

# Modeling and Analysis of Beaconless and Beacon-Based Policies for a Swarm-Intelligent Inspection System\*

Nikolaus Correll and Alcherio Martinoli

*Swarm-Intelligent Systems Group, École Polytechnique Fédérale de Lausanne*

*CH-1015 Lausanne, Switzerland*

*nikolaus.correll|alcherio.martinoli@epfl.ch*

**Abstract**—We are developing a swarm-intelligent inspection system based on a swarm of autonomous, miniature robots, using only on-board, local sensors. To estimate intrinsic advantages and limitations of the proposed possible distributed control solution, we capture the dynamic of the system at a higher abstraction level using non-spatial probabilistic microscopic and macroscopic models. In a previous publication, we showed that we are able to predict quantitatively the performances of the swarm of robots for a given metric and a beaconless policy. In this paper, after briefly reviewing our modeling methodology, we explore the effect of adding an additional state to the individual robot controller, which allow robots to serve as a beacon for teammates and therefore bias their inspection routes. Results show that this additional complexity helps the swarm of robots to be more efficient in terms of energy consumption but not necessarily in terms of time required to complete the inspection. We also demonstrate that a beacon-based policy introduces a strong coupling among the behavior of robots, coupling which in turn results in nonlinearities at the macroscopic model level.

**Index Terms**—Swarm Intelligence, Swarm Robotics, Microscopic and Macroscopic Models, Nonlinear Dynamical Systems, Inspection, Performance Evaluation

## I. INTRODUCTION

Swarm Intelligence (SI) is an emerging computational and behavioral metaphor for solving distributed problems. SI takes its inspiration from biological examples provided by social insects [3] and by swarming, flocking, herding, and shoaling phenomena in vertebrates. One way to increase our understanding of these natural systems but also to predict the performance of an artificial swarm is to use appropriate modeling techniques. Similar to previous case studies concerned with distributed manipulation of objects (see for instance [1], [7]), we make use of non-spatial probabilistic microscopic and macroscopic models in order to understand general properties of the dynamics of the swarm engaged in an inspection task. In particular, we present an extension of a beaconless distributed control algorithm presented in [4], [5] and compare the performance of this new algorithm with the help of our modeling tools.

In order to evaluate the performance of the swarm as an inspection system, we define two different swarm metrics: time to complete inspection and the energy consumption of the swarm for completing the task.

\*Both authors are currently sponsored by a Swiss NSF grant (contract Nr. PP002-68647/1)



Fig. 1. *Left*: Overview of the turbine set-up in the embodied simulator [5]. *Right*: Overview of the real-robot set-up [4].

## II. THE CASE STUDY: TURBINE INSPECTION

We consider the inspection of jet turbine engines as a case study for the inspection of a regular structure. In order to minimize failures, jet turbine engines have to be inspected at regular intervals. This is usually performed visually using borescopes, a process which is time consuming and cost intensive [6]. One possible solution for speeding up and automating the inspection process is to rely on a swarm of autonomous, miniature robots which could be released into the turbine without disassembling it. While this idea is intellectually appealing and could pave the way for other similar applications in coverage/inspection of engineered or natural, regular structures, it involves a series of technical challenges which dramatically limit possible designs of robotic sensors, and hence emphasizes a SI-driven approach. For instance, the shielded, complex, narrow structure of a turbine imposes not only strong miniaturization constraints on the design but also prevents the use of any traditional global positioning and communication system. Furthermore, a limited on-board energy budget might prevent computation of a sophisticated deliberative planning strategy and dramatically narrows the sensor and communication range of our robots.

### A. Simplification and Simulation of the Turbine Scenario

We simplify the real 3D environment by unrolling the axis-symmetric geometry of the turbine into a flat representation with the blades as vertical extrusions. Snapshots of the scenarios in a realistic, multi-robot, embodied, sensor-based simulator (Webots 4.0, see [8]) and a close-up of a real robot set-up are shown in Fig. 1.

In Webots, simulated sensors and actuators are characterized by precise, user-definable nonlinearities and noise. For instance, in our simulations, all sensors and actuators on-board are characterized by  $\pm 10$  percent white noise. As

## FSM

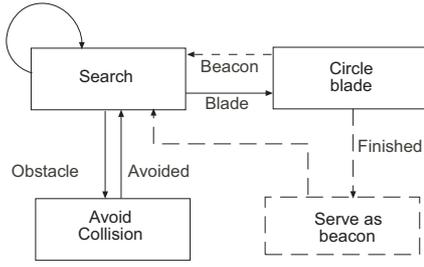


Fig. 2. The high-level behavioral flowchart of the robot controller as a Finite State Machine (FSM).

