Modelling Swarm Robotic Systems: A Case Study in Collective Foraging

Wenguo Liu, Alan F.T. Winfield and Jin Sa
Bristol Robotics Laboratory
Faculty of Computing, Engineering and Mathematical Sciences
University of the West of England,Bristol
BS16 1QY
{wenguo.liu, alan.winfield}@brl.ac.uk, jin.sa@uwe.ac.uk

Abstract

Understanding the effect of individual parameters on the collective performance of swarm robotic systems in order to design and optimise individual robot behaviours is a significant challenge. This paper presents a probabilistic model of a swarm foraging task. A number of difference equations are derived to describe the probabilistic finite state machine for the task at a macroscopic level. A geometrical approach is then developed in order to estimate the state transition probabilities by considering the simple strategies of individual robots. The paper validates the mathematical model using the simulation tools Player/Stage, and results show the model achieves excellent agreement with simulation.

1. Introduction

Inspired by biological systems, swarm robotics is a relatively new approach to the coordination of a large number of simple robots. Despite the limited sensing, communication and computation abilities of each individual robot in the swarm, complex self-organised behaviours, which cannot be observed at the individual level, can emerge from local interactions among robots and between the robots and the environment. Advantages of the swarm robotics approach lie in the properties of scalability, adaptivity and robustness of the group. However, one of the challenges in swarm robotics is to understand the effect of individual parameters on the group performance in order to design and optimise individual robot behaviours and hence achieve the desired collective swarm properties. Much work has been done with either simulation or real robots to test the performance of the group and consequently change individual robot behaviours and their parameters using a trial and error process, but relatively little work has approached this problem with mathematical modelling. A probabilistic model, using both microscopic and macroscopic approaches has, however, been successfully applied to analysis of collective swarm behaviour. Martinoli and coworkers used a microscopic approach to study collective behaviour of a swarm of robots engaged in cluster aggregation (Martinoli et al., 1999) and collaborative stick-pulling (Ijspeert et al., 2001), in which a 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. In contrast, Lerman developed a macroscopic model, as widely used in physics, chemistry, biology and the social sciences, to directly describe the collective behaviour of the robotic swarm. A class of macroscopic models have been used to study the effect of interference in a swarm of foraging robots (Lerman, 2002) and collaborative stick-pulling (Lerman et al., 2002; Martinoli and Easton, 2004).

In our previous work, we described a group of foraging robots with an adaptation mechanism to optimise the energy income to the swarm (Liu et al., 2006). Three adaptation rules are introduced based on local sensing and communications. Individual robots use internal cues (successful food retrieval), environmental cues (collisions with teammates while searching for food) and social cues (teammate success in food retrieval) to dynamically vary the time spent foraging or resting. Simulation results show the adaptation mechanism is able to guide the swarm towards energy optimisation. However, it is unclear how closely the swarm with the adaptation mechanism approaches optimal energy efficiency; such optimal values remain unknown as all the parameters for the adaptation algorithm have hitherto been selected on a trial and error basis and there are no obvious guidelines for fine tuning these parameters. Following an approach similar to those presented by Martinoli and Lerman, we will develop a macroscopic probabilistic model to describe the foraging task for a swarm of homogeneous robots in which each robot has a number of time parameters, and mathematically explore how those parameters effect the group performance.

This paper is organised as follows: a probabilistic finite state machine (PFSM) for swarm foraging is introduced in section 2, we then develop the difference equations to describe the PFSM in section 3. The estimation of all the probabilities and time parameters are presented based on geometric considerations in section 4; we validate the model using simulation in section 5 and conclude the paper in section 6.

2. Probabilistic Finite State Machine for Swarm Foraging

Consider the swarm foraging scenario: there are a number of food-items randomly scattered in the arena and as food is collected more will, over time, 'grow' to replenish the supply. A swarm of robots are searching and retrieving food-items back to the 'nest'. Each food-item collected will deliver an amount of energy to the swarm but the activity of foraging will consume a certain amount of energy at the same time. The foraging task can be described as a probabilistic finite state machine (PFSM) as shown in Fig.1.

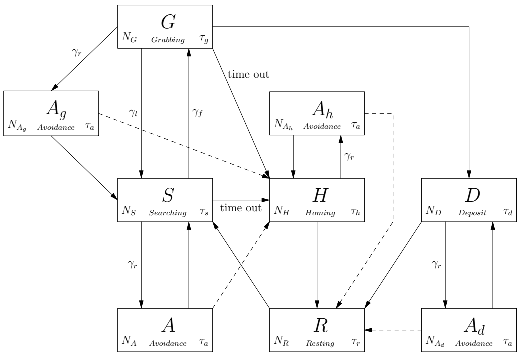

Figure 1: PFSM for foraging task.

By default, a robot will be in state *searching*, denoted by S. In each time step, it has probability γ_f to find a food-item thus move to state *grabbing*, denoted G, in which it will move towards the target food-item until it is close enough to grab it using the gripper. Once the robot successfully grabs the food-item it will move to state *deposit*, say D, in which the robot moves back to the 'nest' carrying the food-item and unloads it. After the robot has unloaded the food-item it will remain in state *resting*, denoted R, for τ_r seconds and then move to state *searching*. Meanwhile, for the robot in state S, if it fails to find a food-item within time τ_s, it will move to state *homing*, denoted H, resulting in the robot moving back to the 'nest' to save energy or minimise interference with other robots. The same rule applies to the robot in state G if its searching time limit is reached even though it may be moving towards a food-item. Due to the competition among robots more than one robot may see the same food-item and thus move towards it at the same time; clearly only one of them can grab it, a robot in state G therefore has probability γ_l to lose sight of the food-item if it has already been grabbed by another robot, which in turn drives the robot back to state *searching*.

