Noname manuscript No.
(will be inserted by the editor)

Modelling a Wireless Connected Swarm of Mobile Robots

Alan FT Winfield · Wenguo Liu · Julien Nembrini · Alcherio Martinoli

Received: date / Accepted: date

Abstract It is a characteristic of swarm robotics that modelling the overall swarm behaviour in terms of the low-level behaviours of individual robots is very difficult. Yet if swarm robotics is to make the transition from the laboratory to real-world engineering realisation such models would be critical for both overall validation of algorithm correctness and detailed parameter optimisation. We seek models with predictive power: models that allow us to determine the effect of modifying parameters in individual robots on the overall swarm behaviour. This paper presents results from a study to apply the probabilistic modelling approach to a class of wireless connected swarms operating in unbounded environments. The paper proposes a probabilistic finite state machine (PFSM) that describes the network connectivity and overall macroscopic behaviour of the swarm, then develops a novel robot-centric approach to the estimation of the state transition probabilities within the PFSM. Using measured data from simulation the paper then carefully validates the PFSM model step by step, allowing us to assess the accuracy and hence the utility of the model.

Keywords Swarm Robotics · Modelling · Wireless Ad-hoc Network

1 Introduction

A robotic swarm is an example of a stochastic, dynamical and often non-linear system. Developing mathematical models that allow overall swarm properties to be predicted

Alan FT Winfield and Wenguo Liu
Bristol Robotics Laboratory
University of the West of England
Coldharbour Lane, Bristol BS16 1QY, UK
Tel: +44 (0)117 328 3159
Fax: +44 (0)117 328 2734
E-mail: {Alan.Winfield, Wenguo.Liu}@uwe.ac.uk

Julien Nembrini and Alcherio Martinoli
Distributed Intelligent Systems and Algorithms Laboratory
École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland
E-mail: {Julien.Nembrini, Alcherio.Martinoli}@epfl.ch

from the low-level microscopic parameters of the individual robots that comprise the swarm is challenging. For this reason many swarm robotics algorithms are validated with reference to simulation studies only, with no underpinning mathematical model or proof. This approach is inherently limited since simulation can only explore small parts of a system's parameter space, and hence provide only weak 'inductive' proof of an algorithm's correctness. Yet if swarm robotic systems are to find real-world application, especially in safety- or mission-critical applications (Rouff et al. 2003; Truszkowski et al. 2004; Winfield et al. 2006), we need the strong validation provided by mathematical or formal models.

In recent years probabilistic approaches to modelling swarm robotic systems have been developed and successfully applied. One way to classify these is based on their representation of the swarm and its units. *Microscopic* models reproduce each real robot in the targeted system separately, with dedicated — more or less detailed — representations. *Macroscopic* models instead reproduce the target swarm robotic system with a single representation, for instance summarising fractions or total numbers of robots in the swarm engaged in specific tasks. One of the first examples of probabilistic modelling of a swarm of robots at the microscopic level is that proposed by Martinoli et al. (1999) to study object aggregation; robot's interactions with other robots and the environment are modelled as a series of stochastic events, with probabilities determined by simple geometric considerations and systematic experiments with one or two real robots. The very same microscopic method was applied to the analysis of collaborative stick pulling (Ijspeert et al. 2001).

Probabilistic macroscopic approaches have often adopted a mean field approach, as widely used in physics, chemistry, biology and the social sciences, to directly describe the collective behaviour of the robotic swarm. The macroscopic approach relies on the assumption that systems in which the individuals' behaviour is stochastic may have statistically predictable overall properties. In swarm robotics macroscopic models have been used to study the effect of interference in a swarm of foraging robots (Lerman and Galstyan 2002), collaborative stick-pulling (Lerman et al. 2002; Martinoli et al. 2004) and object aggregation (Martinoli et al. 1999; Agassounon et al. 2001, 2004; Kazadi et al. 2004). A review of macroscopic models is given in Lerman et al. (2005). Recently, new contributions leveraging hybrid system theory and including probabilistic macroscopic modeling have been proposed. These approaches are particularly promising since they allow for capturing both the continuous physical motion aspect as well as the discrete control logic of a swarm of robots. For instance, Berman et al. (2007) described a macroscopic approach for modelling a biologically inspired algorithm for robot colony nest-site selection; Milutinovic and Lima (2006) presented a method for integrating macroscopic modelling with optimal centralised control.

Not all macroscopic models are designed in the same way and for the same purpose. For instance, typically in the natural sciences macroscopic models are designed with the principle of parsimony, i.e. to be as simple as possible in order to address specific questions about the collective behaviour of a target system. In engineering, models might also be used as a tool for engineering the system. In the specific framework of swarm robotic systems, macroscopic models should also help to engineer the individual agents in order to obtain a desired collective behaviour; a good example is represented by work reported in Martinoli et al. (2004). Macroscopic models have been designed step-by-step based on incremental abstraction starting from very detailed microscopic models (e.g., reproducing very faithfully intra-robot details such as individual sensor and actuator characteristics), with a very precise mapping between different incre-

mental abstraction levels. However, in general, macroscopic models belong to a suite of models characterised by different abstraction, parameterisation, and computational cost and they all serve as potential tools for designing and optimising the target swarm robotic system.

Probabilistic modelling approaches typically rely on two assumptions. Firstly, that each robots' future state depends only on its present state and perhaps also the time it has spent in that state (the semi-Markov model); this assumption is true when robots use reactive control, possibly extended with simple memory capabilities, and can be treated as finite state automata. Secondly, that robots are uniformly distributed in their operational area. These assumptions mean that for robots operating in spatially bounded arenas it is fairly straightforward to geometrically estimate the probability of a robot making a transition from one state to another. The latter assumption is challenged by the robotic system addressed in this paper since our swarm operates in unbounded space. Since we are modelling a wireless connected swarm, we seek a model that manifests its overall connectivity structure. For this reason we adopt, in this paper, the macroscopic modelling approach incrementally constructed from available microscopic information about software (e.g., control structure) and hardware (perception and communication ranges), following and extending the method proposed in Martinoli et al. (2004).

This paper proceeds as follows. Section 2 introduces and summarises the wireless connected swarm algorithm that we are seeking to model. Section 3 then proposes a macroscopic probabilistic model of this wireless connected swarm. The probabilistic model requires us to estimate a number of state transition probabilities and section 4 develops a novel *robot-centric* geometrical approach. Section 5 validates the estimated transition probabilities with reference to a simulated swarm. Section 6 brings together the work of sections 3 and 4 to run the full model and compare its predicted swarm connectivity structure with the structure measured from simulation. Finally, section 7 reviews the model, its utility, limitations and potential for further development.

2 A Wireless Connected Swarm

We have developed a class of algorithms which make use of local wireless connectivity information alone to achieve swarm aggregation (Nembrini et al. 2002; Nembrini 2005). These algorithms make use of situated communications (Støy 2001), in which connectivity information 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 even in unbounded space, and thirdly, the connectivity needed for and generated by the algorithms means that the swarm naturally forms an *ad-hoc* communication network. Such a network would be a significant advantage in many swarm robotics applications such as distributed sensing, exploration or mapping, since it would allow data to be communicated between any two robots and facilitate data collection from the whole swarm via a single connection with just one robot. The algorithm requires that connectivity information is transmitted via only a single hop. Each robot broadcasts only its own ID and, since the maximum number of neighbours a real robot can have is physically constrained and the same for a swarm of 100 or 10,000 robots, the algorithm scales linearly for increasing swarm size. The algorithm thus meets the criteria for swarm robotics articulated by Şahin (2005) and Beni (2005).

We have a swarm of homogeneous and relatively incapable robots with only local sensing and communication capabilities which is *scalable* and highly *robust* (i.e. tolerant to failures, Winfield and Nembrini (2006)). Furthermore, we observe *flexibility* to its environment in that our wirelessly connected swarm demonstrates collective obstacle avoidance and - when equipped with simple beacon sensors - collective taxis towards a beacon and collective beacon encapsulation (Nembrini 2005).

The basic premise of the algorithm is that each robot has range-limited wireless communication which, for simplicity, we model as as circle of radius R_w with the robot at its centre. The boundary of the circle represents the threshold beyond which another robot is out of range. Each robot also has collision avoidance sensors with a range R_a, where $R_a < R_w$. The basic algorithm, which we refer to as the *α-algorithm*, is very simple. The default behaviour of a robot is forward motion. While moving each robot periodically broadcasts an 'I am here' message. The message will of course be received only by those robots that are within wireless range: its neighbours. If the number of a robot's neighbours should fall below the threshold α then it assumes it is moving out of the swarm and will execute a 180° turn. When the number of neighbours rises above α (i.e. when the swarm is regained) the robot then executes a random turn. This is to avoid the swarm simply collapsing on itself. We say that the swarm is *coherent* if any break in its overall connectivity lasts less than a given time constant. Coherence gives rise to both swarm aggregation and a (coherent) connected *ad hoc* wireless network. In the interests of simplicity we can consider each robot as having three basic behaviours, or states: move *forward* (default); *coherence*, triggered by the number of neighbours falling below α, and *avoidance*, triggered by the robot's collision (proximity) sensor.

The robot updates its connectivity information less frequently than its proximity sensor data. The ratio of connectivity sampling rate to the sampling rate of proximity sensors, which we refer to as *cadence*, is introduced into the basic α-algorithm to prevent the robot from updating its connectivity state too frequently (we need to give the robot time to complete its turn in response to a connection loss, for example, before re-checking its connectivity). By default, the robot will move forward at a fixed velocity. It will update its connectivity state after a certain duration, say T_C (steps), and if it finds the number of connected neighbours has dropped below the threshold α, then it will move into the *coherence* state and execute the U-turn behaviour to try to recover the lost connections; if and when the number of connected neighbours then increases, the robot will execute a random turn. Providing the number of connected neighbours remains at or above α, the robot can lose or gain connections but remain in the *forward* state. Thus, depending upon its connectivity, a robot will either remain in the *forward* state or switch between *forward* and *coherence* states unless it collides with other robots (triggered by the proximity sensor). Such an event will cause the robot to move into the *avoidance* state and execute a collision avoiding turn for time T_A (steps), after which the robot will return to its previous *forward* or *coherence* state. Note that changes in connectivity take precedence over collision avoidance, thus if a change of connectivity is detected while the robot is in the *avoidance* state (i.e. taking avoiding action), the robot will - if required - immediately transition into the appropriate coherence or forward behaviours.