## PFSM

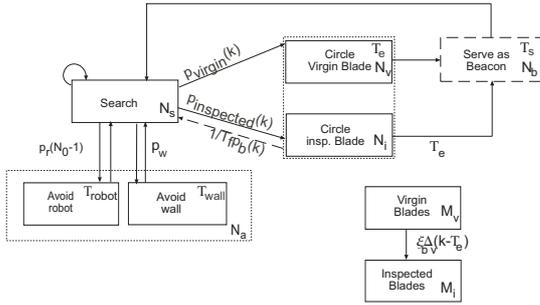


Fig. 3. The corresponding Probabilistic FSM (PFSM) used in the models, capturing details of interest of the schema-based controller. Dashed arrows/states characterize the beacon-based behavior.

shown in previous publications, this simulator can provide realistic results (e.g., [7]) when compared with real robot experiments. It is worth noting that to allow comparison with real robotic experiments [4], we limit the simulated experimental setup to 16 blades in four stages, and the maximum number of robots to 20.

### B. The Robot Controllers

In our case study, we are not concerned with detection of flaws but rather with the individual and group motion in the turbine scenario. For the sake of simplicity, we therefore assume that circumnavigating a blade in its totality is a good emulation of a scanning-for-flaws maneuver. The behavior of a single robot is determined by a schema-based controller [2] that tightly links the platforms' actions to sensor perception while using as little representational knowledge of the world as possible. In a schema-based controller, behavioral responses are represented by vectors generated by local potential fields, and behavioral coordination is achieved by vector addition. Sequencing of behaviors is achieved by a dynamic action-selection mechanism based on internal timers which are in turn set and reset according to schema response. The overall behavior of a robot in a beaconless policy can be summarized as follows (see Fig. 2, without dashed states). The robot searches for blades throughout the turbine, combining schemes that drive the robot forward, avoid obstacles, and follow contours. Using exclusively its on-board sensors, the robot can discriminate between the three different types of objects within the arena: teammates, blades, and external walls.

Teammates and external walls are systematically avoided while blades stimulate a contour-following behavior in the robot. Teammates can be easily differentiated from other objects because of their active emission of infrared signals. A robot can start circumnavigating a blade at any point of its contour but can leave the blade exclusively at its tip. This allows the robot to bias its blade-to-blade trajectory without using any sophisticated navigation mechanisms, exploiting a specific feature of the environmental pattern [4]. The robot can recognize a blade's tip again via a specific sensorial pattern generated by its on-board proximity sensors.

A blade can be left only if a timeout parameter  $T_{max}$  has expired. The corresponding timer is set when the robot attaches to the blade. Hence, we can calculate the mean time  $T_e$ , which is needed to partially circumnavigate a blade—an atomic element of the regular structure—considering a random attaching point, as a function of  $T_{max}$ , and the average time to completely circumnavigate a blade  $T_{fe} = 2T_{he}$ , the time to circumnavigate a half blade. In this paper, we add the following behavior (see Fig. 2, dashed states and arrows): after inspecting a blade, a robot remains at its tip for the time  $T_s$ , where it serves as a beacon and signals to all robots within its range possibly approaching this blade to avoid it, or abort its circumnavigation, in case the other robot has already attached to the blade at another point. Notice that the default signaling range of a robot as a beacon is exactly the same as during search mode, where signaling is used to distinguish amongst robots and other obstacles. Thus, the signaling area only partially overlaps with the blade area; we define the ratio between the perimeter of the blade that is within the signaling area and the whole perimeter of a blade as  $R_b$ .

