robotica 2009
festival nacional de robótica

Supervised Group Size Regulation in a Heterogeneous Robotic Swarm

Rehan O'Grady, Carlo Pinciroli, Anders Lyhne Christensen and Marco Dorigo

***Abstract*— In this study, we work with a heterogeneous swarm of wheeled and aerial robots. We present a self-organised approach inspired by the aggregation behaviour of cockroaches that allows each aerial robot to form a dedicated group of wheeled robots of a particular size. Our approach is based on simple probabilistic rules, but still proves robust and flexible. Different groups can be formed in parallel, and the size of the groups can be dynamically modified. Our work is based on a real robotic platform that is still under development—here, we present results and analysis of extensive simulation-based experiments. We also present a mathematical analysis of our system.**

I. Introduction

We consider a scenario in which a range of different tasks must be carried out by a heterogeneous swarm made up of flying robots (eye-bots) and wheeled robots (foot-bots). Each task is physically executed by a dedicated group of foot-bots. The eye-bots are responsible for exploring the environment, determining the optimal foot-bot group size for each task, aggregating foot-bots into groups of the relevant sizes, and finally for guiding the groups of foot-bots to the task sites.

In this study, we focus on the aggregation and group size regulation aspect of the above scenario. Aggregation is a fundamental process that has been studied in many different contexts such as biology [8], physics [16] and robotics [5]. The aggregation behaviour of cockroaches in an environment containing shelters is relatively simple and has been well explored by biologists—its dynamics can be accurately modelled at the individual cockroach level by mapping local environmental conditions to simple stop/go probabilities. Previous robotics studies have shown how this behaviour can be faithfully mimicked by a group of robots. Existing robotic implementations share key features of the cockroach model. In particular, the equilibrium distribution of agents depends passively on the initial configuration of the environment and on the static mapping of environmental conditions to stop/go probabilities.

In this paper, we take inspiration from the cockroach aggregation model, but extend it so that the equilibrium distribution of cockroach like agents (foot-bots in our case) can be dynamically controlled by the system. To do this, we treat our eye-bots like 'active' shelters under which the foot-bots aggregate. The eye-bots are active in the sense that they can dynamically alter the stop/go probabilities for the foot-bots aggregating underneath them.

We model our system mathematically, and then implement it using simulated eye-bots and foot-bots (the robots we simulate are based on a real robotic platform that is still under development). We demonstrate the feasibility of our system and show that it exhibits several desirable features, namely *stabilisation*, *redistribution* and *balancing* (these concepts are elaborated in Section III).

Carlo Pinciroli, Rehan O'Grady and Marco Dorigo are with Université Libre de Bruxelles, Brussels, Belgium ({cpinciro,rogrady,mdorigo}@ulb.ac.be). Anders Lyhne Christensen is with DCTI at Lisbon University Institute, Portugal (anders.christensen@iscte.pt).

II. Related Work

There is a large body of literature on heterogeneous robotic groups [7], [15]. However, to the best of our knowledge, no existing robotic study investigates group size regulation in heterogeneous robot groups. There is some literature on aggregation and group size regulation in homogeneous groups. Dorigo *et al.* [1] evolved two dimensional distributed aggregation in a swarm of embodied robots, Martinoli *et al.* [5] investigated the effects of probabilistic parameters on the size of object clusters collected by Khepera robots. Neither work provided an explicit group size control mechanism. Melhuish *et al.* [6] controlled group sizes in a swarm of abstracted agents using a firefly-like synchronisation mechanism. However, group size control was not fine grained to the level of individual robots, only one group was formed at a time and the physics of an embodied system was not taken into account.

In a series of experiments in a white circular arena, Jeanson *et al.* [3] derived a probabilistic behavioural model of the first instar larvae of the German cockroach *Blattella germanica*. They showed that individuals switch probabilistically between two alternative behaviours: random walk and resting. Analysis revealed that the probability for a cockroach to switch to (or remain in) one of these two states depends on the number of resting cockroaches in its neighbourhood (in direct antenna contact): as this number increases, the stopping probability also increases, while the probability of leaving an aggregate decreases. Furthermore, experimental evidence shows that cockroaches prefer to aggregate in dark places [13]. If multiple shelters are present in the environment, the majority of cockroaches aggregate under only one shelter rather than spreading evenly among the different shelters. A positive feedback mechanism ensures that this happens even when the shelters are perfectly identical [4].

Garnier *et al.* [2] used Jeanson's behavioural model to show that a group of cockroach-like robots can achieve a collective choice between two different shelters in the environment through simple local interactions.