Figure 1 shows the basic robot Finite State Machine (FSM). We reflect the fact that the avoidance behaviours are subsumed within the two top-level states *coherence* and *forward* by showing sub-states $avoidance^C$ and $avoidance^F$. Note that although changes in connectivity take precedence, because the proximity sensor is sampled more

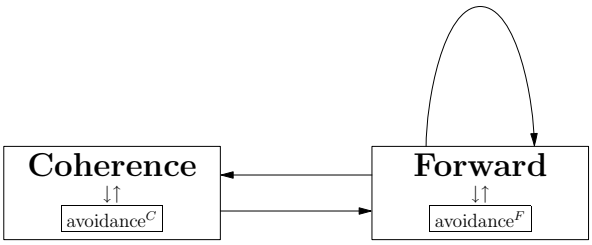

Fig. 1 Robot Finite State Machine.

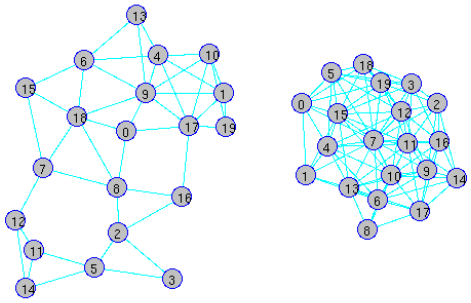

Fig. 2 Swarm with $\alpha = 5$ (left) and $\alpha = 10$ (right). Lines between robots indicate wireless connections.

frequently than the connectivity (defined above as cadence) collision avoidance is still assured.

In fact, this algorithm has the serious limitation that it is unable to prevent the swarm splitting into smaller swarms. When, for example, two subnets joined by only one connection form, the α-algorithm cannot prevent the possibility of the swarm splitting into two. This limitation is completely overcome by the more sophisticated 'Shared Neighbour Algorithm' (Nembrini et al. 2002). However, the α-algorithm does achieve useful swarm coherence in which a larger value of α results in a smaller densely connected swarm and a smaller value of α in a larger, loosely connected swarm, as shown in Figure 2. Because of its simplicity, and to test the approach presented in this paper, we have chosen to initially model the α-algorithm.

3 A Probabilistic Model of Connectivity

Following the probabilistic modelling methodology of Martinoli et al. (2004) and Correll and Martinoli (2004), we describe the individual robot controller as a probabilistic finite state machine (PFSM). Our starting point is the individual robot controller implemented on the target system (see Figure 1). Since the transitions from one behaviour to another are determined by interactions within a robot's neighbourhood, we must differentiate and update states which allow us to keep track of these changes in order to predict the overall dynamics of the system. Notice that the state variables to be

considered are dependent on the scenario and the metrics considered. For instance, in Martinoli et al. (2004) robots were engaged in a distributed manipulation task and therefore, in addition to state variables related to the control modes of robots, the model had to keep track of states relative to objects which could be manipulated. In Correll and Martinoli (2004), robots were involved in a distributed sensing task and in order to properly calculate the inspection metrics, state variables related to the coverage status of objects (not measurable by the robots themselves directly) had to be introduced. In this paper we seek to model a wirelessly connected network thus we seek a PFSM that explicitly models the wireless connectivity.

In the α-algorithm described in the previous section $robot_i$ has a number of connected neighbours d_i. Clearly, the range of values for d_i is bounded. The maximum value d_{max} is determined geometrically by the ratio of the areas covered by the wireless sensor range and the avoidance sensor range; the estimation of d_{max} is given in Appendix A. For the robot to remain in the default *forward* state, the lower bound on d_i is α. Now in the α-algorithm when $d_i < \alpha$ the robot moves into the *coherence* state in which it turns back to try and recover the swarm and hence bring d_i back to a value greater than or equal to α. However, the coherence behaviour is not always successful and it is possible for a robot to have fewer than $(\alpha-1)$ connections (Figure 2:left clearly shows a number of robots with only 2 or 3 connections). In fact, the robot will continue to try and recover the swarm for values of $0 < d_i < \alpha$. Based on these observations we can now propose a PFSM which completely models the swarm connectivity, as shown in Figures 3 and 4.

Figure 3 is, in effect, the simple FSM of Figure 1 expanded to show every possible number of network connections in each of the two states *coherence* and *forward*, together with every possible transition between the states and their probabilities. Each of the discrete *forward* states represents a different value of d_i; the $\overline{F}_m$ state is the state with the maximum number of connections d_{max}, the $\overline{F}_{m-1}$ state is the state with $d_{max} - 1$ connections, counting down until we reach the $\overline{F}_0$ state with 0 connections; there are a total of $d_{max} + 1$ *forward* states, including $\overline{F}_0$. Note that $\overline{F}_0$ is the 'lost robot' state representing the failure of the algorithm to maintain the coherence of the swarm. Consider the *forward* state $\overline{F}_\alpha$. The loss of a connection with probability P_{l_α} will cause a transition into the *coherence* state $\overline{C}_{\alpha-1}$. If the action of that state is successful then the robot will transition, after T_C steps and with recovery probability $P_{r_{\alpha-1}}$ back into the $\overline{F}_\alpha$ state. If, on the other hand, the coherence behaviour fails, the robot will move into the $\overline{F}_{\alpha-1}$ state. The likelihood of this is the coherence failure probability $P_{f_{\alpha-1}}$. A loss of connection in each of the forward states $\overline{F}_1 \ldots \overline{F}_\alpha$ will trigger a transition into *coherence* states $\overline{C}_0 \ldots \overline{C}_{\alpha-1}$ respectively. Note that if the coherence behaviour were to always succeed then the recovery probability $P_r = 1$, the coherence failure probability $P_f = 0$ and all of the states below the dashed line in Figure 3 would disappear.

Figure 4 completes the PFSM by expanding the two states $\overline{C}_i$ and $\overline{F}_i$ into their respective sub-states, again reflecting the structure of the FSM of figure 1. Figure 4:right shows that a robot in each of the *forward* states F_i might collide with another robot, with probability P_{a_i}, triggering a transition into its corresponding *avoidance* state A_i^F, returning to the initial *forward* state after T_A steps. Similarly, figure 4:left shows that a robot in each of the *coherence* states might also collide with another robot triggering its transition into corresponding *avoidance* states A_i^C, also returning after T_A steps. Table 1 summarises each of the state transition probabilities in the full PFSM.

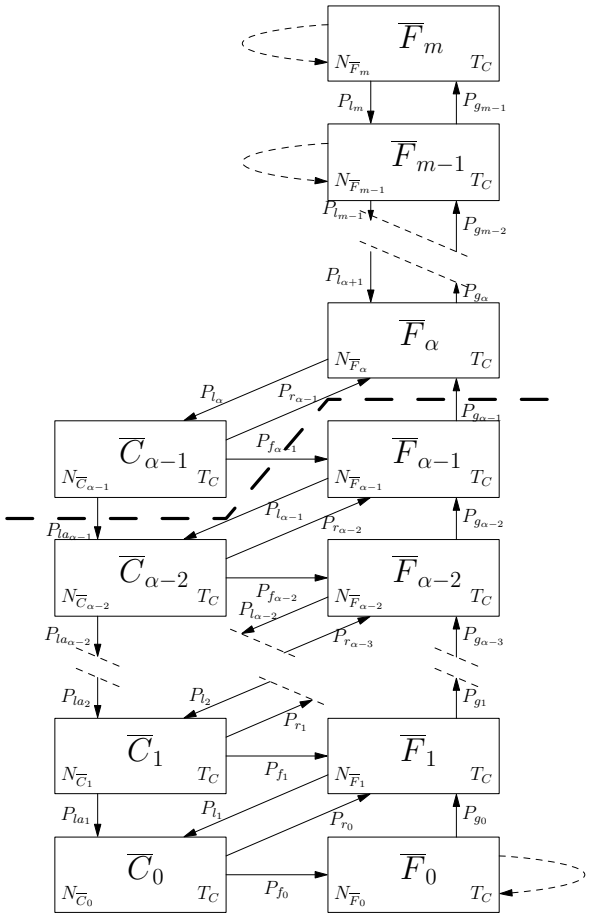

Fig. 3 PFSM of the robot controller. $\overline{F}_i$ represents the *forward* state with i connections; $\overline{C}_i$ represents the *coherence* state with i connections. $N_{\overline{C}_i}$ and $N_{\overline{F}_i}$ indicate the average number of robots in corresponding states and T_C indicates the number of time steps spent in each state. States below the heavy dashed line would not exist if the coherence behaviour were always successful.

Fig. 4 Left: *coherence* state $\overline{C}_i$ expanded to show sub-states A_i^C and C_i. **Right:** *forward* state $\overline{F}_i$ expanded to show sub-states A_i^F and F_i. The average number of robots in each state is shown as $N_.$; T_A is the number of time steps spent in the *avoidance* states A_i^C and A_i^F.

Table 1 State transition probabilities, d_i represents the number of connections for $robot_i$.

probabilities	comments
$P_{a_{d_i}}$	collision with another robot
$P_{l_{d_i}}$	loss of a connection in *forward* state
$P_{g_{d_i}}$	gain of a connection
$P_{r_{d_i}}$	recovery of a connection
$P_{f_{d_i}}$	failure to recover a connection
$P_{la_{d_i}}$	loss of a connection in *coherence* state

3.1 Macroscopic model

We can write down a number of macroscopic difference equations (DEs) for the state transitions from Figures 3 and 4. We adopt the notation that $N_{X_i}(k)$ denotes the average number of robots in the swarm in state X at time step k with i connections.

First consider the two sub-PFSMs of Figure 4. By definition,

$$N_{\overline{F}_i}(k) = N_{A_i^F}(k) + N_{F_i}(k) \tag{1}$$

where $0 < i \leq d_{max}$, and

$$N_{\overline{C}_i}(k) = N_{A_i^C}(k) + N_{C_i}(k) \tag{2}$$

where $0 < i < \alpha$. When $i = 0$, $N_{\overline{F}_0}(k) = N_{F_0}(k)$ and $N_{\overline{C}_0}(k) = N_{C_0}(k)$.