### III. MICROSCOPIC AND MACROSCOPIC MODELS

The central idea of the probabilistic modeling methodology is to describe the experiment as a series of stochastic events with probabilities computed from the interactions' geometrical properties and systematic experiments with one or two real robots [7] or embodied agents [1]. As more extensively detailed in [1], [4], [5], [7], the modeling methodology relies on the assumption that coverage of the arena by the group of robots is uniform and robots' trajectories and objects' positions in the arena do not play a role in the non-spatial metrics of interest. Consistent with previous publications, we can use the controller's FSM depicted in Fig. 2 as a blueprint to devise the Probabilistic FSM (PFSM or Markov chain) representing an individual agent at the microscopic level or the whole swarm at the macroscopic level. The overall PFSM for the system is represented graphically in Fig. 3, using two coupled PFSMs, one representing the robot(s) and one representing the shared turbine environment. At the microscopic level, a specific state represents the current mode of a certain individual and a given state of a blade (virgin, inspected, partially inspected), while a state at the macroscopic level defines the average number of individuals in the same mode

TABLE I  
ENCOUNTERING PROBABILITIES AND INTERACTION TIMES [5]. TIME  
GRANULARITY OF THE MODEL  $T = 1s$ .

Object	Computed enc. prob.	Detect. area	Mean interaction time
Wall	$p_w = 0.0458$	$420cm^2$	$T_w = 10s$
Blade	$p_e = 0.0106$	$96.48cm^2$	$T_{he} = 10s$
Robot	$p_r = 0.0015$	$14cm^2$	$T_r = 4s$

and the same inspection state. The state granularity can be arbitrarily chosen in order to capture details of the robot's controller and environment which influence the swarm performance metrics (in our case, the time to completion and the energy consumption of the swarm). The advantage of having both modeling implementations for the same problem is particularly relevant in case of nonlinearities and discrete quantities in the chosen metrics (e.g., end criterion based on a finite number of inspected blades). All our models are time-discrete (time step  $T$ ), and characterized by two categories of parameters: state-to-state transition probabilities and behavioral delays. These parameters are computed and calibrated with the same method illustrated in [4], [5]. For the sake of completeness, we summarize in Table I all the values we adopted for our models (the same as in [4]).

The mean blade interaction time  $T_e$  is a function of  $T_{he}$ , and  $T_{max}$  and calculates for a blade with one tip to

$$T_e = -T_{he} + (I + 1)2T_{he}, \quad (1)$$

with  $I \geq \frac{T_{max}}{2T_{he}} - \frac{1}{2}$  and  $I \in \mathbb{N}$ , or  $I = \text{ceil}(\frac{T_{max}}{2T_{he}} - \frac{1}{2})$  respectively, where  $\text{ceil}(x)$  rounds to the next higher integer. Here,  $I$  reflects the maximal "allowed" tip encounters that are defined by  $T_{max}$ .

The probability  $p_b(k)$  to meet a beacon is instead a function of the number of robots acting as a beacon  $N_b(k)$  and the number of inspected blades  $M_i(k)$  at time-step  $k$  of the experiment:

$$p_b(k) = \frac{N_b(k)}{M_i(k)}, \quad M_i(k) \geq N_b(k) > 0 \quad (2)$$

assuming that there is always only one robot per blade that acts as a beacon and  $p_b(0) = 0$ . Note the difference of this probability and encountering probabilities; while encountering probabilities during search assume a spatially uniform distribution of the robots, i.e. the robots are hopping around in the arena, during inspection robots are following the blade contour. Thus  $p_b$  is the probability of encountering a beacon when the robot enters the possible beacon's signaling range, which is represented in the model by an average time  $T_b$ , the mean time spent on a blade before the beacon is met. This time  $T_b$  is a function of  $R_b$ , given by the signaling range of the beacon, and the time needed for collision avoidance with the beacon  $T_r$ ,

$$T_b = T_{he}(1 - R_b) + T_r \quad (3)$$

Note that a mean interaction time of  $T_b$  corresponds to robots leaving a blade with a constant rate of  $\frac{1}{T_b}$ . With  $M_v(k)$  being the number of virgin blades at time

$k$ , and with  $M_0 = M_v(k) + M_i(k)$ , we can derive the probabilities to encounter a virgin and an inspected blade respectively:

$$p_{virgin}(k) = p_e M_v(k) \quad (4)$$

$$p_{inspected}(k) = p_e M_i(k) \quad (5)$$

#### A. Mathematical Description of the Macroscopic Model

From Fig. 3, *right*, we can derive a set of difference equations (DE) to capture the dynamics of the whole system at the macroscopic level. We formulate one DE per considered state and exploit equations stating the conservation of the number of robots and the number of blades to replace one of the DEs.