III. Goals

Our system must be able to cope with the dynamic arrival and departure of eye-bots and foot-bots during system execution. It is also reasonable to assume that there will sometimes be insufficient foot-bots available to carry out all tasks at the same time, i.e., there may not be enough foot-bots to fill the quota of every eye-bot (*quota* refers to the desired group size of a single eye-bot). We have identified three key features that we believe

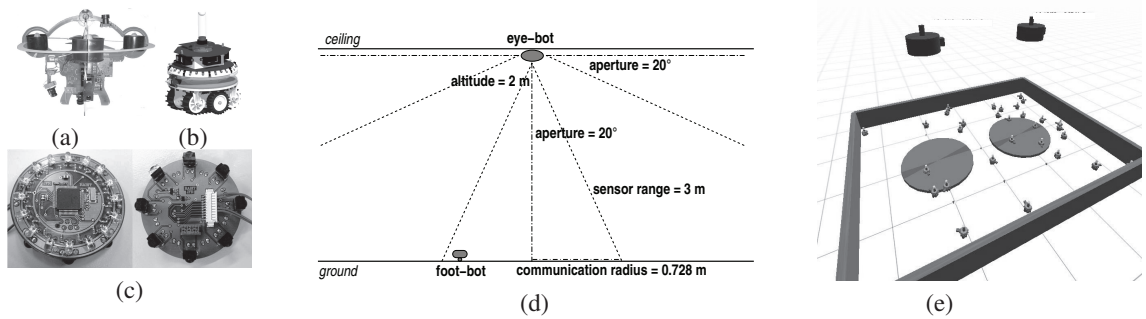


Fig. 1. The robots used in this study. At the time of writing, the robotic hardware is still under development. (a) The eye-bot (aerial robot). (b) The foot-bot (wheeled robot). (c) Hardware prototype of the range and bearing system. (d) The communication range of the eye-bot. (e) The simulation environment. The grey circles represent the signal range of the eye-bot's vertical range and bearing communication system—i.e., the area in which each eye-bot can aggregate foot-bots.

are necessary for a dynamic group size regulating system, namely *stabilisation*, *redistribution* and *balancing*.

Stabilisation means that the system settles down to an equilibrium state that is sufficiently stable to allow each eye-bot to independently determine that equilibrium has been reached (and therefore that it is reasonable to stop aggregating and initiate a subsequent behaviour). This is not trivial when there are insufficient foot-bots to meet the quotas of every eye-bot.

Redistribution means that a running system can respond to the dynamic addition or removal of robots of either type, and efficiently redistribute foot-bots accordingly. We would expect the natural dynamics of the system to encourage redistribution, rather than having to use communication to explicitly coordinate redistribution.

Balancing means that when there are insufficient foot-bots, the system stabilises in a state where each eye-bot has filled the same percentage of its quota of foot-bots (independently of the quota size). This is important as it ensures that no eye-bots arbitrarily take precedence over other eye-bots. Consider a future implementation of this system in which tasks have different priorities. In the case where there are not enough foot-bots, the balancing property will ensure that quotas with low priority are not filled at the expense of quotas with high priority[1].

IV. Robotic Platform and Simulation Environment

We consider a heterogeneous robotic system composed of two types of robots: aerial robots (eye-bots) and wheeled robots (foot-bots). Eye-bots are quad-rotor equipped robots capable of flying and attaching to the ceiling. For the purpose of this study, we assume that the eye-bots are always attached to the ceiling. Eye-bots are equipped with a high resolution camera which allows them to monitor what happens on the ground [12], see Figure 1(a). Foot-bots, on the other hand, move around on the ground. They are equipped with infrared proximity sensors, an omnidirectional camera, and an RGB LED ring that enables them to display their state to robots within visual range, see Figure 1(b).

The eye-bots communicate with the foot-bots using a range and bearing system [11] mounted on both robots see Figures 1(c) and 1(d). This system allows the eye-bots to locally broadcast and receive messages either from other eye-bots in the same plane, or to foot-bots in a cone beneath them. Furthermore, the system allows for *situated communication*. This means that recipients of a message know both the content and the physical origin of the message within their own frame of reference.

At the time of writing, the robotic hardware is still under development. For this reason, the results presented in this paper have been obtained in simulation. A custom physics based simulator called ARGoS [9] has been developed to reproduce the dynamics of the robots' sensors and actuators with reasonable accuracy. A screen shot of simulation environment is shown in Figure 1(e).

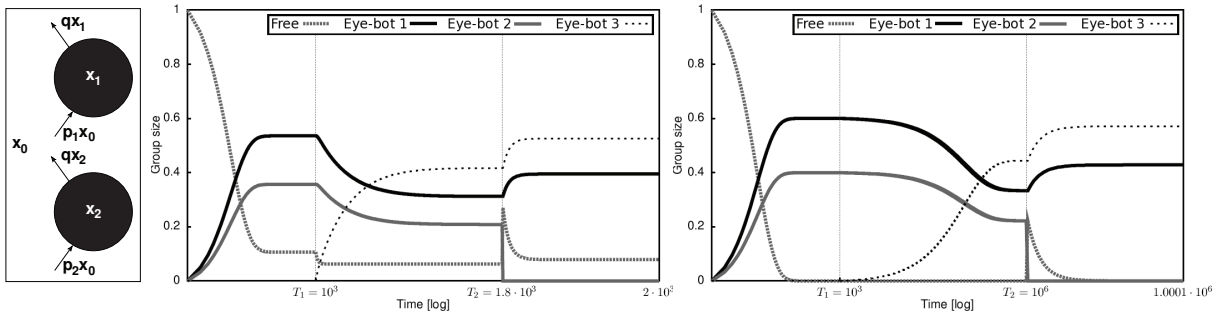
V. Methodology

In our biologically inspired system, we let foot-bots play the role of cockroaches, while eye-bots play the role of shelters. Unlike the static shelters in Garnier's previous system [2], the eye-bots in our system actively broadcast varying stop and go probabilities. By changing the stop and go probabilities that it broadcasts, an eye-bot can actively influence the number of foot-bots in its group. Like cockroaches, foot-bots can be moving (state FREE) or stationary (state IN_GROUP). A foot-bot is considered part of an eye-bot's aggregate if it is underneath it (i.e., within range of that eye-bot's range and bearing communication signal—see Figure 1(d)) and it is in state IN_GROUP.