We can express the number of robots in the next time step $k+1$ as functions of the number of robots in the current time step k for the two sub-PFSMs in Figure 4. For the *avoidance* state A_i^F we have,

$$N_{A_i^F}(k+1) = N_{A_i^F}(k) + P_{a_i}N_{F_i}(k) - P_{a_i}N_{F_i}(k - T_A) \tag{3}$$

where $0 < i \leq d_{max}$. Similarly, for the *avoidance* state A_i^C,

$$N_{A_i^C}(k+1) = N_{A_i^C}(k) + P_{a_i}N_{C_i}(k) - P_{a_i}N_{C_i}(k - T_A) \tag{4}$$

where $0 < i < \alpha$.

Note that this is a time-discrete model with a sampling interval T. As is conventional we omit the T, i.e. $N_{X_i}(k)$ is equivalent to $N_{X_i}(kT)$. T is determined by the proximity sensor sampling rate as defined in Section 5. The second entry on the right hand side of Equation (3) represents the number of robots whose proximity sensors have been triggered and are transferring into *avoidance* state A_i^F, and the third entry represents those that have completed the avoidance behaviour, returning to state F_i after T_A steps.

Now, as described previously, robots update their connectivity and proximity sensor data at different rates, i.e. T_C and 1 steps, respectively. Consider the top-level PFSM of Figure 3. In order to model the fact that the connectivity is sampled every T_C steps, which means that robots can transition between states $\overline{F}_i$ and $\overline{C}_i$ only every $T_C T$ seconds, we now multiply k by T_C. Thus we can write DEs for robots in the $\overline{F}_i$ state as follows,

– when $i = 0$

$$N_{\overline{F}_0}((k+1)T_C) = N_{\overline{F}_0}(kT_C) + P_{f_0} N_{\overline{C}_0}(kT_C) - P_{g_0} N_{\overline{F}_0}(kT_C) \tag{5}$$

– when $0 < i < \alpha$

$$\begin{aligned} N_{\overline{F}_i}((k+1)T_C) = N_{\overline{F}_i}(kT_C) + P_{g_{i-1}} N_{\overline{F}_{i-1}}(kT_C) + P_{f_i} N_{\overline{C}_i}(kT_C) \\ + P_{r_{i-1}} N_{\overline{C}_{i-1}}(kT_C) - (P_{g_i} + P_{l_i}) N_{\overline{F}_i}(kT_C) \end{aligned} \tag{6}$$

– when $i = \alpha$

$$\begin{aligned} N_{\overline{F}_\alpha}((k+1)T_C) = N_{\overline{F}_\alpha}(kT_C) + P_{g_{\alpha-1}} N_{\overline{F}_{\alpha-1}}(kT_C) + P_{l_{\alpha+1}} N_{\overline{F}_{\alpha+1}}(kT_C) \\ + P_{r_{\alpha-1}} N_{\overline{C}_{\alpha-1}}(kT_C) - (P_{g_\alpha} + P_{l_\alpha}) N_{\overline{F}_\alpha}(kT_C) \end{aligned} \tag{7}$$

– when $\alpha < i < d_{max}$

$$\begin{aligned} N_{\overline{F}_i}((k+1)T_C) = N_{\overline{F}_i}(kT_C) + P_{g_{i-1}} N_{\overline{F}_{i-1}}(kT_C) + P_{l_{i+1}} N_{\overline{F}_{i+1}}(kT_C) \\ - P_{g_i} N_{\overline{F}_i}(kT_C) - P_{l_i} N_{\overline{F}_i}(kT_C) \end{aligned} \tag{8}$$

– when $i = d_{max}$

$$N_{\overline{F}_m}((k+1)T_C) = N_{\overline{F}_m}(kT_C) + P_{g_{m-1}} N_{\overline{F}_{m-1}}(kT_C) - P_{l_m} N_{\overline{F}_m}(kT_C) \tag{9}$$

Similarly, for robots in the $\overline{C}_i$ state we have,

– when $i = 0$

$$\begin{aligned} N_{\overline{C}_0}((k+1)T_C) = N_{\overline{C}_0}(kT_C) + P_{la_1} N_{\overline{C}_1}(kT_C) + P_{l_1} N_{\overline{F}_1}(kT_C) \\ - (P_{r_0} + P_{f_0}) N_{\overline{C}_0}(kT_C) \end{aligned} \tag{10}$$

– when $0 < i < \alpha - 1$

$$\begin{aligned} N_{\overline{C}_i}((k+1)T_C) = N_{\overline{C}_i}(kT_C) + P_{la_{i+1}} N_{\overline{C}_{i+1}}(kT_C) + P_{l_{i+1}} N_{\overline{F}_{i+1}}(kT_C) \\ - (P_{r_i} + P_{f_i} + P_{la_i}) N_{\overline{C}_i}(kT_C) \end{aligned} \tag{11}$$

– when $i = \alpha - 1$

$$\begin{aligned} N_{\overline{C}_{\alpha-1}}((k+1)T_C) = N_{\overline{C}_{\alpha-1}}(kT_C) + P_{l_\alpha} N_{\overline{F}_\alpha}(kT_C) \\ - (P_{r_{\alpha-1}} + P_{f_{\alpha-1}} + P_{la_{\alpha-1}}) N_{\overline{C}_{\alpha-1}}(kT_C) \end{aligned} \tag{12}$$

Clearly, the total number of robots in different states remains constant. If there are N robots in the swarm in total then we have:

$$N = \sum_{i=1}^{\alpha-1} N_{A_i^C}(k) + \sum_{i=1}^{m} N_{A_i^F}(k) + \sum_{i=0}^{m} N_{F_i}(k) + \sum_{i=0}^{\alpha-1} N_{C_i}(k) \tag{13}$$

With the correct initial conditions Equations (1) to (12) may be numerically evaluated. Values for $\overline{N}_F$ and $\overline{N}_C$ will change only every T_C time steps, whereas values for sub-PFSM states $N_{A_i^C}$, N_{C_i}, $N_{A_i^F}$ and N_{F_i} may change every time step. In this way we model a controller with different sample rates for the proximity sensor and the connectivity.

4 Geometrical estimation of transition probabilities

In order to solve the equations in section 3.1 we must estimate each of the state transition probabilities. We now develop a geometry-based approach to the estimation of these probabilities in our wirelessly connected swarm. Let V denote the normal forward speed of each robot. It follows that the relative speed between two robots varies from 0 to $2V$ and the relative heading varies from 0° to 360°. Consider one of the robots in the swarm, say $robot_i$, with d_i neighbours at time step k.

Figure 5 illustrates some of its neighbours, shown as $robot_A$, $robot_B$, $robot_C$ and $robot_D$. Let us assume that $robot_i$ is in either *forward* or *coherence* states, then after one time step (of duration T), each of its neighbours will move a distance from 0 to $2VT$. It is clear that only the robots close enough will have a chance of moving into $robot_i$'s collision area (within radius R_a, marked $\mathbb{C}$ in Figure 5), and thus drive $robot_i$ to change to state *avoidance*. For instance, as shown in Figure 5, $robot_A$ may possibly trigger $robot_i$'s avoidance sensor next time step while $robot_B$, $robot_C$ and

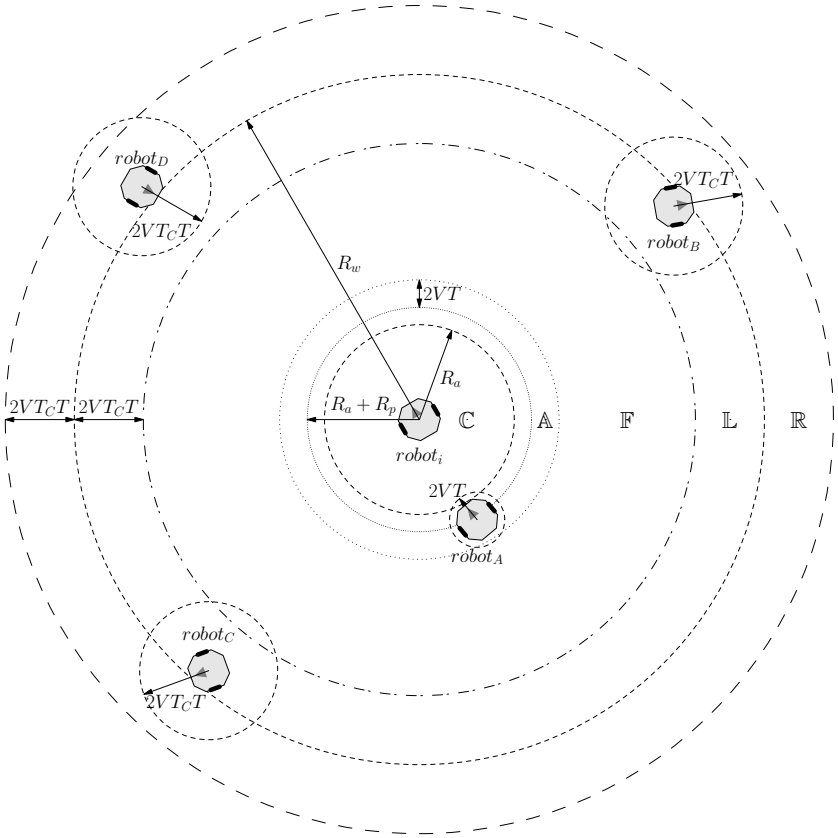

Fig. 5 $Robot_i$ and its neighbours. Robots are marked with filled circles. Each robot has a communication range R_w and avoidance radius R_a, R_p denotes the physical size (radius) of the robot. $\mathbb{C}$, $\mathbb{A}$, $\mathbb{F}$, $\mathbb{L}$ and $\mathbb{R}$ in the figure represent the collision, avoidance, forward, connection loss and connection recovery areas respectively; each is an annular region bounded within two circles with the same origin.

$robot_D$ cannot. Similarly, after T_C steps, $robot_C$ located in area $\mathbb{L}$ will possibly move out of $robot_i$'s communication range resulting in $robot_i$ losing one connection, and $robot_D$ located in area $\mathbb{R}$ might move into $robot_i$'s communication range, with some probability, in which case $robot_i$ will gain one connection at step $k + T_C$. However, $robot_B$ located in area $\mathbb{F}$ can neither trigger $robot_i$'s avoidance sensor nor cause a change in the number of its connected neighbours. Thus, in order to estimate state transition probabilities, we need only consider situations where neighbouring robots fall within the annular regions in Figure 5: $\mathbb{A}$, in which a collision might occur; $\mathbb{L}$, in which a connection loss might occur; or $\mathbb{R}$, in which a connection recovery might occur.