For robots moving in the arena, they may collide with each other because of limited space. Thus robots in states S, G, D and H will move to state *avoidance* A, A_g, A_d and A_h respectively with probability γ_r, as shown in Fig.1. The avoidance behaviour takes τ_a seconds to complete before the robot moves back to its previous state. The transitions in Fig.1 can be divided into two categories; one is triggered by the probability an event occurs, such as catching sight of a food-item or collision with other robots, the other is triggered by the time duration spent in one specific state. For instance, the transition between state R and S is triggered by the time the robot rests at the 'nest' while transfer from state S to H is triggered by the time the robot has been engaged in searching. Note that for the robots in states G, H and D, they are all moving towards a specific target which is either a food-item or the 'nest', hence τ_g, τ_h and τ_d represent the average time the robot will stay in the corresponding state before moving to another state. Note that the transitions from state *avoidance* to other states depend not only on the parameters τ_a but also the parameters τ_s, τ_h or τ_d; the latter are indicated by dotted lines in Fig. 1 with higher priority.

3. Mathematical Description of the PFSM

We now consider the average number of robots in each state based on the probabilistic finite state machine of Fig.1 which describes the foraging task at the level of the individual robot. The change in the average number of robots in each state can be mathematically described with a set of difference equations in the discrete-time domain. The time parameters τ_s, τ_g, τ_h, τ_d, τ_r and τ_a are therefore discretised to be T_s , T_g, T_h, T_d, T_r and T_a accordingly. The complete set of equations are given in the following sections.

3.1 Master Difference Equations

Let $N_S(k)$, $N_G(k)$, $N_D(k)$, $N_H(k)$ and $N_R(k)$ be the average number of robots in state *searching*, *grabbing*, *deposit*, *homing* and *resting* respectively in time step k. Also let $N_A(k)$, $N_{A_g}(k)$, $N_{A_d}(k)$ and $N_{A_h}(k)$ be the average number of robots in states *avoidance* which transfer from states *searching*, *grabbing deposit* and *homing* respectively. Obviously, the total number of robots in the swarm must remain constant from one time step to the next. If N_0 represents the total number of robots in the swarm, then we have

$$\begin{aligned} N_0 = & N_S(k) + N_R(k) + N_G(k) + N_D(k) + N_H(k) \\ & + N_A(k) + N_{A_h(k)} + N_{A_g}(k) + N_{A_d}(k) \end{aligned} \tag{1}$$

Following the full PFSM for swarm foraging shown in Fig.1, we can derive a number of difference equations to describe the changes of average number of robots in each state. Consider $N_S(k)$ first, we have

$$\begin{aligned} N_S(k+1) = & N_S(k) + \gamma_l(k) N_G(k) + \Delta_R(k - T_r) \\ & + \left[\Delta_A(k - T_a) - \mathit{\Omega}_A(k - T_a)\right] \\ & + \left[\Delta_{A_g}(k - T_a) - \mathit{\Omega}_{A_g}(k - T_a)\right] \\ & - \left[\gamma_r(k) + \gamma_f(k)\right] N_S(k) \\ & - \mathit{\Gamma}_S(k+1) \end{aligned} \tag{2}$$

The second term on the right-hand side of Eq.(2) represents the number of robots losing sight of a food-item while moving towards it; the probability of losing sight of a food-item for one robot, $\gamma_l(k)$, varies from time to time depending on the number of food-items available and the number of robots competing for food collection. The third term on the right-hand side denotes the number of robots transferring from state $resting$, where $\Delta_R(k)$ is the number of robots moving to state *resting* from other states at time step k. The fourth to seventh entries count the number of robots completing their avoidance behaviour successfully, in which $\Delta_A(k)$ and $\Delta_{A_g}(k)$ denote the number of robots moving into state *avoidance* at time step k, while $\varOmega_A(k)$ and $\varOmega_{A_g}(k)$ represent the number of robots in state *avoidance* whose searching time is up and will hence transfer to state *homing* instead of *searching*. The following term describes the number of robots which will transfer to state *avoidance* (for $\gamma_r(k)N_S(k)$) and *grabbing* (for $\gamma_f(k)N_S(k)$), where the probability one robot collides with others, i.e. $\gamma_r(k)$, changes with the number of robots foraging (not in state *resting*) changing, and $\gamma_f(k)$ denotes the probability that the robot catch the sight of at least one food-item. The last term on the right-hand side of Eq.(2) represents the number of robots which will give up the *searching* task and transfer to state *homing*.

