non-illuminated robot, then we see the net contribution to swarm taxis described above. This is the situation we see in figure 3 between robots A and B. But there can be exceptional cases where an illuminated robot avoids a shadowed robot, but is not itself the robot that casts the shadow. Once again we can see such a situation in figure 3: robot D, illuminated, will avoid robot C, shadowed. However, robot C is not shadowed by robot D, but some other robot that is closer to the beacon. Thus this collision avoidance will not provide any net contribution to swarm taxis. When we calculate the contribution a robot has on the progress of the swarm we must take such exceptional cases into account. This will be spelled out in detail in the formal analysis.

[2]Note that we could achieve a negative beacon-taxis swarm behaviour by simply reversing the roles of illuminated and non-illuminated robots.

IV. THE FORMAL ANALYSIS, STEP ONE

We saw in the previous section that there will be cases where a collision between an illuminated robot and a non-illuminated robot does not necessarily contribute to swarm taxis. In our simulated environment up is denoted by 0°, right is 90°, down is 180°and left is 270°. The robot swarm starts on the left side of the environment and the beacon source is placed to the right. If we again consider the case of robots A and B in figure 3, we see that robot A's new heading after avoiding robot B will be approximately 90°. The distance moved towards the light, $D_{progress}$, in one time step is

$$D_{progress} = D_{robot} * \sin heading \quad (1)$$

In this case sin of heading is 1 and the movement towards the beacon is the same as the movement per time step, 0.05 units. We can compare this with the case of robots C and D in figure 3. We see that robot D has a heading of 350°. The distance moved towards the beacon for one time step then becomes

$$D_{progress} = 0.05 * \sin 350 = 0.05 * -0.17 = -0.0087 \quad (2)$$

So this interaction will contribute with a small negative movement towards the beacon.

We have measured the new heading of an avoiding robot for each interaction and found the average sine value for the avoiding robots. This value must be taken into account when we then calculate the contribution of one interaction. The average sine value varies a little across experiments as the wireless distance changes. Let θ denote the sine factor. In the first set of simulations θ varies from 0.473 to 0.453. By taking this value into account we can calculate how much a single robot moves towards the beacon per interaction.

We have seen above how *one* interaction between an illuminated and non-illuminated robot yields a small movement towards the beacon. But to be able to determine the speed of the whole swarm we also need to know how often such collisions take place, and equally importantly, after a collision for how long do robots stay in the 'progressive' quasi-state before being disturbed and switching to avoid or coherence states. Let I_{prog} denote the number of interactions between an illuminated robot and a non-illuminated robot. Further, let t_{prog} denote the time a robot stays in the progressive state before being disturbed. Now it might be possible to calculate these values from first principles, but here for simplicity we measure these values from simulation. We argue that this is justified on the grounds that what we measure are micro-level robot properties, in order to model and predict macro-level swarm properties.

Now that we have established the necessary variables, we make our first attempt at predicting the swarm velocity from the interactions.

$$PD_{swarm} = \frac{I_{prog} * t_{prog} * D_{robot} * \theta}{N} \quad (3)$$

where PD_{swarm} is the predicted distance covered by the whole swarm towards the beacon in one single time step.

V. RESULTS FROM THE FIRST STEP IN THE FORMAL ANALYSIS

We have run experiments with different population sizes and different values for the range of the wireless communication. In particular we have used swarm sizes of 10 and 15, communication ranges from 2.5 to 3.5 for swarms with population of 10, and communication range 3.0 to 4.0 for swarms with population 15. We have taken the average of 50 runs for each communication range and swarm size, a total of 1100 runs. We will give values for I_{prog}, t_{prog}, and θ, and calculations for one run. Remaining runs will be summarised.

We start with the first set of experiments. The population in this case is 15. In this case we have measured swarm taxis speed for wireless communication ranges from 3.0 to 4.0 units, in 0.1 step increments. A value less than 3.0 packs the swarm too tightly together so a minimum value of 3.0 was chosen. We see from the table I that there is about one progressive interaction per time step. However, as the wireless communication range increases the time in which the robot stays in the progressive behavior gets longer. The smallest is 1.57 time step up to 2.73 time steps. From these values we can do the first calculation of the swarm velocity. If we use the values when the wireless communication distance is 3.0 we get a swarm velocity of