Given  $M_0$  blades and  $N_0$  robots, the number of robots covering virgin and inspected blades  $N_v$  and  $N_i$ , the number of robots in obstacle avoidance  $N_a$ , the number of robots being a beacon  $N_b$ , and the number of robots in search mode  $N_s$  are given by (6)–(11) (compare also Fig. 2); the number of virgin blades  $M_v$  and the number of inspected blades  $M_i$  are calculated by (12)–(13):

$$N_v(k+1) = N_v(k) + \Delta_v(k) - \Delta_v(k - T_e) \quad (6)$$

$$N_i(k+1) = N_i(k) + \Delta_i(k) - \Delta_b(k) \quad (7)$$

$$- \Delta_i(k - T_e) \Gamma(k - T_e; k) \quad (8)$$

$$N_a(k+1) = N_a(k) + \Delta_r(k) + \Delta_w(k) \quad (9)$$

$$- \Delta_r(k - T_r) - \Delta_w(k - T_w)$$

$$N_b(k+1) = N_b(k) + \Delta_i(k - T_e) \Gamma(k - T_e; k) \quad (10)$$

$$+ \Delta_v(k - T_e) - \Delta_v(k - T_e - T_s)$$

$$- \Delta_i(k - T_e - T_s) \Gamma(k - T_e - T_s; k)$$

$$N_s(k+1) = N_0 - N_v(k+1) - N_i(k+1) \quad (11)$$

$$- N_a(k+1) - N_b(k+1)$$

$$M_v(k+1) = M_v(k) - \xi_e \Delta_v(k - T_e) \quad (12)$$

$$M_i(k+1) = M_0 - M_v(k+1) \quad (13)$$

with  $k$  representing the current time step (and absolute time  $kT$ );  $k = 0 \dots n$ ,  $n$  being the total number of iterations (and therefore  $nT$  the end time of the experiment). The  $\Delta$ -functions define the coupling between state variables of the model and can be calculated as follows:

$$\Delta_v(k) = p_{virgin}(k) N_s(k) \quad (14)$$

$$\Delta_i(k) = p_{inspected}(k) N_s(k) \quad (15)$$

$$\Delta_b(k) = \frac{T}{T_b} p_b(k) N_i(k) \quad (16)$$

$$\Delta_r(k) = p_r (N_0 - 1) N_s(k) \quad (17)$$

$$\Delta_w(k) = p_w N_s(k) \quad (18)$$

We can interpret the first DE (6) as follows. The average number of robots covering a virgin blade is decreased by those that finish inspection after time  $T_e$ ; it is increased by robots starting an inspection. The other state equations can be interpreted in a similar way.

Similar to the collaboration model described in [7], the  $\Gamma$ -function represents the fraction of robots that unavailingly waited for collaboration. In the stick-pulling

experiment [7],  $\Gamma$  represents the fraction of robots that did not find another robot to help with their task, pulling sticks out of the ground. Here instead,  $\Gamma$  expresses the fraction of robots that did not encounter a beacon before leaving a blade after  $T_e$ .

$$\Gamma(k - T_e; k) = \prod_{j=k-T_e}^k \left(1 - \frac{T}{T_b} p_b(k)\right) \quad (19)$$

In the above formulas,  $p_e$ ,  $p_r$ , and  $p_w$  represent the encountering probabilities of blades, robots, and walls respectively.  $T_e$ ,  $T_r$ , and  $T_w$  define the *average* time needed for circumnavigating a blade, avoiding a teammate, and avoiding a wall respectively.  $T_s$  is the time spent signaling as a beacon, and is a fixed model parameter. Hereby, we assume that the robot spends in average  $T_b$  before the beacon is encountered.

Due to the the possibility of leaving a blade before it has been completely covered due to  $T_{max}$ , we introduce a parameter  $\xi_e$  in (12), being the percentage of a blade a robot covers on average at each new interaction with it. Similarly to  $T_e$ ,  $\xi_e$  can be calculated from  $T_{he}$ ,  $T_{max}$ , for blades with one tip with  $I$  as above

$$\xi_e = \min\left(\frac{2I+1}{2}, 1\right), \quad (20)$$

with  $\min(x, y)$  being the minimum of  $x$  and  $y$ . The initial conditions are  $N_s(0) = N_0$  and  $N_a(0) = N_v(0) = N_i(0) = N_b(0) = 0$  for the robotic system (all robots in search mode) while those of the environmental system are  $M_v(0) = M_0$  and  $M_i(0) = 0$  (all blades virgin). As usual for time-delayed DE, we assume  $\Delta_x(k) = N_x(k) = M_x(k) = 0$  for  $k < 0$ .