Similarly, we can write down the difference equations for the other states as follows.

$$N_R(k+1) = N_R(k) + \Delta_R(k+1) - \Delta_R(k-T_r) \tag{3}$$

$$\begin{aligned} N_H(k+1) = & N_H(k) - \gamma_r(k)N_H(k) + \Delta_H(k+1) \\ & + \left[\Delta_{A_h}(k-T_a) - \varOmega_{A_h}(k-T_a)\right] \\ & - \varGamma_H(k+1) \end{aligned} \tag{4}$$

$$\begin{aligned} N_G(k+1) = & N_G(k) + \gamma_f(k)N_S(k) - \Delta_D(k+1) \\ & - \left[\gamma_l(k) + \gamma_r(k)\right]N_G(k) - \varGamma_G(k+1) \end{aligned} \tag{5}$$

$$\begin{aligned} N_D(k+1) = & N_D(k) + \Delta_D(k+1) - \gamma_r(k)N_D(k) \\ & + \left[\Delta_{A_d}(k-T_a) - \varOmega_{A_d}(k-T_a)\right] \\ & - \varGamma_D(k+1) \end{aligned} \tag{6}$$

$$\begin{aligned} N_A(k+1) = & N_A(k) + \gamma_r(k)N_S(k) - \varGamma_A(k+1) \\ & - \left[\Delta_A(k-T_a) - \varOmega_A(k-T_a)\right] \end{aligned} \tag{7}$$

$$\begin{aligned} N_{A_h}(k+1) = & N_{A_h}(k) + \gamma_r(k)N_H(k) - \varGamma_{A_h}(k+1) \\ & - \left[\Delta_{A_h}(k-T_a) - \varOmega_{A_h}(k-T_a)\right] \end{aligned} \tag{8}$$

$$\begin{aligned} N_{A_d}(k+1) = & N_{A_d}(k) + \gamma_r(k)N_D(k) - \varGamma_{A_d}(k+1) \\ & - \left[\Delta_{A_d}(k-T_a) - \varOmega_{A_d}(k-T_a)\right] \end{aligned} \tag{9}$$

$$\begin{aligned} N_{A_g}(k+1) = & N_{A_g}(k) + \gamma_r(k)N_G(k) - \varGamma_{A_g}(k+1) \\ & - \left[\Delta_{A_g}(k-T_a) - \varOmega_{A_g}(k-T_a)\right] \end{aligned} \tag{10}$$

Where $\varGamma_S(k)$, $\varGamma_G(k)$, $\varGamma_A(k)$ and $\varGamma_{A_g}(k)$ denote the number of robots that run out of searching time and will transfer to state *homing* at time step k. $\varGamma_D(k)$, $\varGamma_H(k)$, $\varGamma_{A_d}(k)$ and $\varGamma_{A_h}(k)$ represent the number of robots that will move to state *resting* from state *deposit*, *homing* and *avoidance* respectively. $\varOmega_A(k)$, $\varOmega_{A_g}(k)$ represent the number of robots in state *avoidance* when their searching time is up and will transfer to state *homing* instead of previous state *searching* or *grabbing*; similarly, $\varOmega_{A_h}(k)$ and $\varOmega_{A_d}(k)$ denote those robots that will transfer to state *resting* rather than *homing* or *deposit*. The notations $\Delta_R(k)$, $\Delta_G(k)$, $\Delta_A(k)$, $\Delta_{A_h}(k)$, $\Delta_{A_g}(k)$ and $\Delta_{A_d}(k)$ represent the number of robots moving to corresponding state from any other states at time step k, while $\Delta_S(k)$, $\Delta_H(k)$ and $\Delta_D(k)$ depict the increased number of robots to corresponding states. The following equations explain these notations in detail.

$$\begin{aligned} \Delta_R(k+1) = & \varGamma_D(k+1) + \varGamma_{A_d}(k+1) \\ & + \varGamma_H(k+1) + \varGamma_{A_h}(k+1) \end{aligned} \tag{11}$$

$$\Delta_S(k+1) = \Delta_R(k-T_r) \tag{12}$$

$$\begin{aligned} \Delta_H(k+1) = & \varGamma_S(k+1) + \varGamma_G(k+1) \\ & + \varGamma_A(k+1) + \varGamma_{A_g}(k+1) \end{aligned} \tag{13}$$

$$\Delta_G(k+1) = \gamma_f(k)N_S(k) \tag{14}$$

$$\begin{aligned} \Delta_D(k+1) = & \left[\Delta_G(k-T_g) - \varOmega_G(k-T_g)\right] \\ & * \varLambda_G(k;T_g) \end{aligned} \tag{15}$$

$$\Delta_A(k+1) = \gamma_r(k)N_S(k) \tag{16}$$

$$\Delta_{A_h}(k+1) = \gamma_r(k)N_h(k) \tag{17}$$

$$\Delta_{A_g}(k+1) = \gamma_r(k)N_g(k) \tag{18}$$

$$\Delta_{A_d}(k+1) = \gamma_r(k)N_d(k) \tag{19}$$

$\varOmega_G(k)$ in Eq.(15) represents those transfers to state *grabbing* from state *searching* at time step k but the remaining searching time credit is not enough to allow the robot to grab the food-item successfully, i.e. those robots will move to state *homing* once their searching time is up during the period k to $k+T_g$. $\varLambda_G(k;T_g)$ represents the fraction of robots successfully grabbing the food-item at time step k after spending T_g steps moving towards it. It is equivalent to the probability of no transition to other states during the time interval $[k-T_g, k]$, and can be expressed as follows:

$$\varLambda_G(k;T_g) = \prod_{i=k-T_g}^{k} \left[1 - \gamma_l(i) - \gamma_r(i)\right] \tag{20}$$

As the food-items will 'grow' in the environment at a certain rate, say p_{new}, to replenish the supply, and clearly a proportion will be collected by the robots, equivalent to the number of robots transferring to state *deposit* at time step k, i.e. $\Delta_D(k)$, so $M(k)$ can be expressed as follows.

$$M(k+1) = M(k) + p_{new} - \Delta_D(k) \tag{21}$$

3.2 sub-PFSM for "searching-grabbing" task

We now consider the number of robots which give up the search task after their searching time is up. It is clear that only the robots which move into state *searching* from state *resting* at time step $k-T_s$, with quantity $\Delta_S(k-T_s)$, have a chance to time out and hence transfer to state *homing* at time step k. In order to work out the exact number of robots that move into state *homing* at time step k, we therefore only need to take that fraction of robots into account. Figure 2 plots a sub-PFSM for a "searching-grabbing" task extracted from the full PFSM for swarm foraging shown in Fig.1. Note that the lifetime for this sub-PFSM is T_s.