TABLE I

NUMBER OF PROGRESSIVE INTERACTIONS PR TIME STEP AND CORRESPONDING VELOCITY, AVERAGE OF 50 RUNS

r_w	I_{prog}	t_{prog}	θ	PD_{swarm}
3.0	0.93	1.57	0.45	0.00224
3.1	0.96	1.65	0.45	0.00239
3.2	0.98	1.76	0.44	0.00256
3.3	0.99	1.88	0.43	0.00271
3.4	1.00	2.00	0.43	0.00288
3.5	0.99	2.11	0.42	0.00301
3.6	0.98	2.25	0.42	0.00314
3.7	0.97	2.37	0.41	0.00320
3.8	0.96	2.48	0.41	0.00333
3.9	0.94	2.59	0.41	0.00340
4.0	0.91	2.73	0.41	0.00348

$$PD_{swarm} = \frac{0.93 * 1.57 * 0.05 * 0.45}{15} = 0.0022 \quad (4)$$

In figure 7 we see a graph of the swarm velocity against wireless communication distance. We see that the swarm moves faster as the wireless range increases. This is partly explained by the fact that the higher the wireless communication range, the longer the robots can stay in their progressive state, without being disturbed. Thus each single interaction contributes more to the overall movement.

However, we can see that there is a large offset between the real values and the predicted values. This seems to suggest that there is something else in the interactions between the robots that contributes to the overall taxis. We therefore extend the

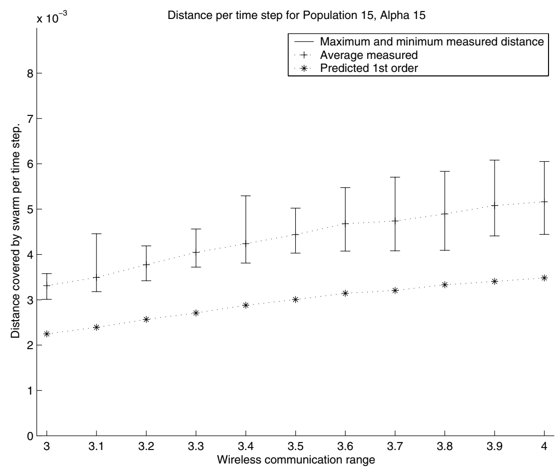

Fig. 7. The measured average D_{swarm}, including minimum and maximum values. The lower line is the PD_{swarm} from model 1.

model by re-visiting some of the initial assumptions that were used in the first part of the analysis.

VI. Formal analysis, step two

This paper initially made a number of assumptions that will now be reconsidered. First, we assumed that an interaction between robots with the same illumination status would be neutral with respect to swarm taxis. Secondly, we assumed that coherence movements would be neutral since they take place in pairs. We will now examine those assumptions further.

The assumption regarding collisions between robots with the same illumination status seems to hold true. However, consider now collisions between robots where one of those robots is in the progressive quasi-state. When a robot is in the progressive state and hence moving towards the beacon, we contend that when it collides with another robot, and thus makes an avoidance movement, the progressive movement is 'transferred' to the robot it collides with. The new robot will now progress towards the beacon, until it must either make a coherence movement or have to avoid yet another robot. We refer to this as 'second order' avoid progress, as the robot is not progressive due to a direct interaction between an illuminated and shadowed robot, but due to avoiding one already in the progressive state. Thus we measure three additional values from the simulation. First, the number of avoidance interactions between robots in the progressive state and other robots, $I_{2nd-avoid}$. We again need to know for how long the robot stays in this state, $t_{2nd-avoid}$ and finally, the average direction of the robot after such an interaction, once again denoted θ. We can then use the same equation as above to calculate the predicted contribution to the swarm movement from this type of interaction,

$$PD_{swarm2a} = \frac{I_{2nd-avoid} * t_{2nd-avoid} * D_{robot} * \theta}{N} \quad (5)$$

where $_{swarm2a}$ refers to second order avoid interaction.

It seems natural to expect that the contribution to swarm taxis from this second order interaction is less than the first order contribution, and this is confirmed from the simulation values. This is mainly for two reasons. The second order avoid interactions take place rarely compared the first order avoid interactions, and the $\sin$ value is much smaller, i.e. the angles of interaction are larger compared to the first order case.