Foot-bots in state FREE perform a random walk in the arena. At each time step, each eye-bot i sends a message containing two pieces of information (that will be received by any foot-bots underneath it): the probability for a foot-bot in state FREE to join the group (j_i) and the probability for a robot in state IN_GROUP to leave it (q).

Throughout this paper, we use a single experimental setup to test the various configurations of our system. We use this experimental setup to test both our abstract mathematical models and our concrete implementations on the robotic platform. In particular, we designed the

[1]In the future, an additional deadlock resolution mechanism will be needed to allow the system to take meaningful action when the system stabilises in a state where none of the eye-bot quotas have been filled. In the priority based system, for example, after stabilisation the eye-bots could each have a probability of releasing some of their aggregated foot-bots. These release probabilities could be linked to the priority of each eye-bot's task — the higher the priority of an eye-bot's task, the lower the probability of releasing foot-bots. Eye-bots with high priority tasks would thus be likely to fill their quotas at the expense of eye-bots with low priority tasks. In this study, we focus on the underlying balancing property that would enable this kind of probabilistic priority mechanism.



(a) (b) $q = 0.01$; Quotas: $[15, 10, 20]$ (c) $q = 0.00001$; Quotas: $[15, 10, 20]$

Fig. 2. (a) A schema of the mathematical model. (b,c) Experiments run using the mathematical abstraction of the *Constant-q* strategy for two different values of q. The eye-bot quotas for both experiments are: eye-bot 1: 15, eye-bot 2: 10, eye-bot 3: 20.

experimental setup to determine whether the different configurations of our system display the desirable properties discussed in Section III. In our setup, we use a rectangular arena containing 30 non-aggregated foot-bots and 3 eye-bots. Our experiment consists of three phases.

In the first phase, only eye-bot 1 and eye-bot 2 are active. The quotas of eye-bots 1 and 2 are always set so that their sum is less than 30. This first phase therefore allows us to test whether the system is capable of stabilising to the correct group sizes in the simple case where there are enough foot-bots to satisfy all eye-bot quotas. After a certain amount of time T_1, eye-bot 3 is activated. This initiates the second phase, in which all three eye-bots are active. This phase tests the redistribution property of the system in response to the arrival of a new eye-bot. It also tests the balancing property of the system, as eye-bot 3's quota is always chosen so as to make the sum of the quotas greater than the total number of foot-bots in the arena. We also always set eye-bot 3's quota to be greater than or equal to the quotas of eye-bot 1 and eye-bot 2, thus requiring the system to redistribute foot-bots efficiently. The third and final phase is initiated at time T_2 by the disactivation of eye-bot 2, and again tests both the redistribution and the balancing properties of the system, this time in response to the addition of more free foot-bots into the arena. The stabilisation property of the system can be tested in all three phases of the simulation based experiments—ideally, each eye-bot should be able to detect stabilisation in all phases.

VI. An Initial Implementation

In this section, we describe and analyse a simple implementation of our system. For each eye-bot, we set the join probability proportional to the eye-bot's quota of desired foot-bots. We keep the leave probability common for all eye-bots and constant over time. Therefore, we subsequently refer to this implementation as the *Constant-q* strategy.

Figure 2(a) illustrates the abstraction of the system that we use for mathematical analysis. In Figure 2(a), the arena is rectangular and the communication range of each eye-bot is drawn as a circular grey area. Let the total number of foot-bots in the arena be n and the number of foot-bots aggregated at time step t under eye-bot i be $g_i(t)$. Then, the fraction $x_i(t)$ of foot-bots aggregated under eye-bot i at time t is given by:

$$x_i(t) = \frac{g_i(t)}{n}.$$

Analogously, the fraction $x_0(t)$ of free (i.e. not part of any eye-bot's aggregated group) foot-bots present in the arena at time t is given by:

$$x_0(t) = \frac{n - \sum_i^n g_i(t)}{n} = 1 - \sum_i^n x_i(t).$$

We define p_i and q as constant probability parameters of our model. During a single time step, each free foot-bot (i.e., every foot-bot that is not part of any eye-bot's aggregated group) has probability p_i to join the aggregated group under eye-bot i. Each aggregated foot-bot has probability q of disaggregating (i.e., leaving the group of foot-bots which it had previously joined).