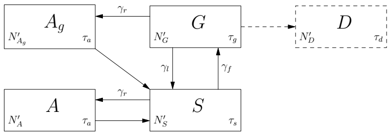

Figure 2: sub-PFSM for robots engaged in the "searching-grabbing" task, some robots will move to state *deposit* meanwhile.

Let $N'_S(i;k)$ be the number of robots in state *searching*, and $N'_G(i;k)$, $N'_{A_g}(i;k)$, $N'_A(i;k)$ for the other states accordingly, where k indicates the date of birth (DOB) of the sub-PFSM from the full PFSM. i indicates the current time step for the sub-PFSM and $i \in [k, k+T_s]$.

Using the same approach presented in section 3.1, we can write down a number of difference equations to describe the changes of the average number of robots for each state in the sub-PFSM shown in Fig.2 as follows.

$$\begin{aligned} N'_S(i+1;k) =& N'_S(i;k) + \Delta'_{A_g}(i-T_a;k) \\ &+ \Delta'_A(i-T_a;k) + \gamma_l(i)N'_G(i;k) \\ &- \big[\gamma_f(i)+\gamma_r(i)\big]N'_S(i;k) \end{aligned} \tag{22}$$

$$\begin{aligned} N'_G(i+1;k) =& N'_G(i;k) + \gamma_f(i)N'_S(i;k) \\ &- [\gamma_r(i)+\gamma_l(i)]N'_G(i;k) \\ &- \Delta'_G(i-T_g;k)\Lambda_G(i;T_g) \end{aligned} \tag{23}$$

$$\begin{aligned} N'_{A_g}(i+1;k) =& N'_{A_g}(i;k) + \gamma_r(i)N'_G(i;k) \\ &- \Delta'_{A_g}(i-T_a;k) \end{aligned} \tag{24}$$

$$\begin{aligned} N'_A(i+1;k) =& N'_A(i;k) + \gamma_r(i)N'_S(i;k) \\ &- \Delta'_A(i-T_a;k) \end{aligned} \tag{25}$$

Where $\Delta'_A(i;k)$, $\Delta'_{A_g}(i;k)$ and $\Delta'_G(i;k)$ represent the number of robots moving to state *avoidance* and *grabbing* at time step i, and can be expressed as

$$\Delta'_A(i+1;k) = \gamma_r(i)N'_S(i;k) \tag{26}$$

$$\Delta'_{A_g}(i+1;k) = \gamma_r(i)N'_{A_g}(i;k) \tag{27}$$

$$\Delta'_G(i+1;k) = \gamma_f(i)N'_S(i;k) \tag{28}$$

Note that $i \in [k, k+T_s-1]$ and the initial condition is $N'_S(k;k) = \Delta_S(k)$. The average number of robots in each state in the sub-PFSM will evolve at the same time that they are evolving in the full PFSM. Consider the meaning of $\varGamma_S(k)$, $\varGamma_G(k)$, $\varGamma_A(k)$ and $\varGamma_{A_g}(k)$ in the master difference equations in section 3.1, then we have

$$\varGamma_S(k) = N'_S(k;k-T_s) \tag{29}$$

$$\varGamma_G(k) = N'_G(k;k-T_s) \tag{30}$$

$$\varGamma_A(k) = N'_A(k;k-T_s) \tag{31}$$

$$\varGamma_{A_g}(k) = N'_{A_g}(k;k-T_s) \tag{32}$$

Clearly, $\varOmega_G(k-T_g)$ is equivalent to the sum of fraction of robots transferring to state *grabbing* at time step $k-T_g$ for each sub-swarm and can be expressed as

$$\varOmega_G(k-T_g) = \sum_{m=k-T_s-T_g+1}^{k-T_s} \Delta'_G(k-T_g;m) \tag{33}$$

Similarly, we have

$$\varOmega_A(k-T_a) = \sum_{m=k-T_s-T_a+1}^{k-T_s} \Delta'_A(k-T_a;m) \tag{34}$$

$$\varOmega_{A_g}(k-T_a) = \sum_{m=k-T_s-T_a+1}^{k-T_s} \Delta'_{A_g}(k-T_a;m) \tag{35}$$

3.3 sub-PFSM for "deposit" and "homing" task

Likewise, we can deduce the equations for $\varGamma_D(k)$, $\varGamma_{A_d}(k)$, and $\varOmega_{A_d}$ using the same methodology. Let us consider the sub-PFSMs for tasks "deposit" and "homing", as shown in Fig.3.

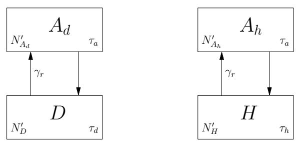

Figure 3: sub-PFSM for robots engaged in sub task "deposit" (left) and "homing" (right).