4.1 Avoidance probability $P_{a_{d_i}}$

Assume all the connected neighbours of $robot_i$ are dispersed uniformly within the communication radius. As shown in Figure 5, then the probability of one robot being within area $\mathbb{A}$ can be depicted as the ratio of two corresponding areas:

$$P_{in\_A} = \frac{(R_a + R_p + 2VT)^2 - (R_a + R_p)^2}{R_w^2 - (R_a + R_p)^2} \tag{14}$$

Let P_{a-c} denote the probability of the robot located in area $\mathbb{A}$ moving into $robot_i$'s collision area $\mathbb{C}$. Then the probability of at least one robot moving into $robot_i$'s collision area, resulting in $robot_i$ changing its state to *avoidance*, is:

$$P_{a_{d_i}} = 1 - (1 - P_{a-c}P_{in\_A})^{d_i} \tag{15}$$

4.2 Probabilities of connection loss or gain $P_{l_{d_i}}$, $P_{g_{d_i}}$, $P_{f_{d_i}}$, $P_{r_{d_i}}$ and $P_{la_{d_i}}$

To determine probabilities of connection loss or gain, we need to estimate the change in the number of robots in areas $\mathbb{L}$ and $\mathbb{R}$ based on the geometrical approach above. If only one robot moves from area $\mathbb{L}$ to $\mathbb{R}$ and no robot moves from $\mathbb{R}$ to $\mathbb{L}$, then $robot_i$ will lose a connection at step $k + T_C$. Since all robots act autonomously, $robot_i$ might possibly lose connections with several robots while gaining new connections with several others. Clearly $robot_i$ will experience a net connection loss by the next update connection at step $k + T_C$ only when more robots have moved out of the connection radius than have moved in. Assume the robot distribution density in $robot_i$'s communication range and its vicinity are equal, then the number of robots within area $\mathbb{R}$ at step k, denoted with d_i', is given by

$$d_i' = \left\lfloor \frac{(R_w + 2VT_CT)^2 - R_w^2}{R_w^2} (d_i + 1) \right\rfloor \tag{16}$$

Table 2 shows all possible situations in which $robot_i$ will lose one connection at step $k + T_C$, where $n = \min(d_i, d_i' + 1)$.

For each neighbour of $robot_i$, if P_{in_L} denotes the probability it is in area $\mathbb{L}$, P_{l-r} denotes the probability it will move from area $\mathbb{L}$ to $\mathbb{R}$ at step $k + T_C$, and P_{r-l} denotes the probability the robot in area $\mathbb{R}$ will move from area $\mathbb{R}$ to $\mathbb{L}$, then

$$P_{in\_L} = \frac{R_w^2 - (R_w - 2VT_CT)^2}{R_w^2 - (R_a + R_p)^2} \tag{17}$$

Table 2 $robot_i$ single connection loss.

situation s	lost connections	gained connections	probabilities
1	1	0	$P^{(1)}_{l_{d_i}}$
2	2	1	$P^{(2)}_{l_{d_i}}$
...	...	...	...
n	n	n-1	$P^{(n)}_{l_{d_i}}$

Moreover, as we will discuss in next section, the value of P_{l-r} and P_{r-l} are different in state *forward* and *coherence*, if $P_{F_{l-r}}$ ($P_{C_{l-r}}$) and $P_{F_{r-l}}$ ($P_{C_{r-l}}$) represent the corresponding probabilities for the robots in state *forward* (*coherence*), then $P^{(1)}_{l_{d_i}}$ in Table 2 is

$$P^{(1)}_{l_{d_i}} = C(d_i,1)P_{in\_L}P_{F_{l-r}}(1-P_{in\_L}P_{F_{l-r}})^{d_i-1}\cdot C(d'_i,0)P^0_{F_{r-l}}(1-P_{F_{r-l}})^{d'_i} \quad (18)$$

where

$$C(d_i,s) = \frac{d_i!}{(d_i-s)!s!} \quad (19)$$

Let

$$\begin{aligned} P_1 &= P_{in\_L}P_{F_{l-r}} \\ P_2 &= P_{F_{r-l}} \end{aligned} \quad (20)$$

then

$$P^{(1)}_{l_{d_i}} = C(d_i,1)P_1(1-P_1)^{d_i-1}\cdot C(d'_i,0)(1-P_2)^{d'_i} \quad (21)$$

Similarly, for situation s ($s = 1,2,\ldots,n$) in Table 2,

$$P^{(s)}_{l_{d_i}} = C(d_i,s)P_1^s(1-P_1)^{d_i-s}\cdot C(d'_i,s-1)P_2^{s-1}(1-P_2)^{d'_i-s+1} \quad (22)$$

Clearly the probability that $robot_i$ experiences a net loss of one connection after T_C time steps is the sum of probabilities for all situations in Table 2, thus

$$P_{l_{d_i}} = \sum_{s=1}^{n} P^{(s)}_{l_{d_i}} \quad (23)$$

Using the same method we can calculate the remaining transition probabilities for the PFSM model as follows:

– probability of connection gain $P_{g_{d_i}}$

$$\begin{aligned} P_{g_{d_i}} &= \sum_{s=0}^{n} P^{(s)}_{g_{d_i}} \\ P^{(s)}_{g_{d_i}} &= C(d_i,s)P_1^s(1-P_1)^{d_i-s}\cdot C(d'_i,s+1)P_2^{s+1}(1-P_2)^{d'_i-s-1} \\ n &= \min(d_i, d'_i-1) \end{aligned} \quad (24)$$

and if $robot_i$ is in the *coherence* state then:

– probability of connection recovery $P_{r_{d_i}}$

$$\begin{aligned} P_{r_{d_i}} &= \sum_{s=0}^{n} P_{r_{d_i}}^{(s)} \\ P_{r_{d_i}}^{(s)} &= C(d_i, s) {P_1'}^{s} (1 - P_1')^{d_i - s} \cdot C(d_i', s+1) {P_2'}^{s+1} (1 - P_2')^{d_i' - s - 1} \\ n &= \min(d_i, d_i' - 1) \end{aligned} \tag{25}$$

– probability of connection loss while trying to recover $P_{la_{d_i}}$

$$\begin{aligned} P_{la_{d_i}} &= \sum_{s=1}^{n} P_{la_{d_i}}^{(s)} \\ P_{la_{d_i}}^{(s)} &= C(d_i, s) {P_1'}^{s} (1 - P_1')^{d_i - s} \cdot C(d_i', s-1) {P_2'}^{s-1} (1 - P_2')^{d_i' - s + 1} \\ n &= \min(d_i, d_i' + 1) \end{aligned} \tag{26}$$

– probability of failure to recover a lost connection $P_{f_{d_i}}$

$$\begin{aligned} P_{f_{d_i}} &= \sum_{s=0}^{n} P_{f_{d_i}}^{(s)} \\ P_{f_{d_i}}^{(s)} &= C(d_i, s) {P_1'}^{s} (1 - P_1')^{d_i - s} \cdot C(d_i', s) {P_2'}^{s} (1 - P_2')^{d_i' - s} \\ n &= \min(d_i, d_i') \end{aligned} \tag{27}$$

Where

$$\begin{aligned} P_1' &= P_{in\_L} P_{C_{l-r}} \\ P_2' &= P_{C_{r-l}} \end{aligned} \tag{28}$$

4.3 Calculation of P_{a-c}, $P_{F_{l-r}}$ ($P_{C_{l-r}}$) and $P_{F_{r-l}}$ ($P_{C_{r-l}}$)

To complete the estimation of state transition probabilities we need to determine P_{a-c}, the probability of one robot moving from area $\mathbb{A}$ to area $\mathbb{C}$ (for Equation 15), and the probabilities $P_{F_{l-r}}$ ($P_{C_{l-r}}$) and $P_{F_{r-l}}$ ($P_{C_{r-l}}$) that one robot will move from area $\mathbb{L}$ to $\mathbb{R}$, and from area $\mathbb{R}$ to $\mathbb{L}$ respectively (for Equations (20)(28)).

Consider the possible motion between $robot_i$ and $robot_B$ in Figure 5. If we establish Cartesian coordinates with origin located at the centre of $robot_B$, as shown in Figure 6, the x-axis is the extension of a ray from the centre of $robot_i$ to the centre of $robot_B$ and the y-axis is across $robot_B$ from South to North. The relative speed of $robot_B$, with reference to $robot_i$, will vary from 0 to $2V$, and the relative heading will differ from 0° to 360°. Clearly the distance between the two robots will decrease if the relative heading varies from 90° to 270° (*moving towards*), otherwise the distance will increase (*moving away*). The motion *moving towards*, could possibly trigger $robot_i$'s avoidance sensor at step $k+1$, if the two robots are already close enough at step k, or possibly increase the number of neighbours for $robot_i$ at step $k + T_C$, if $robot_B$ is in area $\mathbb{R}$ at step k. Conversely, the motion *moving away* will possibly decrease the number of connected neighbours for $robot_i$ at step $k + T_C$.