Foot-bots perform random walk with obstacle avoidance, thus spreading uniformly in the environment (see Section VII for details). When a foot-bot enters the communication range of eye-bot i, it joins the aggregate with a constant probability j_i. Under these assumptions, an easy way to calculate p_i is the following:

$$p_i = \frac{\text{Area(eye-bot)}}{\text{Area(arena)}} j_i$$

The analytical expression of the mathematical model is then:

$$\begin{cases} x_0(t+1) = x_0(t) - (\sum_i^n p_i)x_0(t) + q\sum_i^n x_i(t) \\ x_1(t+1) = x_1(t) + p_1 x_0(t) - q x_1(t) \\ \quad\vdots \\ x_n(t+1) = x_n(t) + p_n x_0(t) - q x_n(t) \end{cases}$$

and by imposing the steady-state condition

$$\forall k \in [0, n] \qquad x_k(t+1) \equiv x_k(t) \equiv x_k^*$$

we can derive the expression for x_k^* at convergence (see [10] for the full mathematical derivation):

$$x_0^* = \frac{q}{\sum_i^n p_i + q}$$
$$x_1^* = \frac{p_1}{\sum_i^n p_i + q}$$
$$\vdots$$
$$x_n^* = \frac{p_n}{\sum_i^n p_i + q}$$

Figures 2(b) and 2(c) illustrate the dynamics of our mathematical model of the *Constant-q* strategy for two different values of q. Notice that for $q = 0.01$ a significant portion of foot-bots is free when the system reaches its first equilibrium with two eye-bots, while for $q = 0.00001$ this portion is very close to zero. At $t = T_1$, the third eye-bot is added. Since $q > 0$, some foot-bots leave their aggregated groups and a new equilibrium is reached (redistribution). The value of parameter q affects the speed of convergence[2] to the new equilibrium: for $q = 0.01$ convergence in this second phase is three orders of magnitude faster than for $q = 0.00001$ because many more foot-bots are free to join eye-bot 3. At $t = T_2$, eye-bot 2 leaves the arena and frees all its foot-bots. A new equilibrium is quickly reached in about 100 time steps.

VII. Transferal onto Robotic Platform

A. Counting and Probabilities

In the mathematical model, there is no sense of convergence to a particular number of robots. Each abstract eye-bot converges to a fraction of the total number of robots proportional to its join probability (j_i). The first step in transferring our system onto an embodied platform is, therefore, to introduce the notion of counting.

Counting is achieved by each eye-bot monitoring the number of foot-bots underneath it using its camera. For each eye-bot i, when the aggregated foot-bot group size is smaller than its quota, it must have a non-zero join probability ($j_i > 0$). As soon as its quota is filled, the eye-bot prevents any more foot-bots from joining its aggregated group, by setting its join probability to zero ($j_i = 0$).

Clearly, we must somehow choose values for j_i and q. A natural choice for the value of j_i is to consider the fraction of the maximum foot-bot aggregated group size represented by the quota of eye-bot i. The maximum aggregated group size is determined by the physical size of the foot-bot and the communication range of the eye-bot[3]. We therefore set:

$$j_i = \frac{\text{eye-bot } i \text{ 's quota}}{\text{max foot-bot aggregated group size}}.$$

To find a sensible value of q is non–trivial—the impact of q may even depend on the density of robots in the arena. An incorrect choice of q can have dramatic effects on system performance. Note that the mathematical model treats probabilities p_i and q as the rates of free foot-bots joining and leaving aggregates respectively, while in reality this is only true as an average over time. In a realistic implementation, fluctuations are always present and perturb the expected dynamics of the system. We found that when q is only one order of magnitude smaller than p_i, the rate of robots leaving eye-bot i's aggregate over time is too high, thus making that aggregate unstable. On the other hand, choosing too low a value for q hinders redistribution. The role of q in this system is discussed in more detail in Section VII-F.

B. Physical Interference

The physical dynamics of embodied agents make the successful transferal of any model onto a real robotic platform non-trivial. We found that in a naïve implementation of our system, physical interference between robots disrupted the effects of the transmitted probabilities, catastrophically changing the resulting equilibrium. For example, a moving foot-bot would take a long time to traverse the space underneath an eye-bot that had already aggregated a large number of foot-bots, due to the overhead of using obstacle avoidance to thread a path through the aggregated foot-bots. In such cases, both arriving (non-aggregated) and leaving (disaggregating) foot-bots would repeatedly apply the join probability transmitted by the eye-bot. As a result, foot-bots tended never to leave large aggregates.

We also noticed other more direct physical effects. For example, to let disaggregating foot-bots leave an aggregate, it was necessary to make the other aggregated foot-bots get out of the way (using obstacle avoidance). However, this had the result that aggregated foot-bots positioned near the edge of an eye-bot's signal range often got pushed out of the aggregate by leaving foot-bots.

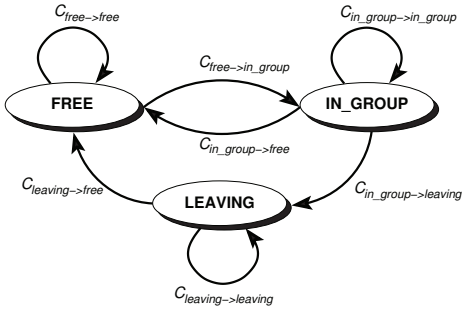
To get around these issues, we implemented a clustering technique, in which aggregating foot-bots were attracted to each other and to the centre of the eye-bot with which they were associated. Using this clustering algorithm as a base, our high level foot-bot state transition behaviour was a much closer match to the behaviour of the mathematical model presented in the previous section.