If $N'_D(i+1;k)$ and $N'_{A_d}(i+1;k)$ denote the number of robots in state *deposit* and *avoidance* respectively, then we have

$$\begin{aligned} N'_D(i+1;k) =& N'_D(i;k) - \gamma_r(i)N'_D(i;k) \\ &+ \Delta'_{A_d}(i-T_a;k) \end{aligned} \tag{36}$$

$$\begin{aligned} N'_{A_d}(i+1;k) =& N'_{A_d}(i;k) + \gamma_r(i)N'_D(i;k) \\ &- \Delta'_{A_d}(i-T_a;k) \end{aligned} \tag{37}$$

$$\Delta'_{A_d}(i+1;k) = \gamma_r(i;k)N'_D(i;k) \tag{38}$$

Where k indicates the DOB of sub-PFSM for the "deposit" task, i indicates the current time step and $i \in [k, k+T_d-1]$. Eq.(36)-(38) will evolve with the full PFSM evolving, with initial condition $N'_D(k;k) = \Delta_D(k)$. Therefore, $\varGamma_D(k)$,

$\mathit{\Gamma}_{A_d}(k)$ and $\mathit{\Omega}_{A_d}$ are given by

$$\mathit{\Gamma}_D(k) = N'_D(k; k - T_d) \tag{39}$$

$$\mathit{\Gamma}_{A_d}(k) = N'_{A_d}(k; k - T_d) \tag{40}$$

$$\mathit{\Omega}_{A_d}(k - T_a) = \sum_{m=k-T_d-T_a+1}^{k-T_d} \Delta'_{A_d}(k - T_a; m) \tag{41}$$

Again, for the sub-PFSM of task "homing", as shown in Fig.3:right, we have

$$\begin{aligned} N'_H(i+1;k) = {} & N'_H(i;k) - \gamma_r(i) N'_H(i;k) \\ & + \Delta'_{A_h}(i - T_a; k) \end{aligned} \tag{42}$$

$$\begin{aligned} N'_{A_h}(i+1;k) = {} & N'_{A_h}(i;k) + \gamma_r(i) N'_H(i;k) \\ & - \Delta'_{A_h}(i - T_a; k) \end{aligned} \tag{43}$$

$$\Delta'_{A_h}(i+1;k) = \gamma_r(i;k) N'_H(i;k) \tag{44}$$

Where $i \in [k, k + T_h - 1]$ and $N'_H(k;k) = \Delta_H(k)$. $\mathit{\Gamma}_H(k)$, $\mathit{\Gamma}_{A_h}(k)$ and $\mathit{\Omega}_{A_h}(k - T_a)$ can be expressed as

$$\mathit{\Gamma}_H(k) = N'_H(k; k - T_h) \tag{45}$$

$$\mathit{\Gamma}_{A_h}(k) = N'_{A_h}(k; k - T_h) \tag{46}$$

$$\mathit{\Omega}_{A_h}(k - T_a) = \sum_{m=k-T_h-T_a+1}^{k-T_h} \Delta'_{A_h}(k - T_a; m) \tag{47}$$

3.4 *the swarm with energy consumption*

For simplicity, let us assume the robot will consume E_r units of energy in state *resting* and αE_r units of energy in other states each time step, where $\alpha \gg 1$. One collected food-item will deliver the swarm with E_c units of energy. If $E(k)$ denotes the net energy income for the swarm then

$$\begin{aligned} E(k+1) = {} & E(k) + E_c \Delta_D(k - T_d) - E_r N_R(k) \\ & - \alpha E_r (N_0 - N_R(k)) \end{aligned} \tag{48}$$

4. Geometrical Estimation of Transition Probabilities and T_g, T_d, T_h

In order to solve the equations in section 3, we must estimate each of the state transition probabilities and time parameters T_g, T_d and T_h (T_s, T_r and T_a are part of the design parameters for the behaviours of each robot in the swarm). These parameters, however, are related to the strategy of the individual robots used and the environmental parameters, such as the forward speed of the robots, the size of arena, etc. Figure 4 shows a typical setup for the foraging task, which we will use to validate the probabilistic model in simulation. There is a home region located in the centre with radius R_h, where the robots will rest and unload the food-items. The whole arena is bounded with a wall to prevent robots moving too far away from the 'nest'. For simplicity, assume the food-items will 'grow' randomly within the ring area between radius R_{inner} and radius R_{outer}, as shown in Fig.4. Based on geometrical considerations and a robot's strategy in the swarm, we can then estimate all required transition probabilities and time parameters as follows.

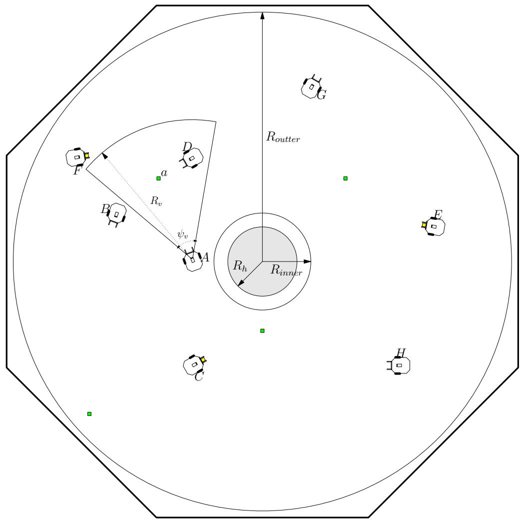

Figure 4: Environmental setup for foraging task.