### B. Swarm Performance Metrics

We consider two metrics for evaluating the performance of the swarm: time to completion  $nT$  and energy consumption of the swarm. The task is completed if all blades are inspected ( $M_v(n) = \epsilon$ ), with  $0 < \epsilon$  a certain degree of confidence. For calculating the energy consumption, we use the following model for the energy consumption of the individual agent. During every time step  $T$  that a robot is moving (Search, Avoidance, or Inspection modes, compare Fig. 2), it consumes  $\eta$  Watt on average, while it consumes  $\gamma$  Watts during every time step it is in beacon mode to account for idling and communication cost. Thus, the swarm energy used from the beginning of the experiment up to time step  $k$  can be approximated by

$$E(k, N_0, N_b) = ((N_0 - N_b^*)\eta + N_b^*\gamma)kT, \quad (21)$$

with  $N_0$  the number of robots and  $N_b^*$  the average number of robots acting as a beacon.

To compute the time to completion  $nT$ ,  $M_v(n) = 0$  is an easy condition to apply in the experiment and in the microscopic model. However, in the macroscopic model, this represents a limit condition as  $\lim_{k \rightarrow \infty} M_v(k) = 0$ , and thus we solve the macroscopic model for  $M_v(n) = \epsilon$ , with  $\epsilon$  a reasonable small value.

### C. Analysis of the Linear Macroscopic Model

According to (4), (12), and (14), we compute the number of virgin blades at time  $k$  as

$$M_v(k+1) = M_v(k) - \xi_e p_e M_v(k) N_s(k - T_e) \quad (22)$$

For  $T_s = 0$ , i.e. no beacons, the system reduces to a linear system, which can easily be seen by substituting  $T_s = 0$  into (10). As we will show below, the number of searching robots at time  $k$ ,  $N_s(k)|_{T_s=0}$ , i.e. no communication among the robots, has a steady state<sup>1</sup>  $N_s^*(T_e)$ , for which (22) simplifies to a linear, homogenous dynamical system. Its solution is given by

$$M_v(k) = (1 - \xi_e p_e N_s^*(T_e))^k M_0, \quad (23)$$

which will be exponentially decreasing to  $M_v^*(k) = 0$  if, and only if,

$$0 < 1 - \xi_e p_e N_s^*(T_e) < 1 \quad \text{and} \quad M_0 > 0. \quad (24)$$

Hence, we can solve (22) for arbitrary values of  $\epsilon$  with  $0 < \epsilon < M_0$  as a function of  $\epsilon$ ,  $\xi_e$  and  $T_e$  by approximating  $N_s(k)$  with its steady-state  $N_s^*$ , yielding

$$n(\epsilon, \xi_e, T_e) = \frac{\ln \frac{\epsilon}{M_0}}{\ln(1 - \xi_e p_e N_s^*(T_e))} + T_e, \quad (25)$$

the number of time steps for covering  $M_0 - \epsilon$  blades. Note, that we add  $T_e$  to reflect the fact that (22) is a delayed state, i.e. a blade is assumed to be covered only after  $T_e$ .

### D. Analysis of the Non-Linear Macroscopic Model

For  $T_s > 0$ , the model from (22) becomes non-linear. Transforming the equation to a simple linear dynamical system becomes less straightforward, as it is difficult to determine steady states for  $N_s(k)$  and  $N_b(k)$ . However, similar as in [7], we can approximate  $N_b^*$  in the time domain by integrating the inflow of robots into the beacon state during  $T_s$ , as every robot covering a beacon-less blade will become a beacon at some point. Note however, that we do not reduce the probability to hit an inspected blade by the number of beacons as this would yield additional non-linearity.

$$N_b^* = \sum_k^{k+T_s} \tilde{N}_s^* p_e M_0 \left(1 - \frac{N_b^*}{M_0}\right) = \frac{\tilde{N}_s^* p_e M_0 T_s}{1 + \tilde{N}_s^* p_e T_s}. \quad (26)$$