TABLE II

NUMBER OF INTERACTIONS 2ND ORDER AVOID, AVERAGE OF 50 RUNS

r_w	$I_{2nd-avoid}$	$t_{2nd-avoid}$	θ	$PD_{swarm2a}$
3.0	0.33	2.12	0.17	0.00041
3.1	0.33	2.29	0.17	0.00045
3.2	0.34	2.46	0.18	0.00051
3.3	0.35	2.67	0.17	0.00056
3.4	0.35	2.87	0.18	0.00063
3.5	0.36	3.07	0.18	0.00068
3.6	0.36	3.30	0.18	0.00077
3.7	0.37	3.51	0.18	0.00079
3.8	0.37	3.70	0.18	0.00085
3.9	0.37	3.90	0.18	0.00093
4.0	0.38	4.18	0.18	0.00098

Nonetheless, there is a clear bias in the second order avoidance. In table II we can see the same tendency as we see in the first order avoidance, i.e. as the communication range increases, so does the time spent in the 2nd order avoid-state. Once again, this helps to explain why the swarm moves faster with longer communication range.

The second assumption we will revisit is the assumption that coherence movements are always neutral. We will follow the same line of thought as above: when a robot has to perform a coherence move it will contribute to movement towards the beacon if the robot it lost the connection to was in a first order progressive state. Thus we must measure how many such interactions there are per time step, denoted $I_{2nd-coh}$. We must also measure the duration of this state, denoted $t_{2nd-coh}$. From this we calculate the contribution to the swarm movement from this type of interaction.

$$PD_{swarm2c} = \frac{I_{2nd-coh} * t_{2nd-coh} * D_{robot} * \theta}{N} \quad (6)$$

There is one problem with this. We can easily measure from our simulation both $I_{2nd-coh}$ and $t_{2nd-coh}$, and the results are shown in table III. However, due to our simulation setup it is very hard to measure the heading a second order coherence move makes the robot take. For our calculations we have used the same heading as the first order heading, but this is almost guaranteed to be too high. First order interactions can take place almost all over the swarm. We see that the second order avoid sine value is approximately 0.18 whereas the first order sine value varies from 0.47 to 0.41, more than twice the distance covered per time step. Most likely the variations between the second order coherence will be larger. Perhaps a better estimate is to use the second order avoid value, or an

TABLE III

NUMBER OF INTERACTIONS 2ND ORDER COHERENCE, AVERAGE OF 50 RUNS. SINE VALUE OMITTED.

r_w	$I_{2nd-coh}$	$t_{2nd-coh}$	$PD_{swarm2c}$.
3.0	0.35	1.04	0.00057
3.1	0.33	1.04	0.00053
3.2	0.31	1.04	0.00048
3.3	0.28	1.04	0.00044
3.4	0.26	1.05	0.00041
3.5	0.25	1.05	0.00038
3.6	0.23	1.05	0.00036
3.7	0.22	1.05	0.00033
3.8	0.21	1.05	0.00032
3.9	0.20	1.05	0.00030
4.0	0.19	1.05	0.00029

average of the two. This is surely a point that needs further examination.

We can see from table III that the time spent in the coherence state does not increase with increasing communication range. At first this may seem surprising, but remember that when a robot enters the coherence state it turns around to move towards the remainder of the swarm. As it gets closer and consequently increases the number of connections it will make a random turn. Since a robot will turn around as soon as a connection is lost, it will reconnect almost instantly independent of the wireless communication radius. Now that

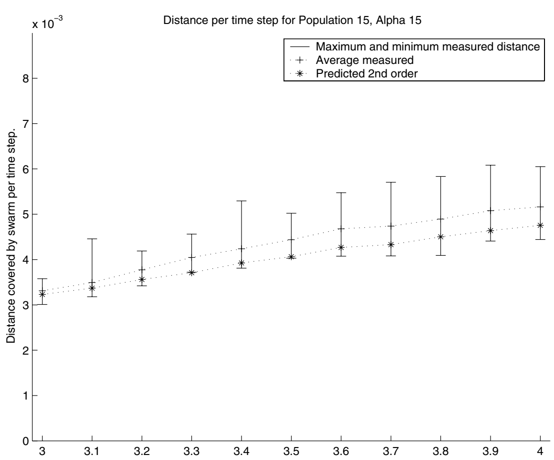

Fig. 8. The measured average D_{swarm}, including minimum and maximum values. The lower line is $PD_{swarm-total}$.

we know the velocity contribution from first order avoid, second order coherence and second order avoid, we can sum them to get net velocity towards the beacon. The predicted swarm movement per time step equals

$$PD_{swarm-total} = PD_{swarm} + PD_{swarm2a} + PD_{swarm2c} \tag{7}$$

When we sum the contribution from all three elements we see a reasonable correlation between the predicted and measured values for swarm velocity as shown in figure 8.