4.1 *Probability to find food-items $\gamma_f(k)$*

As food-items 'grow' randomly within the arena, the probability that one food-item is found by a robot, denoted by P_f, can be expressed as the ratio of the detection region covered by the robot in each time step to the size of area in which the food-items could be located. If S_{detect} denotes the area covered by the robot and S_f denotes the area the food-items will grow up in, then $P_f = S_{detect}/S_f$. Assume the robot moves forward at speed V, and is equipped with a camera which has a view angle ψ_v and detection range R_v, as the robot A shown in Fig.4, then we have

$$P_f = \frac{S_{dectect}}{S_f} = \frac{\psi_v R_v V \Delta t}{\pi(R^2_{outer} - R^2_{inner})} \tag{49}$$

Where Δt is the duration of one time step. Then the probability that the robot catch the sight of at least one food-item can be expressed as

$$\gamma_f(k) = 1 - (1 - P_f)^{M(k)} \tag{50}$$

Normally, $P_f \ll 1$, then Eq.(50) can be simplified as

$$\gamma_f(k) = P_f M(k) \tag{51}$$

4.2 *Probability of collision with teammate $\gamma_r(k)$*

Consider one robot and teammates within its vicinity, as shown in Fig.5:(a). Robot A may collide with its teammates, robot B and C, only if robot B or C is close enough to trigger robot A's bumper sensors after one time step. As each robot moves around the arena simultaneously, assume the heading of robot B or C relative to robot A varies from $0°$ to $360°$ uniformly, then the relative speed of robot B or C to robot A will vary from 0 to $2V$, as shown in Fig.5:(b). As a result, if robot B or C is located in area $\mathbb{C}$ as shown in Fig.5:(c),

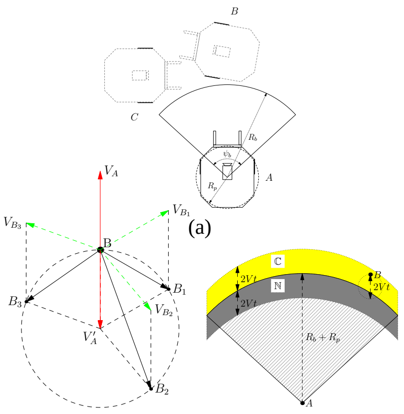

Figure 5: (a): Robots ready to collide with each other. (b): The speed of robot B relative to robot A. (c): Geometrical consideration for probability of collision.

they will probably move into robot A's bumper sensor detection region, the region with hatched lines shown in Fig.5:(c), which in turn forces robot A to move into state *avoidance* in the next time step. Let P_{in} denote the probability that a robot is located in area $\mathbb{C}$, P_a the probability that the robot located in area $\mathbb{C}$ will trigger robot A's bumper sensor next time step, and $N_{active}(k)$ the number of robots working in the arena currently, then the probability that robot A will move to state *avoidance* in the next time step can be expressed as follows

$$\gamma_r(k+1) = 1 - (1 - P_{in}P_a)^{N_{active}(k)-1} \tag{52}$$

Where $N_{active}(k) = N_0 - N_R(k)$. Generally, $P_{in}P_a \ll 1$, then Eq.(52) can be simplified as

$$\gamma_r(k+1) = (N_{active}(k) - 1)P_{in}P_a \tag{53}$$

P_{in} can be obtained with simple geometrical considerations as the ratio of size of area $\mathbb{C}$ to the size of the whole working area, that is

$$P_{in} = \frac{2V\Delta t\psi_b(R_b + R_p)}{\pi(R^2_{outer} - R^2_{inner})} \tag{54}$$

Where R_b and ψ_b is the detection range and detection angle of bumper sensors respectively, and R_p is the radius of the robot. Note that the physical size of robots are approximated by circles, as shown in Fig.5:(a).

To calculate P_a, let us treat each robot as a point located in its mechanical centre, as shown in Fig.5:(c). If we take A as a reference point then, after one time step, robot B will move to a new position which could be any position located on the circle shown in Fig.5:(b). Clearly, if the new position is within area $\mathbb{N}$ it will trigger robot A's bumper sensors. As robot B could possibly be in any position within the area $\mathbb{C}$, its new possible position after one time step, then, could cover anywhere within area $\mathbb{C}$ and $\mathbb{N}$. Therefore P_a is simply the ratio of size of area $\mathbb{N}$ to the total size of areas $\mathbb{C}$ and $\mathbb{N}$, i.e $P_a = 0.5$.

4.3 Probability to lose sight of target $\gamma_l(k)$

The robot will potentially lose sight of the target when it is moving towards a food-item because of the competition among robots. The robots, except those in state *resting*, *homing* and *deposit*, will take part in food competition. Clearly, the more robots in competition, the more likely one robot will lose sight of its target. On the other hand, with more food-items available there is less chance the robot will lose sight of its target. If $N'(k)$ denotes the number of robots competing for food, then we have

$$\begin{aligned} N'(k) =& N_0 - \big[N_R(k) + N_H(k) + N_{A_h}(k) \\ &+ N_D(k) + N_{A_d}(k)\big] \end{aligned} \tag{55}$$