Similar we get for  $N_a^*$ ,  $N_i^*$ ,  $N_v^*$ , and  $\Gamma^*$

$$N_a^* = p_r (N_0 - 1) N_s^* T_r + p_w N_s^* T_w \quad (27)$$

$$N_i^* = N_s^* p_e M_0 (T_b + \Gamma^*(T_e - T_b)) \quad (28)$$

$$N_v^* = 0 \quad (29)$$

$$\Gamma^* = \left(1 - \frac{N_b^*}{T_b M_0}\right)^{T_e} \approx \left(1 - \frac{T_e N_b^*}{T_b M_0}\right), \quad (30)$$

where we approximate  $\Gamma^*$  with its first-order binomial expansion as  $\frac{N_b^*}{T_b M_0} \ll 1$ . Hence,  $\tilde{N}_s^*$  can be calculated by

$$\tilde{N}_s^* = N_0 - N_b^* - N_i^* - N_a^*, \quad (31)$$

<sup>1</sup> $N_s^* = \lim_{k \rightarrow \infty} N_s(k)$

as the number of robots is constant. For clarity of presentation, we solve (31) assuming  $T_r$  and  $T_w$  to be much smaller than  $T_s$  and  $T_e$ , i.e.  $N_a^* = 0$ , yielding

$$\tilde{N}_{s1/2}^* = \frac{T_b + M_0 p_e T_b (T_e + T_s) - N_0 p_e T_b T_s \pm \Xi}{2 p_e T_s (M_0 p_e T_e (T_e - 2 T_b))}, \quad (32)$$

with  $\Xi$  given by

$$\Xi = \frac{T_b((1 - p_e(N_0 T_s + M_0(T_e + T_s)))^2 - 4 N_0 p_e T_s (M_0 p_e T_e (T_e - 2 T_b) - T_b))}{(33)}$$

where only one value will be larger than zero,  $\tilde{N}_s^*$  in the remainder. Note however, that the solution for (31) is only valid for  $T_s > 0$  while it is equal to

$$N_s^*(T_e) = \frac{N_0}{1 + p_e M_0 T_e + p_r (N_0 - 1) T_r + p_w T_w} \quad (34)$$

for  $T_s = 0$ . As the system is linear for  $T_s = 0$ ,  $N_s^*$  could be equally well calculated using the Z-transform's finite value theorem, as shown in [7].

With the existence of the steady states from (26)–(31), equation 22 simplifies to

$$\tilde{M}_v(k+1) = \tilde{M}_v(k) - \tilde{M}_v(k) \xi_e p_e \tilde{N}_s^*, \quad (35)$$

which is again a linear dynamical system. Its solution is

$$\tilde{n}(\epsilon, \xi_e, T_e) = \frac{\ln \frac{\epsilon}{M_0}}{\ln(1 - \xi_e p_e \tilde{N}_s^*)} + T_e, \quad (36)$$

where we again add  $T_e$  to account for the delay state.

#### IV. RESULTS

In order to validate our analysis on the macroscopic model, we simulated the PFSM from Fig. 3 for 20 robots, 16 blades, and the encountering rates given in Table I. We plot the time to completion for  $\epsilon = 0.5$  versus  $T_{max}$  according to (25) and compare it with the prediction of the microscopic model for the linear model (Fig. 4, *left*) and for the non-linear model (Fig. 4, *right*).

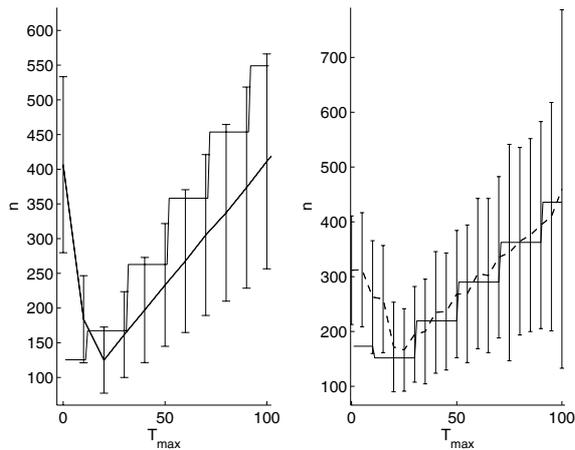


Fig. 4. Microscopic (10000 runs for each value of  $T_{max}$ ) vs. Macroscopic simulation. Microscopic results are plotted with error bars (standard deviation). *Left*: Linear model,  $T_s = 0$ . *Right*: Nonlinear model,  $T_s = 20$ .