We have also run the same simulations and calculations for a swarm with only ten robots. We will not give the detailed numbers here, but the summarised results are shown in figure 9. We see that the model again predicts swarm velocity with reasonable accuracy for the smaller population size. However,

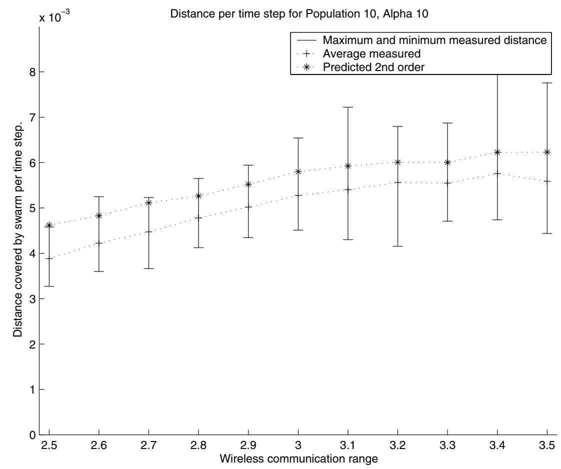

Fig. 9. The measured average D_{swarm} and predicted value $PD_{swarm-total}$ for a swarm with population size $N = 10$. The topmost line is $PD_{swarm-total}$.

we should make a number of critical observations. In the case of population size 15, we always predict a slightly smaller swarm velocity than the measured value. Our predicted values are always between the average and minimum measured values. But when we examine the results for a population size of 10, we tend to predict somewhat higher swarm velocity values. Once again, we are always within the bounds of the natural variations in the experiments, but on the other side of the average. At this stage we can only speculate about why this is so; one possible explanation might be that as the swarm size increases, there will be room for third and perhaps even fourth order effects, that we do not take into account.

VII. CONCLUSION AND FURTHER WORK

In this paper we have described an improved algorithm for emergent taxis in a swarm of wireless connected mobile robots. We have seen how a set of simple behavioural rules for the robots can generate beacon-taxis even though none of the robots have the necessary sensing ability to determine the direction of the beacon. The emergent swarm taxis is a strictly collective property that does not work with less than four or five robots and whose performance improves with increasing swarm size. We have developed an informal analysis of the swarm that explains the micro-level interactions between the robots and how they combine to make swarm taxis possible. We have, with reasonable accuracy, been able to formalize our initial analysis and calculate how much the different types of interactions contribute to the overall velocity of the swarm, and based on this formalization we can predict the velocity of the swarm. Even though we have made several simplifying assumptions, e.g robot turning is instant; complete uniformity of avoidance sensors; no real motion dynamics, the results from this analysis provide us with some confidence that a more advanced model, that takes these issues into account is possible.

Compared to the original algorithm for emergent taxis [7], [8] the algorithm presented here is improved because the robots do not need to have a representation of the connectivity or illumination status of their neighbours. Thus the communication in purely situated, and bandwidth utilization and simplicity have been improved.

There are, however, three problems not addressed so far that should be addressed. Firstly, in this paper we have used our simulation to measure and numerically characterize the different micro-level interactions between the robots. Arguably this makes the results somewhat circular, and the interaction values need to be derived analytically or geometrically if we really want to call our estimate of swarm velocity a prediction. This poses a formidable challenge, particularly since several of the state transition probabilities are conditional, and small errors in estimation can lead to large errors on the macro level. Even though the taxis behaviours in general are very robust, e.g. stable and reliable under relatively large variations on the micro level, the exact velocity of the taxis is very sensitive to small variations on the micro level.

Another problem posed by this work is our simplifying assumption of α = population. This means that the swarm will attempt to be fully connected at all times and this surely goes against the swarm paradigm. One of the greatest benefits of a swarm system is exactly that communication is *local*, which in turn allows swarm systems to scale well. We have made this assumption to make the analysis simpler, but further work will need to remove this assumption and hence apply to wireless connected swarms with local communication only.