The probabilities of *moving towards* and *moving away* should be different for the robots in different states and with different connectivities. For instance, a robot in the

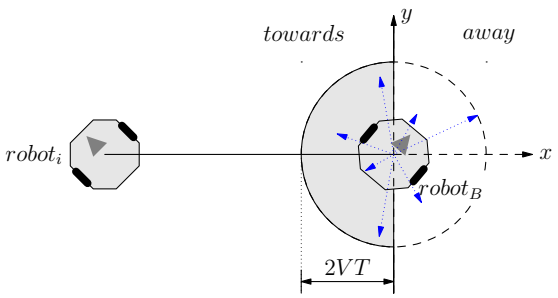

Fig. 6 The relative heading between two robots. The grey area represents the *moving towards* heading range.

coherence state will make a U-turn in order to recover the lost connection, thus its neighbours are more likely to be *moving towards* than *moving away*. However, in a crowded situation the robot has a large number of neighbours within communication range, and the robots are more likely to be *moving away* instead of *moving towards* because of the avoidance behaviours. In fact, it is the difference between the probabilities of *moving forwards* and *moving away* that drives the swarm toward aggregation. Since the robot updates its avoidance sensors much more frequently than its connectivity information (due to cadence) its neighbours, statistically, will have the same probabilities of *moving towards* as *moving away* in the short intervals between avoidance sensors updates. To estimate P_{a-c}, let $P_A(x)$ denote the probability of *moving towards* for the neighbours of $robot_i$ during the avoidance sensors update time interval, where x represents the number of neighbours of $robot_i$, then

$$P_A(x) = 0.5 \qquad (0 < x \leq d_{max}) \tag{29}$$

If $robot_i$ lost connections during the last T_C steps, it will enter state *coherence* and make a U-turn. Consider the situation when $robot_i$ is in the *coherence* state without any neighbours, i.e. in state C_0. There should be at least one robot in area $\mathbb{R}$ ($robot_D$ in Figure 5) since $robot_i$ lost one connection (that's why it is in state C_0 now). In this situation the distance between $robot_i$ and $robot_D$ is initially increasing before $robot_i$ makes the U-turn, but then after the U-turn $robot_D$ must be *moving towards* $robot_i$, i.e. with probability 100%. But then if the number of connected neighbours is increasing, the probability of *moving towards* will drop, and when $d_i = \alpha - 1$ we can suppose the probability will be closer to 50% due to the random-turn on connection recovery behaviour among its neighbours. For simplicity, let us assume the probabilities of *moving towards* for $robot_i$'s neighbours, denoted by $P_C(x)$, will decrease linearly between these two points, with the number of neighbours increasing, when $robot_i$ is in the *coherence* state,

$$P_C(x) = 1 - 0.5x/\alpha \qquad (0 \leq x < \alpha) \tag{30}$$

If $robot_i$ is in the *forward* state with $d_i < \alpha$ neighbours, then it has moved from the *coherence* state with $d_i - 1$ or d_i connections. There was a U-turn behaviour executed at step $k - T_C$ which must still have some effect on increasing the probabilities of *moving towards* for its neighbours. Meanwhile, the avoidance behaviour the neighbours themselves execute will also increase the probability of *moving away*. However, for the situation where $d_i \geq \alpha$, only the avoidance behaviour will increase the probability of

moving away. Assume the U-turn behaviour has the same effect on the probability of *moving towards* as the avoidance behaviour (but negative instead of positive), then the probability of *moving towards* for $robot_i$'s neighbours in this situation, say $P_F(x)$, is given by:

$$P_F(x) = \begin{cases} \dfrac{P_C(x) + 0.5 - P_{a_x}}{2} & 0 < x < \alpha \\ \dfrac{1 - P_{a_x}}{2} & \alpha \leq x \leq d_{max} \end{cases} \tag{31}$$

Where P_{a_x} can be obtained following Equation (15)

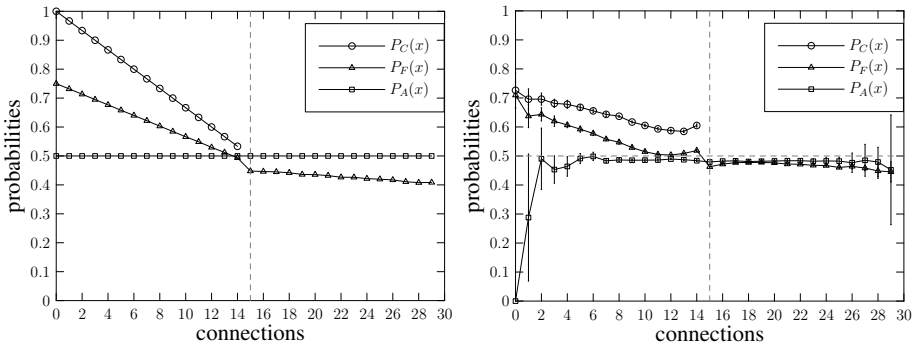

Fig. 7 Probabilities $P_A(x)$, $P_C(x)$ and $P_F(x)$ of *moving towards* in *avoidance*, *coherence* and *forward* states, for $\alpha = 15$. **Left:** probabilities obtained from Equations (29)-(31). **Right:** mean probabilities measured from simulation, error bars indicate the standard deviation of 10 runs, each run lasts 10000 seconds with a swarm of 40 robots.

Figure 7 (left) plots the probabilities of *moving towards* obtained from Equation (29)-(31), where $\alpha = 15$ and the geometrical parameters are given in Table 3. Figure 7 (right) then demonstrates the corresponding probabilities measured from simulation with the same parameters. It shows clearly that there is good agreement for $P_A(x)$, which is one of the most fundamental assumptions, and less satisfactory agreement for $P_C(x)$ and $P_F(x)$. The gaps between estimated and measured values are caused mainly by the fact that robots do need time to make a turning action, which is absent in the model. However, the relationship between $P_A(x)$, $P_C(x)$ and $P_F(x)$ is correct, i.e. $P_C(x) > P_F(x) > P_A(x)$ when $x \leq \alpha$ and $P_A(x) > P_F(x)$ when $x > \alpha$, which is the essential condition to gain the crossover between $P_{g_{d_i}}$ and $P_{l_{d_i}}$ (to be discussed in the next section). If we additionally assume that relative speeds are uniformly distributed between 0 and $2V$, we can estimate P_{a-c}, $P_{F_{l-r}}$ ($P_{C_{l-r}}$) and $P_{F_{r-l}}$ ($P_{C_{l-r}}$) using the Monte-Carlo method. Pseudo-code for this estimation is shown in appendix B.

5 Validation of transition probabilities

We use the sensor-based simulation toolset Player/Stage (Gerkey et al. 2003) to validate the probabilities estimation and the PFSM model. Figure 8 shows a screenshot of the simulation and the robot model used for simulation. The robots used in the simulation are models of the wirelessly networked Linuxbots in the Bristol Robotics Laboratory (Winfield and Holland 2000). 40 robots are simulated with $\alpha = 5$, $\alpha = 10$ and $\alpha = 15$.

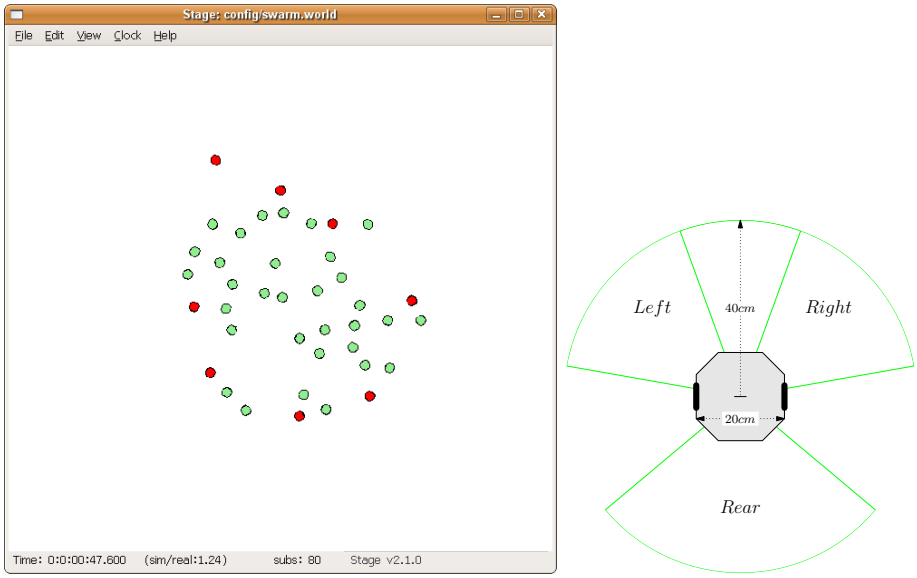

Fig. 8 Left: a screenshot of simulation, the darker robots are in the *coherence* state, the lighter ones are in the *forward* state. **Right:** the robot model used in simulation. A video presentation showing basic behaviours together with simulation runs for different values of α is provided on-line as supplementary material in file swarm-alpha-algorithm.wmv.

Each robot is sized 0.2 m $\times$ 0.2 m and equipped with three avoidance sensors, two at the front and one at the rear; each can detect obstacles within range $R_a = 0.4$ m and output a binary signal. Initially, robots are randomly dispersed within a 2 m circle area with random headings. Each robot will poll its avoidance sensor at frequency 5 Hz $(1/T)$, whenever one or more sensors are triggered the robot will execute an avoidance behaviour, i.e. turn away from the colliding robot or obstacle. The avoidance turn speed depends on which sensors are triggered and the robot will keep turning for 1 s $(T_A T)$. The communication range is 2 m for each robot and the robot will check its connectivity information every 3 s $(T_C T)$. For each α, simulations are run 10 times with the same initial conditions and each lasts for 10000 seconds; probabilities are measured by counting each type of state transition and the total number of cycles the robot remains in each state. Figure 9 (left) plots transition probabilities against the connectivity, error bars show the standard deviation of 10 simulation runs. Probabilities estimated using the geometrical approach in Section 4 are plotted in Figure 9 (right). The fixed parameters for both simulation and probabilities estimation are given in Table 3.

Figure 9 shows that measured and estimated transition probabilities are in good agreement, particularly for avoid probability P_a. There are some gaps between estimated and measured values for probabilities P_l, P_g, P_r, P_f and P_{la} but the relative positions of each probability curve match quite well. The very large error bars shown on the right-hand sides of Figure 9 (left), from simulation, reflect the fact that there are very few occurrences of the high connectivities, especially when $\alpha = 5$ or $\alpha = 10$. This is not a concern because the important range for the transition probabilities is observed from simulation to be from 0 to about 2α. Note that we clearly see a crossover

Table 3 Robot parameters for model validation. Note that the time step duration is given by the avoidance sensor update rate 5 Hz.

parameter	value	description
T	0.2 s	Time step duration
T_C	15	Coherence duration in time steps
T_A	5	Avoidance duration in time steps
V	0.15 m/s	Robot forward velocity
R_a	0.4 m	Avoidance sensor range
R_p	0.1 m	Robot body radius
R_w	2.0 m	Wireless range

between the curves of probabilities P_l and P_g at connectivities close to $\alpha - 1$ in both measured and estimated curves in Figure 9. This is particularly important since if the probability of connection loss P_l were higher than the probability of connection gain P_g over the whole connectivity range, the swarm would simply disperse. Conversely, if P_g were greater than P_l over the whole connection range the swarm would clump tightly together. The action of the *coherence* behaviour for robots with connectivities less than α creates the crossover in P_l and P_g and allows the swarm to maintain coherence and achieve a dynamic equilibrium between expansion and contraction.

Given the simplifying assumptions that we have made in the geometrical estimation of transition probabilities it is perhaps surprising that we achieve the reasonable match between measured and estimated probabilities over the connectivity ranges of interest we see in Figure 9. This gives confidence in the validity of the approach.