C. Clustering Algorithm

Our distributed clustering algorithm uses only local interactions. We consider each foot-bot to be immersed in two virtual potential fields. The minimum energy of the first potential field is at the point of the vertical projection of the centre of the eye-bot on the ground. The foot-bots calculate this potential field using information from the transmitted range and bearing messages coming from an eye-bot. The second potential field is analogous to the behaviour of molecules in physical systems. Using its camera, each aggregated foot-bot measures the distance to its neighbouring foot-bots and calculates the potential field whose minimum energy configuration is at distance σ_S to all neighbouring foot-bots. Each foot-bot superimposes its two fields to come up with a resultant force that it translates into appropriate angular velocities that it applies to its wheel actuators. The result is that the local system of aggregated foot-bots converges quickly towards its minimum energy configuration: a hexagonal lattice of foot-bots centred around the vertical projection of the eye-bot under which they are aggregating [14].

[2]Convergence in the mathematical model is the equivalent of the stabilisation property in the embodied robotic implementation (see section VII).

[3]By manually experimenting with the foot-bot hardware prototype, we came up with the (approximate) value of 25 as the number of foot-bots that could aggregate under an eye-bot without being so close as to collide.



State transition conditions	
$C_{free \to in\_group}$	`WithinRange()` = *true* and `Rand()` $< j_i$
$C_{free \to free}$	`WithinRange()` = *false* or `Rand()` $> j_i$
$C_{in\_group \to free}$	`WithinRange()` = *false*
$C_{in\_group \to leaving}$	`WithinRange()` = *true* and `Rand()` $< q$
$C_{in\_group \to in\_group}$	`WithinRange()` = *true* and `Rand()` $> q$
$C_{leaving \to free}$	`WithinRange()` = *false*
$C_{leaving \to leaving}$	`WithinRange()` = *true*

Fig. 3. State transition logic for foot-bots at each time step. `WithinRange()` is a function returning *true* when the robot is within the communication range of an eye-bot. `Rand()` is a function returning a random number in $\mathcal{U}(0,1)$. j_i is the join probability for eye-bot i, q is the common disaggregation probability. State transition conditions are represented be the symbol C and a subscript. For example, $C_{in\_group \to in\_group}$ represents the conditions under which an aggregated foot-bot will stay aggregated in its group (i.e., will not disaggregate) in a single time step.

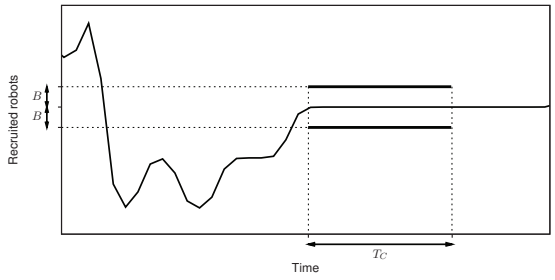


Fig. 4. Convergence detection is based on a tolerance boundary B which is a function of the leaving probability q, the current size of the aggregate g_i and the length of the monitoring period T_C.

Analytically, the potential field attracting a foot-bot towards the centre of its associated eye-bot at distance r_E is given by:

$$V_E(r_E) = \eta r_E^3$$

The equation for the second potential field that manages interactions between two aggregated foot-bots at mutual distance r_F is given by

$$V_F(r_F;\sigma_S) = \epsilon\left[\left(\frac{\sigma_S}{r_F}\right)^4 - 2\left(\frac{\sigma}{r_F}\right)^2\right]$$

Thanks to the fact that force $\vec{F}(r) = -\nabla V(r)$ and defining $\bar{r_E}$ ($\bar{r_F}$) as the normalised vector pointed from the centre of the foot-bot towards the centre of the eye-bot (foot-bot), we can derive

$$\vec{F}_E(r_E) = -3\eta r_E^2 \bar{r_E}$$

and

$$\vec{F}_F(r_F;\sigma_S) = \frac{4\epsilon}{r_F}\left[\left(\frac{\sigma_S}{r_F}\right)^4 - \left(\frac{\sigma_S}{r_F}\right)^2\right]\bar{r}_F.$$

Distance r_E is obtained from the range and bearing sensor, and distance r_F from the camera. At each time step, each aggregated foot-bot calculates $\vec{F}_E(r_E)$ and $\vec{F}_F^m(r_F^m;\sigma_S)$ for each neighbour m. The resulting force

$$\vec{F}_{IN\_GROUP} = \vec{F}_E(r_E) + \sum_m \vec{F}_F^m(r_F^m;\sigma_S)$$

is then directly transformed into wheel actuation on the foot-bot.