Suppose there are N_{f_a} robots located in the vicinity of food item a within radius R_v. Not all of them will catch sight of food-item a, as robot B shown in Fig.4. Let p_g denote the probability that each of those robots will compete for food-item a (move toward it). Assume each robot has a random heading, then p_g can be estimated based on the following equation

$$p_g = \frac{\psi_v}{2\pi} \tag{56}$$

To calculate γ_l for robot A shown in Fig.4, which is moving towards food-item a (in state *grabbing*), let us assume each robot has the same chance to get close to food-item a. It follows that robot A has $1/N_{f_a}$ probability to be the closest robot. Since robot A is moving towards food-item a, it will finally pick up food-item a successfully after T_g steps if it is the closest robot. In the case that robot A is not the closest robot to food-item a, with probability $(1 - 1/N_{f_a})$, it can still pick up the food-item only if none of the other $N_{f_a} - 1$ robots see food-item a or take a as a target. It is possible that one robot can see more than one, say M_{f_a}, food-items at the same time as the food-items are scattered randomly, but clearly the robot will only be able to move toward and grab one of them, therefore, each food-item has probability $1/M_{f_a}$ to be the target food-item of the robot. For those $N_{f_a} - 1$ robots, the probability of any of them competing for item a can therefore be expressed as $p_g * 1/M_{f_a}$. So robot A will lose sight of food-item a if it is not the closest robot and there is at least one other robot competing for food-item a. On average, it will take $(T_g/2)$ steps for another closer robot to collect the food-item a. The probability that robot A loses sight of food-item a at each time step can therefore be expressed as

$$\gamma_l(k) = 2\left(1 - \frac{1}{N_{f_a}}\right)\left(1 - (1 - \frac{p_g}{M_{f_a}})^{N_{f_a}-1}\right)/T_g \tag{57}$$

As the food-items are randomly scattered, N_{f_a} and M_{f_a} can be obtained with simple geometrical consideration as:

$$N_{f_a} = \frac{R_v^2}{(R^2_{outer} - R^2_{inner})}N'(k) \tag{58}$$

$$M_{f_a} = \frac{\psi_v R_v^2}{2\pi(R^2_{outer} - R^2_{inner})}M(k) \tag{59}$$

4.4 *Time parameters T_g, T_d and T_h*

The average time it takes the robot to move to a target postion depends on the strategy the robot uses. We now estimate the average time the robot will stay in state *grabbing*, i.e. T_g. For simplicity, assume that the robot will turn to face the food-item at speed w_1 until the food-item is located in the centre of the camera view, then the robot will move towards the food-item at normal speed V and grab it as soon as it is close enough. As a result, the average time the robot stays in state *grabbing* is the sum of time spent on 'turning', t_r, 'forward', t_m, and 'loading', t_l, that is,

$$\tau_g = t_r + t_m + t_l \tag{60}$$

Where t_l depends on the response time of grippers and is a design parameter. t_r is decided by the average angle, Φ, the robot needs to turn in order to face the food-item and the rotation speed w_1, i.e. $t_r = \Phi/w_1$. t_m then depends on the average distance, D, the robot needs to move before it can grab the food-item, and the forward speed V, that is $t_m = D/V$. As the food-items grow randomly within the arena, without loss of generality, we can assume $D = R_v$ and $\Phi = \psi_v/2$, thus we obtain

$$\tau_g = \frac{\psi_v}{2w_1} + \frac{R_v}{V} + t_l \tag{61}$$

In a similar manner, the robot will turn and face the direction of the nest as soon as it has grabbed the food-item successfully (move to state *deposit*), and it will then move forward until it reaches the nest region. So we can estimate the average deposit time τ_d as follows.

$$\tau_d = \frac{D_d}{V} + t_f \tag{62}$$

where t_f is the average time the robot spends turning to face home, and D_d is the average distance the robot then travels. Clearly, D_d is related to the distribution of food-items in the arena as the robot will start to move back to the nest from where it grabbed the food-items. The food-items grow up randomly (and uniformly) within the arena according to the task description, therefore D_d is estimated as

$$D_d = \frac{2(R_{outer}^3 - R_{inner}^3)}{3(R_{outer}^2 - R_{inner}^2)} - R_h \tag{63}$$

Let Φ_2 denote the average angle the robot needs to turn in order to face home, and w_2 be the turning speed, then

$$t_f = \Phi_2/w_2 \tag{64}$$

Suppose the robot has a random heading when it starts to turn, then the relative angle it need to turn will vary from $0°$ to $180°$, so that $\Phi_2 = 90°$.

Now we have obtained τ_g and τ_d we can discretise as T_g and T_d by dividing by the duration of each time step, Δt. For a robot in state *homing*, clearly it uses the same strategy to return home except that there is no food-item in its gripper, hence we assume $T_h = T_d$.

5. Validation of the Model

We have used the sensor-based simulation tools Player/Stage (Gerkey et al., 2003) to validate our probabilistic model for the swarm foraging task. Eight robots are each equipped with three bumper sensors to avoid collisions with teammates or walls, three light sensors to find the way back to the 'nest', and a camera to find the food-items scattered in an octagonal arena, as shown in Fig.4. Each robot has the same behaviour set as described in our previous paper (Liu et al., 2006). Table 1 summarises the value of parameters used for simulation.

Table 1: Parameters for simulation

parameters	value	parameters	value
V	$0.15m/s$	R_{outer}	$3m$
w_1	$15°/s$	E_r	$1unit$
w_2	$20°/s$	αE_r	$10unit$
ψ_v	$60°$	E_c	$2000units$
ψ_b	$95°$	Δt	$0.25sec$
R_v	$2m$	t_l	$2sec$
R_b	$0.4m$	τ_a	$2sec$
R_p	$0.13m$	τ_s	$100sec$
R_h	$0.5m$	τ_r	$[0, 200]sec$
R_{inner}	$0.7m$	p_{new}	0.04