The final problem is generalisability. The analysis and model presented in this paper is very specific to this particular swarm system. We wonder if it would be possible to make models that are more general, and would be able to capture the underlying mechanisms for taxis, even if taxis was achieved by using different behavioural rules. However, there is a possibility that different swarm taxis implementations will be too different to be captured by a similar model.

Notwithstanding these self-critical observations, we believe that the *methodology* presented in this paper is of generic value: that is a reductive approach that identifies both first and second order micro-level pair-wise interactions between robots and then attempts to quantify the contribution of each such interaction to the overall swarm behaviour.

A. We need real robots

The swarm taxis algorithm has to date been evaluated only in simulation. Furthermore, although simulated in 2D, the algorithm is inherently dimensionless, and should in principle work in 3D just as well (providing the robots have sufficient controllable degrees of freedom).

We aim to validate both the algorithm and the model using both 2D and 3D robots. For 2D experiments we have a fleet of wheeled Linux-based robots already built in our laboratory. For 3D experiments we are currently upgrading a small swarm of lighter-than-air aerobots.

AKNOWLEDGEMENTS.

We would like to thank Wenguo Liu and Christopher Harper for many useful and interesting discussions on this topic during our Friday research lunches at the UWE campus, Bristol.

REFERENCES

[1] R. Brooks and A. M. Flynn, “Fast, cheap and out of control: A robot invasion of the solar system,” *Journal of The British Interplanetary Society*, vol. 42, pp. 478–485, 1989.
[2] E. Bonabeau, M. Dorigo, and G. Theraluaz, *Swarm Intelligence: From Natural to Artificial Systems.* Santa Fe Studies in the Sciences of Complexity, Oxford University Press., 1999.
[3] P. E. Rybski, A. Larson, H. Veeraraghavan, M. LaPoint, and M. Gini, “Communication strategies in multi-robot search and retrieval: Experiences with mindart,” *Proceedings of DARS’04, the 7th International Symposium on Distributed Autonomous Robotic Systems*, pp. 301–310, 2004.
[4] O. E. Holland and C. Melhuish, “Stigmergy, self-organisation, and sorting in collective robotics.” *Artificial Life*, vol. 5:2, pp. 173–202, 1999.
[5] C. Parker, H. Zang, and C. Kube, “Blind bulldozing: multiple robot nest construction,” *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 2, pp. 2010–2015, 2003.
[6] A. Martinoli, K. Easton, and W. Agassounon, “Modeling Swarm Robotic Systems: A Case Study in Collaborative Distributed Manipulation,” *Int. Journal of Robotics Research*, vol. 23, no. 4, pp. 415–436, 2004, special Issue on Experimental Robotics, B. Siciliano, editor.
[7] J. Nembrini, A. F. Winfield, and C. Melhuish, “Minimalist coherent swarming of wireless networked autonomous mobile robots.” *Proceedings of the seventh international conference on simulation of adaptive behaviour on From animals to animats.*, pp. 373–382, 2002.
[8] J. Nembrini, *Minimalist Coherent Swarming of Wirless Networked Autonomous Mobile Robots.* PhD thesis, University of the West of England, 2005.
[9] M. Minsky, “Our robotized future,” *Robotics, Omni Publications International*, pp. 268–307, 1985.
[10] R. A. F. Jr., *Nanomedicine, Volume I: Basic Capabilities.* Landes Bioscience, Georgetown, TX, 1999.
[11] K. Lerman, A. Martinoli, and A. Galstyan, “A review of probabilistic macroscopic models for swarm robotic systems,” *Proc. of the Swarm Robotics Workshop at the Eight Int. Conference on the Simulation of Adaptive Behavior SAB-04,*, 2004.
[12] E. Sahin, “Swarm robotics: From sources of inspiration to domains of application,” in *Swarm Robotics Workshop: State-of-the-art Survey*, ser. Lecture Notes in Computer Science, E. Sahin and W. Spears, Eds., no. 3342. Berlin Heidelberg: Springer-Verlag, 2005, pp. 10–20.
[13] G. Beni, “From swarm intelligence to swarm robotics,” in *Swarm Robotics Workshop: State-of-the-art Survey*, E. Sahin and W. Spears, Eds., no. 3342. Berlin Heidelberg: Springer-Verlag, 2005, pp. 1–9.
[14] K. Stoy, “Using situated communication in distributed autonomous mobile robotics,” in *SCAI ’01: Proceedings of the Seventh Scandinavian Conference on Artificial Intelligence*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2001, pp. 44–52.