6 Running the macroscopic model

We can now run the macroscopic model by plugging both measured and estimated probability values into the difference equations developed in Section 3.1.

6.1 Stability and convergence of the model

To test the stability and convergence of our macroscopic model, we run the model with different initial conditions, as shown in Table 4, for $\alpha = 15$.

Table 4 Initial condition sets for macroscopic model, $F_0(0) = 40$ means there are 40 robots in state *forward*, each with 0 neighbours at time step 0.

case	initial condition
1	$F_0(0) = 40$
2	$F_{15}(0) = 40$
3	$F_{22}(0) = 40$
4	$F_{29}(0) = 20, C_{14}(0) = 20$

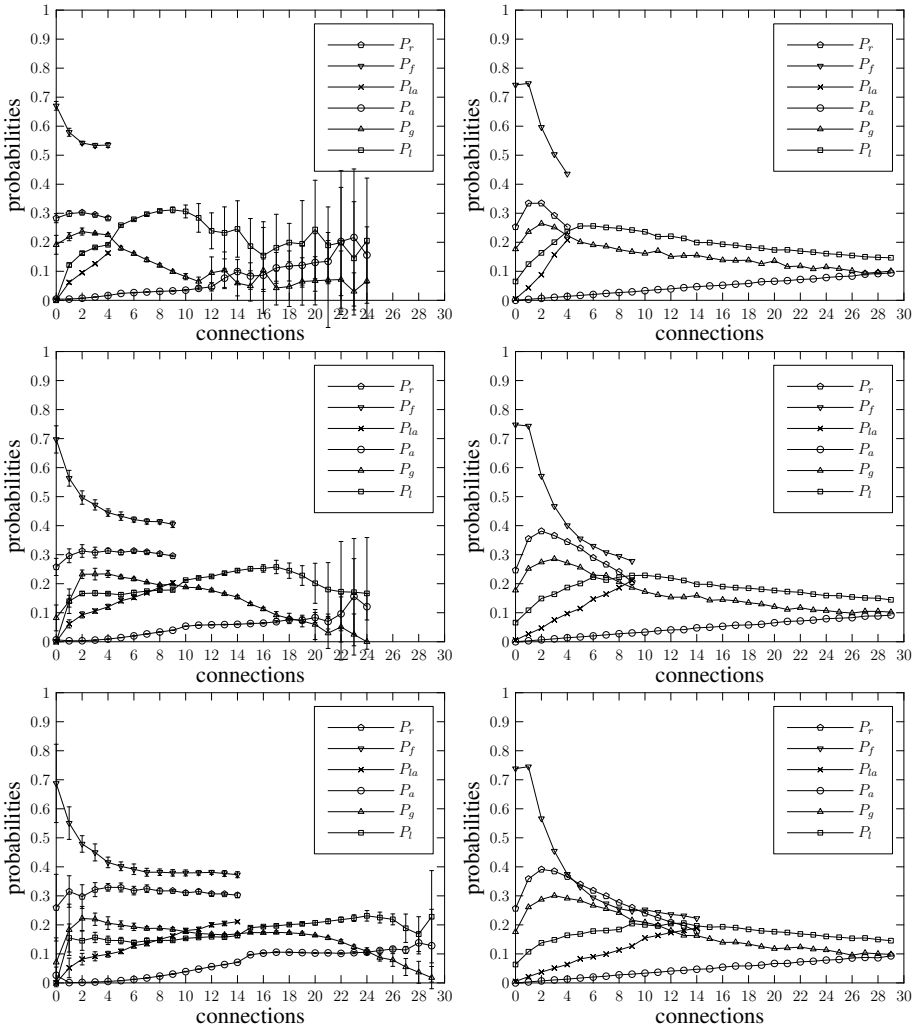

Fig. 9 State transition probabilities plotted against connectivity. From top to bottom, $\alpha = 5, 10$ and 15 respectively. **Left:** measured probabilities from the Player/Stage simulation (with a swarm of 40 robots). **Right:** estimated probabilities using the geometry-based approach.

The relative iteration error is defined as follows, as a measure of the speed of convergence of the numerical solution to the DEs in Section 3.1.

$$e = \left[\sum_{i=1}^{\alpha-1} \left(N_{A_i^C}(k) - N_{A_i^C}(k-1) \right)^2 + \sum_{i=1}^{m} \left(N_{A_i^F}(k) - N_{A_i^F}(k-1) \right)^2 + \sum_{i=0}^{m} \left(N_{F_i}(k) - N_{F_i}(k-1) \right)^2 + \sum_{i=0}^{\alpha-1} \left(N_{C_i}(k) - N_{C_i}(k-1) \right)^2 \right]^{1/2} \tag{32}$$

Figure 10 plots the relative iteration error against iteration steps for the model with different initial conditions. Note that for clarity, Figure 10 only shows e for time

steps from 500 to 10000. The model clearly converges under different initial conditions and the relative iteration error falls to less than $1e-6$ after 10000 iterations each time; *case 2* shows the fastest convergence. Moreover, regardless of the different initial conditions and speeds of convergence, all cases eventually reach the same point suggesting the existence of stable steady states. The same occurs with $\alpha = 5$ and 10. Thus our PFSM model is robust to different initial conditions and can reach a solution for the connectivity structure of the swarm.

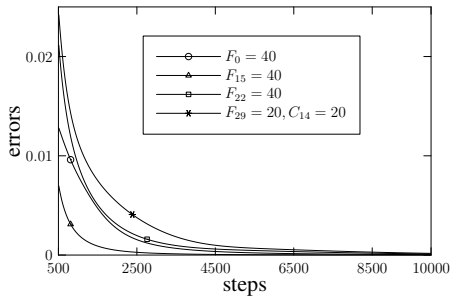

Fig. 10 Iteration error for different initial conditions, obtained using estimated transition probabilities.

6.2 Validating the macroscopic model structure

In order to validate the macroscopic model structure in isolation from the geometrically estimated transition probabilities we now compare connectivity results from simulation with connectivity results predicted by the model, using the *measured* transition probabilities from simulation in the model. By using measured probabilities we focus only on the macroscopic model from Section 3.

Figure 11 shows the average number of robots in states *forward*, *coherence* and *avoidance* after reaching the steady state, in which we merge states A^C and A^F from the sub-PFSMs in Figure 4, plotted against connectivity. The left-hand plots show the results collected from the Player/Stage simulation while the right-hand side plots show the results from the macroscopic model run with the measured probabilities, also from the Player/Stage simulation. The total average number of robots, summing all states for each connectivity value, is also plotted in Figure 11 as the topmost curve; the area below this curve is constant and represents all 40 robots involved in the experiment. The curves on both sides, except perhaps for state *coherence*, show reasonable agreement in both values and curve shape. For the total number of robots, we see that the average number of robots describes a bell shaped curve reaching a peak just below the threshold value α, as we would expect.

Since this comparison is made primarily to provide validation of the structure of the macroscopic model we will defer detailed discussion until we see results from the full model using probabilities estimated from geometrical considerations. However, we should note that the two sets of curves do not match precisely. Results for $\alpha = 5$ show the strongest correlation, but $\alpha = 10$ and $\alpha = 15$ less so. In particular, our macroscopic model appears to underestimate the number of robots in the *coherence* state with a

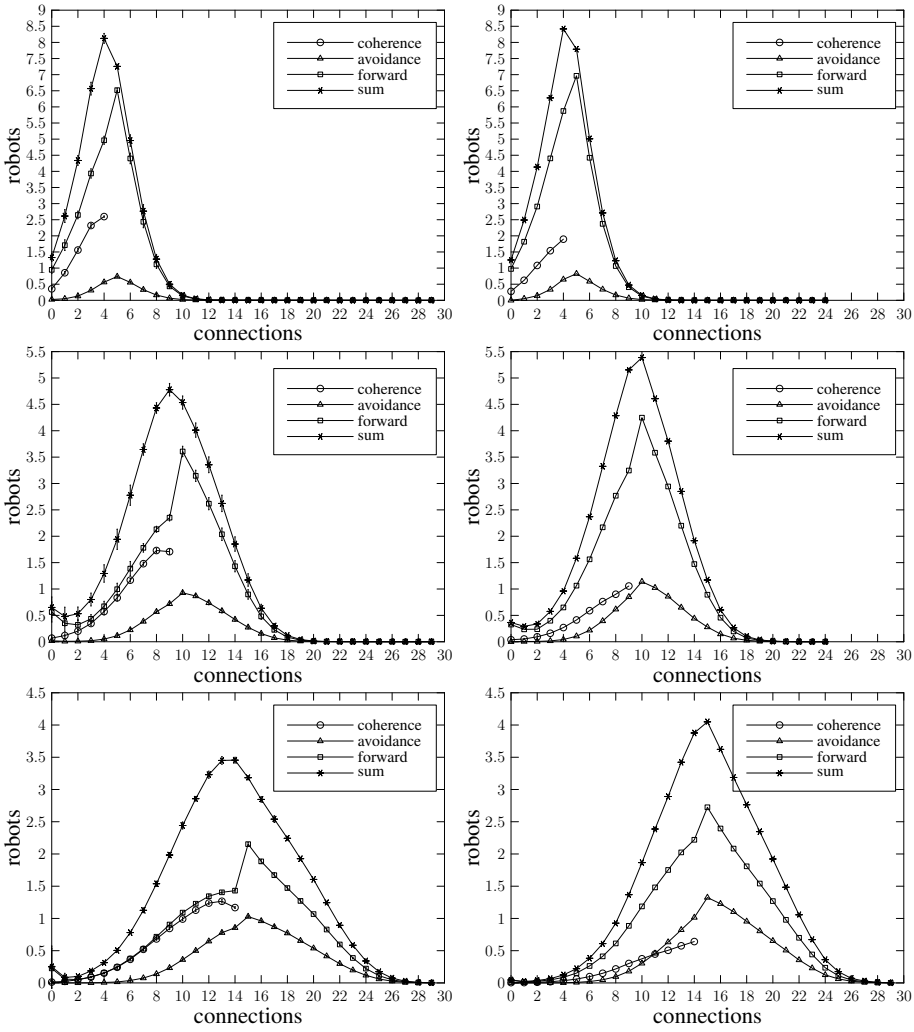

Fig. 11 Number of robots in state *coherence*, *forward* and *avoidance* plotted against number of neighbours (connectivity). From top to bottom, $\alpha = 5,\ 10$ and 15. **Left:** simulation average of 10 runs, each simulation lasts for 10000 seconds. **Right:** macroscopic model using measured probabilities from simulation.

balancing overestimate of the number of robots in the *forward* state. The model does, with reasonable accuracy, predict the number of robots in the *avoidance* state. Since we are using measured transition probabilities then any differences must be due to structural inaccuracies in the macroscopic model of Figure 3. The underestimate in the number of robots in the *coherence* state, particularly at connectivity values approaching α, is almost certainly due to the simplifying assumption, in the PFSM, that robots lose only a single connection at a time. In practice a robot could, for example, lose 2 connections within time T_C and transition directly from state $\overline{F}_\alpha$ to state $\overline{C}_{\alpha-2}$. Such simplifying assumptions are necessary in a model, and we argue that the results of Figure 11 provide acceptable validation of the structure of the macroscopic model.