A simple extension to this system provides an elegant solution to allow foot-bots to leave an aggregated group while causing minimum disruption to the rest of the group. To differentiate leaving foot-bots, we make them illuminate their blue LEDs. It then suffices to define an interaction potential between red (aggregated) foot-bots and blue (leaving) foot-bots $\vec{F}_F(r_F;\sigma_L)$ with $\sigma_L > \sigma_S$. This new potential creates a 'bubble' in the hexagonal lattice of aggregated foot-bots around the leaving foot-bot. The choice of σ_L is done in such a way that the bubble is large enough for a leaving foot-bot to pass through. For leaving foot-bots we also invert the potential field that attracts it to the eye-bot, resulting in an applied repulsive potential of $-V_E(r_E)$. With this extension, the composite force for an aggregated foot-bot becomes:

$$\vec{F}_{IN\_GROUP} = \vec{F}_E(r_E) + \sum_m \vec{F}_F^m(r_F^m;\sigma_S) + \sum_l \vec{F}_F^l(r_F^l;\sigma_L).$$

For a leaving (blue) foot-bot, on the other hand, the composite force applied is:

$$\vec{F}_{LEAVE} = -\vec{F}_E(r_E) + \sum_m \vec{F}_F^m(r_F^m;\sigma_S) + \sum_l \vec{F}_F^l(r_F^l;\sigma_L).$$

D. Foot-bot state transition logic

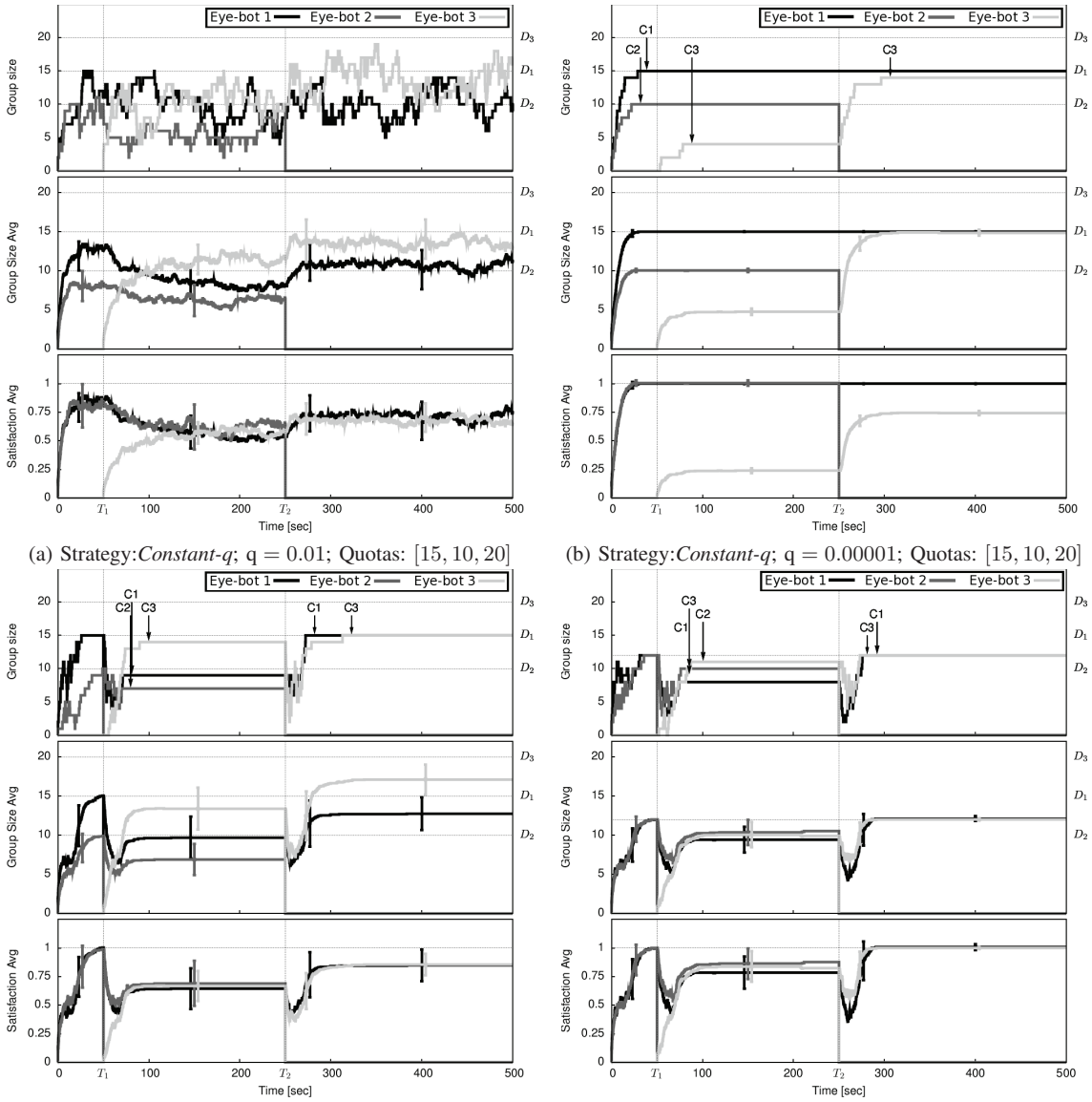
Using the clustering algorithm as a base, we define three states that the foot-bots can be in:

- state FREE: Foot-bot does not belong to any eye-bot's aggregated group. Foot-bot performs random walk with obstacle avoidance. This state is signalled with LEDs lit up in green.
- state IN_GROUP: Foot-bot is part of an aggregated group under an eye-bot. This state is signalled with LEDs lit up in red.
- state LEAVING: Foot-bot is leaving an aggregated group to which it previously belonged. This state is signalled with LEDs lit up in blue.