We also examine time to completion and energy consumption for different values of  $T_s$ , and compare it with the microscopic model. Time to completion is normalized with  $\tilde{n}^{opt} = T_{fe}$  as the best possible outcome, while energy is normalized with  $\tilde{n}^{opt} \eta N_0$ , i.e. the least possible energy consumed by the swarm. For simplicity, we assume  $\eta = 1W$  and  $\gamma = 0$ , i.e. no energy consumption while the robot is in the beacon state. Results for both metrics are plotted in Fig. 5. Also, we examine the influence of the parameter  $R_b$  that is given by the quality of marking of the beacon on our metrics by plotting the performance vs. the effective  $T_b$  (Fig. 6); an optimal beacon would mark the blade as a whole, leading to  $T_b = T_r$ , that is blades guarded by a beacon would cost only the time of a simple collision. In the worst case, the robot circumnavigates half a blade in average ( $T_{he} + T_r$ ).

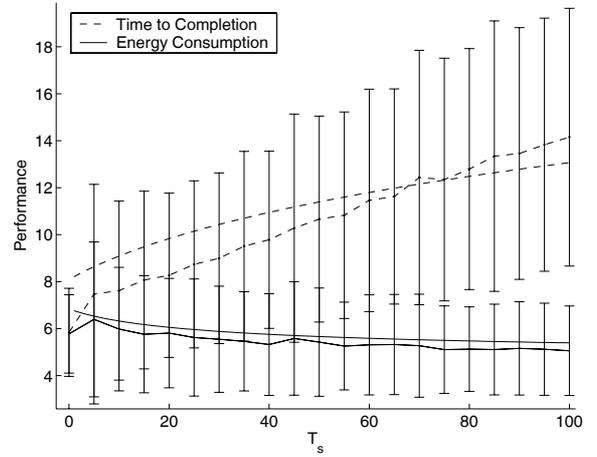


Fig. 5. Prediction of microscopic and macroscopic model for time to completion and energy consumption for 16 blades ( $T_{max} = 20$ ,  $N_0 = 20$ ,  $\eta = 1$ ,  $\gamma = 0$ ) for different values of  $T_s > 0$ .

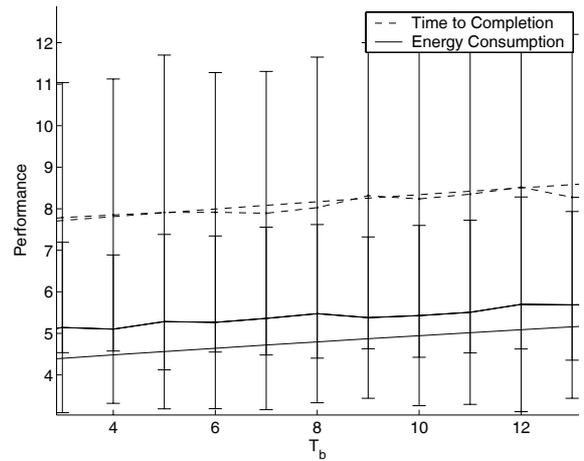


Fig. 6. Prediction of microscopic and macroscopic model for time to completion and energy consumption for 16 blades ( $T_{max} = 20$ ,  $N_0 = 20$ ,  $\eta = 1$ ,  $\gamma = 0$ ) for different values of  $T_r \leq T_b \leq T_{he} + T_r$ .

## V. DISCUSSION

From Fig. 5 one can clearly see that time to completion increases with  $T_s$  for the chosen parameters. On the other hand, the average energy consumed by a team member is decreasing. As the smaller energy consumption is easily explained by robots consuming less energy being a beacon, increased time to completion suggests that marking inspected blades with robotic beacons, is not very effective—a robot does better continuing to search rather than serving as a beacon, as long energy consumption does not matter. On the other hand, the effect of collaboration might be improved by increasing the signaling range of the robot and thus decreasing  $T_b$ , the average time the robot travels along the blade until the beacon at its tip is met. By this, the benefit of meeting a beacon increases, as robots quit already inspected blades earlier and hence reduce redundant coverage. However, in reality increasing the signaling range leads to a quadratic increase of  $\gamma$ , the energy consumed during signaling, a fact that is not represented in the above plot.