6.3 Running the macroscopic model with geometrically estimated probabilities

We now run the macroscopic model with state transition probabilities estimated using the geometrical approach in Section 4. Figure 12 shows the average number of robots in states *forward*, *coherence* and *avoidance*, in which we again merge states A^C and A^F from the sub-PFSMs in Figure 4, plotted against connectivity. The left-hand plots show the same results collected from the Player/Stage simulation as shown in Figure 11 (left), while the right-hand side plots now show the results from the PFSM model run with the estimated state transition probabilities. The total average number of robots, summing all states, is also plotted as the topmost curve in each graph.

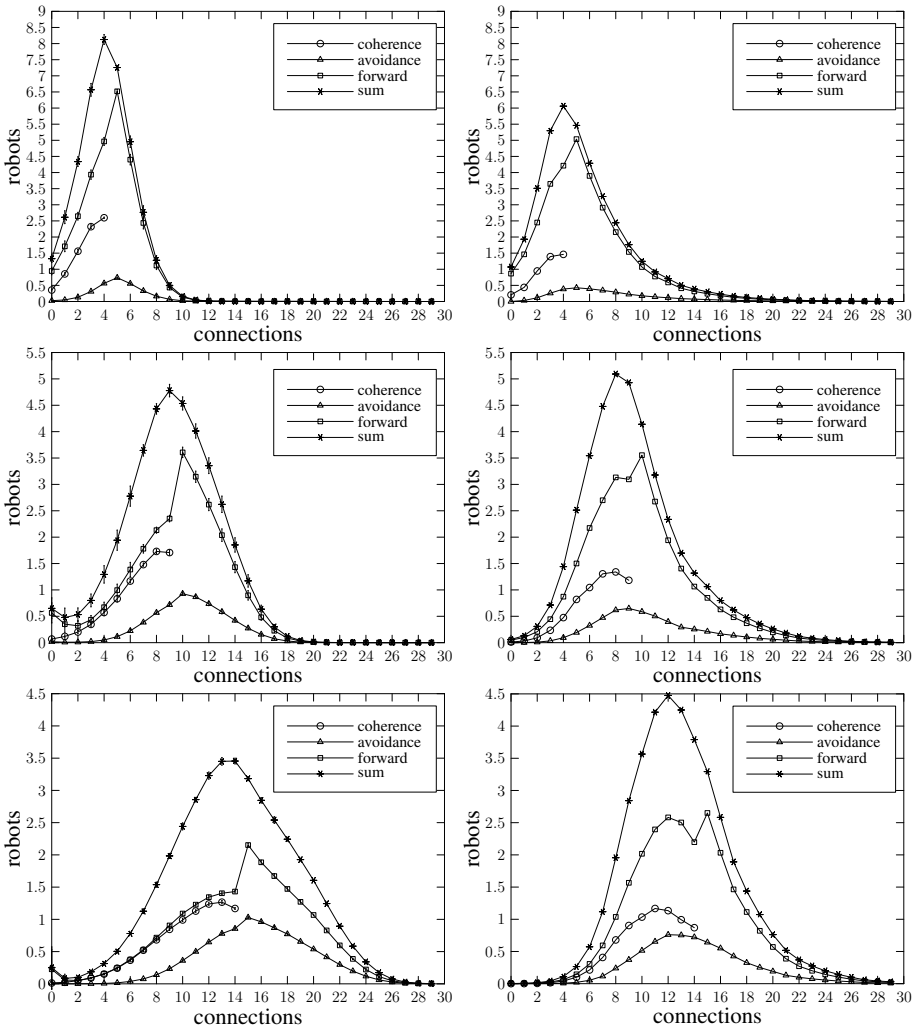

Fig. 12 Number of robots in state *coherence*, *forward* and *avoidance* plotted against number of neighbours (connectivity). From top to bottom, $\alpha = 5$, 10 and 15. **Left:** simulation average of 10 runs, each simulation lasts for 10000 seconds. **Right:** macroscopic model using geometrically estimated probabilities.

First we note that the PFSM model generates the same "bell" shaped curves as the simulation, and for all three values of α the peak occurs at or very close to the same connectivity value. The PFSM model for $\alpha = 5$ somewhat underestimates the number of robots in all three states and also shows a longer "tail" of robots with high connectivity values than is measured from simulation; however, the model shows reasonable agreement at very low connectivity values, especially in predicting "lost" robots (with connectivity of zero). At $\alpha = 10$ the macroscopic model again shows a longer tail of high connectivity robots than the simulation; also evident is the same overestimate in the number of robots in the *forward* state at connectivity values below α observed in Figure 11. The overestimate in *forward* robots is even more pronounced at $\alpha = 15$. We also see that the "lost" and very low connectivity robots are not seen in the model for $\alpha = 10$ and $\alpha = 15$. In all three pairs of results the greatest discrepancy between the macroscopic model and the simulation is in robots in the *forward* state. In contrast, the macroscopic model shows much stronger agreement with simulation for the number of robots in *coherence* and *avoidance* states.

Accounting precisely for the differences between the macroscopic model and simulation is clearly difficult, given that the model incorporates a number of simplifying assumptions, both in the PFSM structure (as discussed in the previous section), and in the development of the geometrical estimation of state transition probabilities. Furthermore, it is possible that some assumptions have counterbalancing effects. Let us now review the main simplifying assumptions, and their possible impact on the accuracy of the macroscopic model:

1. The structure of the PFSM in Figure 3 assumes that connections are lost or gained strictly one at a time whereas, in practice, in the T_C steps between connectivity updates a robot could lose or gain more than one connection. This is likely to be particularly true at higher connectivity values, and almost certainly accounts for the underestimate in the number of robots in the *coherence* state, and a balancing overestimate in robots in the *forward* state in Figure 11.
2. In estimating transition probabilities we have assumed that robots are distributed uniformly. Without such an assumption we could not estimate the likelihood of robots falling within the key regions in which robot collision, connection loss, or connection recovery are possible (labelled $\mathbb{A}$, $\mathbb{L}$ and $\mathbb{R}$ in Figure 5).
3. The third key assumption is that made in estimating probabilities of *moving towards* and *moving away* in section 4.3, in particular the assumption of linear functions for $P_A(x)$, $P_C(x)$ and $P_F(x)$ and the underlying assumption that robots turn in zero time.

Clearly, the second and third assumptions above must, together, account for the inaccuracies seen in Figure 12 (right) that are not present in Figure 11 (right), in particular the larger "tail" of robots with very high connectivities in cases $\alpha = 5$ and $\alpha = 10$, and at very low connectivities $0, 1, 2$ in cases $\alpha = 10$ and $\alpha = 15$. It is difficult to disaggregate the effect of each assumption in explaining these model errors. However, the assumption of uniformly distributed robots is clearly weakest at the edges of the swarm which is typically where robots with the lowest connectivity are to be found. Assumption 2 is therefore most likely to account for model errors at very low connectivities. Note also that assumptions 2 and/or 3 do appear to counterbalance the effect of assumption 1, noting the apparently improved accuracy in predicting the number of robots in the *coherence* state from Figure 11 (right) to Figure 12 (right). Given the simplifying assumptions in the model developed in this paper it is perhaps

surprising that the macroscopic model does generate such plausible results for the swarm connectivity structure.

7 Conclusions

This paper has proposed a macroscopic probabilistic model for an *ad-hoc* wirelessly connected swarm of mobile robots. We have described the development of the model in two stages: first, the probabilistic finite state machine (PFSM) and its description as a set of difference equations and, second, a novel robot-centric geometrical approach to the estimation of the state transition probabilities of the PFSM. We have, as one would expect in any model, made a number of simplifying assumptions. However, the full macroscopic model is developed entirely from first principles and requires neither measured or heuristic probability data from real or simulated robot swarms, nor *a posteriori* fitting. The paper has carefully validated each stage of the model against measured data from simulation. Finally, the full macroscopic model has been shown to predict the steady-state connectivity structure of the robot swarm to a reasonable degree of approximation.

We can draw a number of general conclusions, as follows:

- We have demonstrated that the probabilistic modelling approach can usefully be extended to the class of robotic swarms operating in unbounded space; in particular the robot-centric approach we have developed for estimation of transition probabilities requires us to make no limiting assumptions at all about the environment in which the swarm operates. This approach could, we believe, have merit in modelling other swarm robotic systems operating in unbounded 2D or 3D space and thus help to support the validation that would be a necessary part of the real-world application of such systems.
- The actual swarm we have modelled in this paper makes use of limited-range wireless network connectivity in order to maintain swarm aggregation and thus, as an emergent property, forms an *ad-hoc* network of mobile agents. We would argue therefore, that the modelling approach developed in this paper could find application in the analysis of mobile ad-hoc networks (MANETs). As far as we are aware this paper is the first to present a state-machine approach to the analysis of a MANET.

Specific conclusions about the macroscopic model for the wireless connected swarm are as follows:

- Notwithstanding the model errors resulting from the simplifying assumptions, the macroscopic model does have utility. In particular, the model has value in allowing us to rapidly assess the qualitative and quantitative effect on the swarm connectivity structure of varying the key parameters of the swarm. This paper has demonstrated that the model does, with reasonable accuracy, predict the effect of varying the α threshold value. Although not studied here, the model also readily allows us to study the effect of varying other parameters such as R_a, R_w or T_C. We can model swarms of arbitrary size characterised by the same numerical computation time, in contrast with simulation which quickly becomes impractical because of processing time for swarms much larger than 40, as used in this paper.