We have run six types of experiment by changing the resting time parameters τ_r in simulation, from 0 to 200 seconds with an interval of 40 seconds. Each experiment is repeated 10 times and each simulation lasts for 20000 seconds. With the same parameters used in the simulation we have computed the probabilistic model presented in previous sections. Figure 6 compares the final net energy of the swarm from the simulation and the model. Error bars show the standard deviation for 10 runs of simulation. Fig. 6 clearly shows that results from the probabilistic model match those from simulation very well. In addition, with τ_r increasing from 0 to 160,

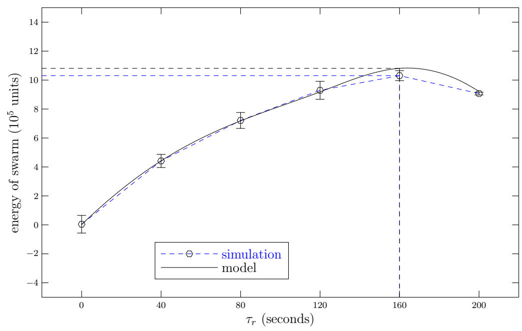

Figure 6: Total net energy of swarm after 20000 seconds for a swarm of 8 robots with different resting time threshold.

the net energy of the swarm increases but will decrease if τ_r is set to 200 seconds, which implies that in order to gain the maximal net energy for the swarm there should be an optimal value of τ_r for individual robots.

Figure 7 plots the instantaneous net energy of the swarm from both simulation and mathematical model with $\tau_r = 80$

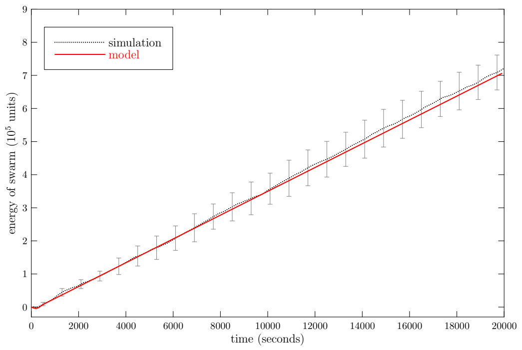

Figure 7: Simulated and predicted instantaneous net energy of the swarm.

seconds. A very good agreement between the predicted and simulated instantaneous net energy of the swarm can be observed clearly. We also notice that the increase of net energy of the swarm over time is almost linear for all simulations. To examine this we plot the instantaneous average

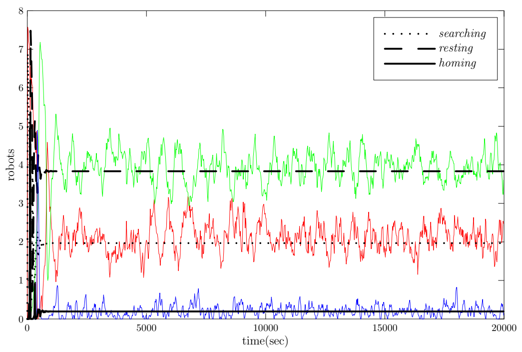

Figure 8: Instantaneous number of robots in states *searching*, *resting* and *homing* for the swarm with $\tau_r = 80$.

number of robots, from both simulation and mathematical model, in states *searching*, *resting* and *homing* for the swarm with $\tau_r = 80$. Note that for clarity the other states are not plotted but show similar relationships between measured and predicted values. As shown in Fig.8, for each state, the average number of robots from the probabilistic model will reach a steady value after some time steps. Alternatively, the average number of robots from simulation oscillates over time but stays near the value predicted by the model. Consider the definition of net energy of the swarm in Eq.(48), it is not suprising that the increase of net energy of the swarm is almost linear as shown in Fig.7. Thus, we have good reason to believe that the model can be used to capture the dynamics of the system with reasonable accuracy.

6. Conclusions and Future work

In this paper we have proposed a probabilistic model for a swarm of foraging robots. A probabilistic finite state machine (PFSM) for the swarm foraging task and its description as a set of difference equations has been presented. With some simplifying assumptions, we have developed a novel geometrical approach for estimation of the state transition probabilities for the PFSM, from consideration of detailed actions of robot behaviours. We have validated our probabilistic model using a sensor-based simulation and the results show excellent agreement between the model and the simulation. We conclude that the model can be used to analyse the effect of individual parameters on the performance of the swarm and to optimise robot parameters. Combining with an appropriate searching technique, for example, a genetic algorithm, the model can be used to find an optimal resting time threshold for individual robots in order to improve the performance. However, as presented in this paper, the model assumes homogeneous robots whereas the full algorithm in (Liu et al., 2006) allows robots to individually adapt their search and rest time parameters to optimise net energy income to the swarm for a given food density. We will expand the model to deal with swarm adaptation in future work.

References

Gerkey, B. P., Vaughan, R. T., and Howard, A. (2003). The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the International Conference on Advanced Robotics*, pages 317–323, Coimbra, Portugal.

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

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

Liu, W., Winfield, A., Sa, J., Chen, J., and Dou, L. (2006). Strategies for energy optimisation in a swarm of foraging robts. In Şahin, E., Spears, W. M., and Winfield, A. F. T., (Eds.), *Second SAB 2006 International Workshop, Rome, Italy, September 30-October 1, 2006 Revised Selected Papers*, volume 4433 of *Lecture Notes in Computer Science*, pages 14–26. Springer.

Martinoli, A. and Easton, K. (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 Gambardella, L. M. (1999). A probabilistic model for understanding and comparing collective aggregation mechansims. In *ECAL '99: Proceedings of the 5th European Conference on Advances in Artificial Life*, pages 575–584, London, UK. Springer-Verlag.