While the microscopic model allows us to capture some details more accurately, such as the fact that a robot might cover a part of a blade that has been already covered by another robot and thus does not contribute in decreasing the number of virgin blades, in the macroscopic model we instead assume constant coverage ( $\xi_e$ ) for each interaction. This effect can be observed in Fig. 4, where the macroscopic model predicts a slightly smaller value for the optimum  $T_{max}$  than the microscopic model does. This is because a  $T_{max} < T_{fe}$  would lead to partial coverage that would be necessarily covered by the next visit in the macroscopic model. In the microscopic model instead it is very likely that the next robot would—more faithful to reality—cover big parts of the already inspected part of the blade.

Contrary to solving the macroscopic difference equations numerically which leads to good agreement with the microscopic model [4], [5], steady state analysis introduces an approximation that *might* lead to wrong conclusions. For instance, for high values of  $T_s$  steady state analysis yields better performance as the microscopic model (Fig. 5), as steady state analysis considers conditions at the end of the experiment where all blades are already covered. Thus, the steady state estimate for the number of beacons is higher than the effective number of beacons during the experiment. Also, discontinuities that arise in the transition from linear to non-linear analysis (compare eq. 34 and eq. 32) merit special attention, as they eventually lead to unfaithful prediction in the vicinity of the discontinuity (compare Fig. 5 for small values of  $T_s$ ).

## VI. CONCLUSION

We presented an extension to our swarm-intelligent inspection system [4], [5], that makes use of explicit communication in order to minimize the energy consumption of the swarm. Due to the simplicity of the chosen means of communication, time to task completion increases with the

amount of time robots spend as beacons, as those robots are missing for search and inspection during the signaling period. On the other hand, the beacon based approach leads to considerable savings in swarm energy (and therefore average individual energy consumption), a crucial property when designing the individual swarm member.

We conclude that in order to speed up the inspection process, blades would need to be marked independently of the presence of another robot. This could be either achieved by permanently marking the inspected blade, for instance by an (evaporating) chemical, or by enabling the robots to keep track of task progress using some form of internal representation of the environment, for instance a topological map.

A beacon-based system becomes more complex to analyze, as communication amongst the robots leads to non-linear difference equations on the macroscopic level. Doing a steady state analysis of both the linear and non-linear system let us to transform our non-linear metric, the time to task completion, into a linear dynamical system, which allows for getting quantitative insight into the task performance and shows good agreement with microscopic modeling that in turn showed good agreement with embodied simulations and real robotic platforms. In comparison to the microscopic model, predictions can be achieved in considerably less time (1000 or more simulations for every parameter set versus calculating the value of a second order logarithmic function). Furthermore, the macroscopic model allows analytical insight as the performance becomes mathematically tractable.

Finally, we showed that special care needs to be taken during the steady state analysis to preserve the delay properties of the difference equations in the approximated solution.

## REFERENCES

- [1] W. Agassounon, A. Martinoli, and K. Easton. Macroscopic modeling of aggregation experiments using embodied agents in teams of constant and time-varying sizes. *Autonomous Robots*, 17(2-3):163–192, 2004. Special issue on Swarm Robotics, M. Dorigo and E. Sahin, editors.
- [2] R. Arkin. *Behavior-Based Robotics*. The MIT press, Cambridge, MA, USA, 2000.
- [3] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. SFI Studies in the Science of Complexity, Oxford University Press, New York, NY, USA, 1999.
- [4] N. Correll and A. Martinoli. Collective inspection of regular structures using a swarm of miniature robots. In *Proceedings of the 9th International Symposium of Experimental Robotics (ISER)*. Singapore, June 2004. Springer Tracts for Advanced Robotics (STAR), 2005. In press.
- [5] N. Correll and A. Martinoli. Modeling and optimization of a swarm-intelligent inspection system. In *Proceedings of the 7th Symposium on Distributed Autonomous Robotic System (DARS)*. Toulouse, June 2004. Distributed Autonomous Systems 6, 2005. In press.
- [6] K. Martin and C.V. Stewart. Real time tracking of borescope tip pose. *Image and Vision Computing*, 10(18):795–804, July 2000.
- [7] A. Martinoli, K. Easton, and W. Agassounon. Modeling of swarm robotic systems: A case study in collaborative distributed manipulation. *Int. Journal of Robotics Research*, 23(4):415–436, 2004. Special issue on Experimental Robotics, P. Dario and B. Siciliano, editors. Invited paper.
- [8] O. Michel. Webots: Professional mobile robot simulation. *Journal of Advanced Robotic Systems*, 1(1):39–42, 2004.