– The development of the macroscopic model for the wireless connected swarm has beneficially deepened our understanding of the α-algorithm and provided new insights which could, for instance, result in improved variants of the algorithm. The model has, for instance, provided valuable new qualitative insights into the α-algorithm, in particular the crossover of probabilities P_l and P_g necessary to achieve the spatial dynamic equilibrium of the steady-state swarm (Section 5).

Further work This paper has presented a work in progress. We have proposed a macroscopic probabilistic model for a wirelessly connected swarm of mobile robots and provided initial confirmation of the approach by validating the model against a sensor-based simulation. However, the model needs further development in order for its full potential to be realised. Thus further work will seek to:

– Undertake a steady-state analysis of the DEs in Section 3.1 in order to further validate the model and provide further insights into the α-algorithm.
– Incorporate a more realistic wireless propagation model into the Player/Stage simulation, in order to test the model (and the algorithm) for potential application in real-world scenarios.
– Extend the model to incorporate additional swarm behaviours, including collective swarm movement towards a beacon (i.e. swarm photo-taxis, Nembrini (2005)), and hence develop the model to the point where it can be used to gain quantitative insights including parameter optimisation.

Acknowledgements We are indebted to both the anonymous reviewers of this manuscript for their detailed and insightful comments, and to Editor-in-Chief Marco Dorigo for his additional detailed comments. Wenguo Liu has been supported by the EU Asia-link Selection Training and Assessment of Future Faculty (STAFF) project. Julien Nembrini was sponsored during this work by a grant from the Swiss National Science Foundation under grant number PP002-116913.

References

Agassounon, W., Martinoli, A., and Easton, K. (2004). Macroscopic modeling of aggregation experiments using embodied agents in teams of constant and time-varying sizes. *Autonomous Robots*, 17(2-3):163–192.

Agassounon, W., Martinoli, A., and Goodman, R. M. (2001). A scalable, distributed algorithm for allocating workers in embedded systems. In *Proceedings of the IEEE Conference on Systems, Man. and Cybernetics*, pages 3367–3373. IEEE Press, Piscataway, NJ.

Beni, G. (2005). From swarm intelligence to swarm robotics. In Şahin, E. and Spears, W. M., editors, *Swarm Robotics - SAB 2004 International Workshop*, volume 3342 of *LNCS*, pages 1–9. Springer-Verlag, Berlin, Germany.

Berman, S., Halasz, A., Kumar, V., and Pratt, S. (2007). Algorithms for the analysis and synthesis of a bio-inspired swarm robotic system. In Şahin, E., Spears, W. M., and Winfield, A. F. T., editors, *Swarm Robotics - second SAB 2006 International Workshop*, volume 4433 of *LNCS*, pages 56–70. Springer-Verlag, Berlin, Germany.

Correll, N. and Martinoli, A. (2004). Modeling and optimisation of a swarm-intelligent inspection system. In Alami, R., Asama, H., and Chatila, R., editors, *Proceedings of the 7th Symposium on Distributed Autonomous Robotic Systems (DARS'04)*, pages 369–378. Springer-Verlag, Berlin, Germany.

Gerkey, B., Vaughan, R., and Howard, A. (2003). The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th International Conf. on Advanced Robotics*, pages 317–323. IEEE Press, Piscataway, NJ.

Ijspeert, A., Martinoli, A., Billard, A., and Gambardella, L. (2001). Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment. *Autonomous Robots*, 11(2):149–171.

Kazadi, S., Chung, M., Lee, B., and Cho, R. (2004). On the dynamics of puck clustering systems. *Robotics and Autonomous Systems*, 46(1):1–27.

Lerman, K. and Galstyan, A. (2002). Mathematical model of foraging in a group of robots: Effect of interference. *Autonomous Robots*, 13(2):127–141.

Lerman, K., Galstyan, A., Martinoli, A., and Ijspeert, A. (2002). A macroscopic analytical model of collaboration in distributed robotic systems. *Artificial Life*, 7:375–393.

Lerman, K., Martinoli, A., and Galstyan, A. (2005). A review of probabilistic macroscopic models for swarm robotic systems. In Şahin, E. and Spears, W. M., editors, *Swarm Robotics - SAB 2004 International Workshop*, volume 3342 of *LNCS*, pages 143–152. Springer-Verlag, Berlin, Germany.

Martinoli, A., Easton, K., and Agassounon, W. (2004). Modeling swarm robotic systems: A case study in collaborative distributed manipulation. *International Journal of Robotics Research*, 23(4):415–436.

Martinoli, A., Ijspeert, A. J., and Mondada, F. (1999). Understanding collective aggregation mechanisms: From probabilistic modelling to experiments with real robots. *Robotics and Autonomous Systems*, 29(1):51–63.

Milutinovic, D. and Lima, P. (2006). Modeling and optimal centralized control of a large-size robotic population. *IEEE Transactions on Robotics*, 22(6):1280–1285.

Nembrini, J. (2005). *Minimalist Coherent Swarming of Wireless Networked Autonomous Mobile Robots*. PhD Thesis, University of the West of England, Bristol, UK.

Nembrini, J., Winfield, A. F. T., and Melhuish, C. (2002). Minimalist coherent swarming of wireless networked autonomous mobile robots. In *From Animals to Animats (SAB'02)*, pages 373–382. MIT Press, Cambridge, MA.

Rouff, C., Truszkowski, W., Rash, J., and Hinchey, M. (2003). Formal approaches to intelligent swarms. In *IEEE/NASA Software Engineering Workshop (SEW'03)*, pages 51–57. IEEE Computer Society, Los Alamitos, CA.

Şahin, E. (2005). Swarm robotics: From sources of inspiration to domains of application. In Şahin, E. and Spears, W. M., editors, *Swarm Robotics - SAB 2004 International Workshop*, volume 3342 of *LNCS*, pages 10–20. Springer-Verlag, Berlin, Germany.

Støy, K. (2001). Using situated communication in distributed autonomous mobile robotics. In *Proceedings of the 7th Scandinavian conf. on Artificial Intelligence*, pages 44–52. IOS Press, Amsterdam, The Netherlands.

Truszkowski, W., Hinchey, M., Rash, J., and Rouff, C. (2004). NASA's swarm missions: The challenge of building autonomous software. *IT Professional*, 6(5):47–52.

Winfield, A. F. T., Harper, C., and Nembrini, J. (2006). Towards the application of swarm intelligence in safety-critical applications. In *Proceedings of the 1st IET International Conf. on System Safety*, pages 89–95. IET Press, London, UK.

Winfield, A. F. T. and Holland, O. (2000). The application of wireless local area network technology to the control of mobile robots. *Microprocessors and Microsystems*, 23:597–607.

Winfield, A. F. T. and Nembrini, J. (2006). Safety in numbers: fault-tolerance in robot swarms. *International Journal of Modelling, Identification and Control*, 1(1):30–37.

Appendix A: Estimation of maximum number of neighbours d_{max}

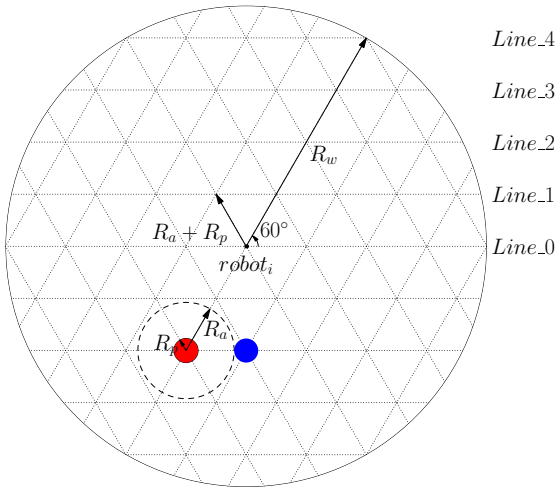

Fig. 13 $robot_i$ and its maximal neighbours

The maximum number of neighbours for the robot d_{max} is constrained by the physical size of robot R_p, the radius of the avoidance sensor R_a and the radius of the communication range R_w. Consider the extreme case where the avoidance sensors of each robot are ready to be triggered, then the distribution of robots within radius R_w could be shown as Figure 13. The intersections between lines represents the possible position of robots. It is obvious that we can calculate d_{max} by counting the number of intersections shown in Figure 13. In order to count the intersections, we name the lines $Line_0$, $Line_1$ ... $Line_{k_{max}}$ as shown in Figure 13. Where

$$k_{max} < \left\lfloor \frac{2R_w}{\sqrt{3}(R_a + R_p)} \right\rfloor \tag{33}$$

For $Line_k$, $k = (0, 1, \ldots, k_{max})$, the number of intersections is given by:

$$N_k = \left\lfloor \frac{\sqrt{4R_w^2 - 3(R_a + Rp)^2 k^2}}{R_a + R_p} \right\rfloor + 1 \tag{34}$$

Therefore, the maximum number of neighbours for $robot_i$ is:

$$d_{max} = N_0 + 2 \sum_{k=1}^{k_{max}} N_k - 1 \tag{35}$$

However, the maximum number of neighbours should be smaller due to the dynamic nature of the swarm thus, in practice, we can choose $m = 0.8 \sim 0.9\ d_{max}$.

Appendix B: Estimation of probabilities P_{a-c}, $P_{F_{l-r}}$ ($P_{C_{l-r}}$) and $P_{F_{r-l}}$ ($P_{C_{r-l}}$)

Note that the listing below only shows the process of calculating P_{a-c}. To calculate $P_{F_{l-r}}$ ($P_{C_{l-r}}$) and $P_{F_{r-l}}$ ($P_{C_{r-l}}$), we need to change the area $\mathbb{A}$ in line 7 to $\mathbb{L}$ or $\mathbb{R}$, and the area $\mathbb{C}$ in line 15, to $\mathbb{R}$ or $\mathbb{L}$, respectively.

```
 1      Initialise the parameters
 2      count = 0, hit = 0
 3
 4      while ( count < MAXLOOP ){
 5        count ++
 6
 7        Generate a random point P within area $A$
 8        dir = random heading obeying the assumption of
 9              probability moving-towards
10        vel = random velocity between (0,2V)
11
12        new_P.x = P.x + vel * step * sin(dir)
13        new_P.y = P.y + vel * step * cos(dir)
14
15        if new_P located within area $C$
16           hit ++
17      }
18
19      return hit / count;
```