Eye-bots count the number of foot-bots aggregated beneath them by using their cameras to count the number of foot-bots in state IN_GROUP (i.e. lit up in red). Figure 3 shows the state transition logic of foot-bots upon receipt of a message from an eye-bot.

E. Stabilisation Detection

To detect stabilisation, each eye-bot monitors the fluctuations of the number of its aggregated foot-bots over a period T_C (see Figure 4). If fluctuations stay within some tolerance boundaries for the entire period, the eye-bot



(a) Strategy:*Constant-q*; q $= 0.01$; Quotas: $[15, 10, 20]$

(b) Strategy:*Constant-q*; q $= 0.00001$; Quotas: $[15, 10, 20]$

(c) Strategy:*Decaying-q*; Quotas: $[15, 10, 20]$

(d) Strategy:*Decaying-q*; Quotas: $[12, 12, 12]$

Fig. 5. Simulation results. Each group of 3 plots (a,b,c,d) describes a set of 20 experiments conducted with a particular system configuration. For each set, the top graph represents a single representative experiment that we selected from the set of 20, the middle graph shows the average results of all experiments (bars at selected times indicate std.dev.), the bottom graph shows the percentage of each eye-bot's quota that is filled. In every experiment, eye-bots 1 and 2 are active at the start of the experiment, eye-bot 3 is introduced at time $T_1 = 50$ s, eye-bot 1 leaves the experiment at time $T_2 = 250$ s. Quotas for eye-bots 1, 2 and 3 are given in correspondingly ordered square brackets, and shown by the correspondingly numbered symbols D_1, D_2, D_3. Symbols $C1, C2, C3$ indicate the moments at which the correspondingly numbered eye-bot detected stabilisation.

considers the system to have stabilised to its steady state. The tolerance boundary B is defined as a function of the leaving probability q, the current size of the aggregate g_i and the length of the monitoring period T_C:

$$B = \sqrt{T_C g_i q(1-q)}$$

B is derived by considering the changing number of aggregated foot-bots under an eye-bot as a time series. B is simply the formula for the standard deviation of such a time series over a given monitoring period.

F. Results

Figures 5(a) and 5(b) show the results of experiments using the *Constant-q* model with simulated robots. Looking at the sample run (top plots), we can clearly see that a lower value for q (0.00001) makes stabilisation possible (Figure 5(b)). With the higher value of q (0.01) (Figure 5(a)), there was too much noise in the system for stabilisation to be detected. However, using the lower value of q, the system does not display effective redistribution or balancing. Looking at the middle and bottom plots, we can see that for both q values, on average eye-bots 1 and 2 successfully aggregate the correct number of foot-bots between time 0 s and time T_1 (with the higher value of q displaying more noise). Once eye-bot 3 is introduced at time T_1, with $q = 0.00001$ the system does not display redistribution and balancing— the system arrives at equilibrium with eye-bot 3 having fulfilled a much smaller percentage of its quota than eye-bots 1 and 2. By contrast, with $q = 0.01$, the system displays effective redistribution

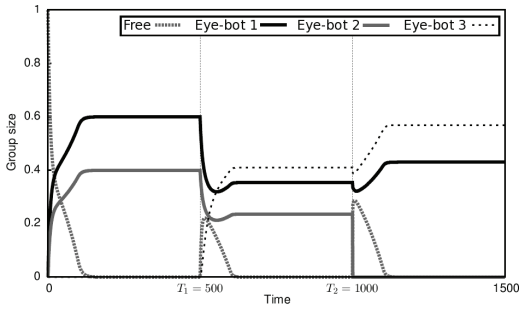


Fig. 6. *Decaying-q* strategy. Dynamics of the mathematical model. Eye-bots 1 and 2 are introduced at $t = 0$, Eye-bot 3 is introduced at $t = 500$, Eye-bot 1 leaves at $t = 1000$.

and balancing—after the introduction of eye-bot 3, the system on average quickly arrives at an equilibrium where all three eye-bots have filled the same percentage of their quota.

VIII. Decaying-q Strategy

Experiments in the previous section indicate that the *Constant-q* strategy allows the system to display the redistribution property or the stabilisation property but not both at the same time. To solve this problem, we modify our system so that individual eye-bots vary q between two values during system execution—a high q value ($q = 0.05$) that allows for efficient foot-bot redistribution, and a low q value ($q = 0.00001$) that encourages stabilisation. Parameter q is 'spiked' to the high value when the system starts, and whenever an eye-bot is activated or deactivated. After a spike, q is exponentially decayed to the low value over 20 seconds and then remains at the low value until the next spike.

The dynamics of a mathematical model of this strategy are shown in Figure 6. We can see that the new system shares the positive features of the previous linear model using both high and low q. After the introduction of eye-bot 3, the new system converges even faster than the previous *Constant-q* model with the high q of 0.01.

On the embodied robotic platform, at the moment that an eye-bot either enters the system or leaves the system, the eye-bot uses its horizontal range and bearing system (see Figure 1(d)) to communicate to local eye-bots that they should spike their q values. The results of running this model with the simulated robots are shown in Figure 5. We have run the system with three eye-bots, but with two different sets of target quotas—[15,10,20] (Figure 5(c)) and [12,12,12] (Figure 5(d)). In both cases, we can see that the system displays stabilisation, redistribution and balancing.

IX. Conclusions and Future Work

In this paper, we adapted an existing model for cockroach aggregation, in which the group sizes were determined a priori by the environment, and transformed it into an active model that could dynamically control group size. We demonstrated properties of our system with a mathematical analysis, then showed how the system could be implemented using simulated versions of a real-world robotic platform. We enhanced our system to show that the seemingly contradictory goals of redistribution and stabilisation could be achieved by a single system.

We believe that the underlying dynamics of our system are sufficiently simple that they could be implemented in other heterogeneous robotic platforms. To this end, we think it would be interesting to try other less explicit communication modalities (e.g. light intensity) as a means of transmitting probabilities.

We are currently investigating the scalability of the system as we introduce larger numbers of eye-bots. By leveraging local communication, we aim to restrict the effects of perturbations (introduction and removal of tasks and/or robots) to local regions of the system. We are also trying to embed our system as part of a more complete task execution scenario, of the type outlined in the introduction.

Acknowledgements. This research was carried out in the framework of Swarmanoid, a project funded by the Future and Emerging Technologies programme (IST-FET) of the European Commission under grant IST-022888. Marco Dorigo acknowledges support from the Belgian F.R.S.-FNRS, of which he is Research Director.

References

[1] M. Dorigo, V. Trianni, E. Sahin, R. Gross, T.H. Labella, G. Baldassarre, S. Nolfi, J.-L. Deneubourg, F. Mondada, D. Floreano, and L.M. Gambardella. Evolving self-organizing behaviors for a swarm-bot. *Autonomous Robots*, 17(2-3):223–245, 2004.

[2] S. Garnier, C. Jost, R. Jeanson, J. Gautrais, M. Asadpour, G. Caprari, and G. Theraulaz. Aggregation behaviour as a source of collective decision in a group of cockroach-like robots. *Advances in Artificial Life*, 3630:169–178, 2005.

[3] R. Jeanson, C. Rivault, J.-L. Denebourg, S. Blanco, R. Fournier, C. Jost, and Guy Theraulaz. Self-organized aggregation in cockroaches. *Animal Behavior*, 69:169–180, 2004.

[4] A. Ledoux. Étude experimentale du grégarisme et de l'interattraction sociale chez les Blattidés. *Annales des Sciences Naturelles Zoologie et Biologie Animale*, 7:76–103, 1945.

[5] A. Martinoli, A.-J. Ijspeert, and F. Mondada. Understanding collective aggregation mechanisms: from probabilistic modelling to experiments with real robots. *Robotics and Autonomous Systems*, 29:51–63, 1999.

[6] C. Melhuish, O. Holland, and S. Hoddell. Convoying: Using chorusing to form travelling groups of minimal agents. *Robotics and Autonomous Systems*, 28:207–216, 1999.

[7] L.E. Parker. ALLIANCE: an architecture for fault tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, 1998.

[8] J.K. Parrish and W.M. Hammer. *Animal Groups in Three Dimensions*. Cambridge University Press, UK, 1997.

[9] C. Pinciroli. Object retrieval by a swarm of ground based robots driven by aerial robots. Mémoire de DEA, Universite Libre de Bruxelles, Bruxelles, Belgium, 2007.

[10] C. Pinciroli and R. O'Grady. A mathematical framework for symbiotic recruitment. Technical Report TR/IRIDIA/2009-006, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium, 2009.

[11] J. Pugh and A. Martinoli. Relative localization and communication module for small-scale multi-robot systems. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA2006)*, pages 188–193, 2006.

[12] J. Roberts, T. Stirling, J. Zufferey, and D. Floreano. Quadrotor using minimal sensing for autonomous indoor flight. In *Proceedings of the European Micro Air Vehicle Conference and Flight Competition (EMAV2007)*, 2007.

[13] M.K. Rust, J.M. Owens, and D.A. Reierson. *Understanding and controlling the german cockroach*. Oxford University Press, UK, 1995.

[14] W. Spears, D. Spears, J. Hamann, and R. Heil. Distributed, physics-based control of swarms of vehicles. *Autonomous Robots*, 17(2-3), 2004.

[15] G.S. Sukhatme, J.F. Montgomery, and R.T. Vaughan. Experiments with aerial-ground robots. In T. Balch and L.E. Parker, editors, *Robot Teams: From Diversity to Polymorphism*, pages 345–367. AK Peters, 2001.

[16] A. Zangwill. Statistical physics: Advances in aggregation. *Nature*, 411:651–652, 2001.