2011 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 25-30, 2011. San Francisco, CA, USA

Communication assisted navigation in robotic swarms: self-organization and cooperation

Frederick Ducatelle*, Gianni A. Di Caro*, Carlo Pinciroli**, Francesco Mondada*** and Luca Gambardella*

***Abstract*— We present a communication based navigation algorithm for robotic swarms. It lets robots guide each other's navigation by exchanging messages containing navigation information through the wireless network formed among the swarm. We study the use of this algorithm in two different scenarios. In the first scenario, the swarm guides a single robot to a target, while in the second, all robots of the swarm navigate back and forth between two targets. In both cases, the algorithm provides efficient navigation, while being robust to failures of robots in the swarm. Moreover, we show that in the latter case, the system lets the swarm self-organize into a robust dynamic structure. This self-organization further improves navigation efficiency, and is able to find shortest paths in cluttered environments. We test our system both in simulation and on real robots.**

I. Introduction

Swarm robotics studies systems consisting of large groups of relatively simple robots that interact and cooperate with each other in order to jointly solve tasks that are outside their own individual capabilities [1]. Such cooperative task solving often relies on self-organization and emergence, where self-organization refers to the fact that the swarm's organization comes from within the system (i.e., is not imposed from outside), and emergence means that the organization comes about in a decentralized way, from local interactions between individual robots [2].

Many studies in the area of swarm robotics address a collective navigation problem, where robots need to move back and forth between two locations, e.g. to transport items from one place to another [3], [4], [5], [6], [7], [8]. Most of this work is inspired by the foraging behavior of certain types of ants, which relies on stigmergic communication through pheromone trails: each ant moving between the nest and a food source leaves pheromone in the environment, which attracts other ants and guides them to the food. The interesting aspect is that the collective process of pheromone laying and following reinforces the most efficient paths, so that eventually the shortest path emerges as a consequence of the swarm's collective actions.

In this paper, we take a different approach to this swarm navigation task, based on network communication. We show that also this new approach gives rise to emergent behavior that lets the swarm navigate over efficient paths. We first focus on the *navigation of a single robot* to a target location, guided by the other robots of the swarm, and after that study the *collective navigation* of all robots between two targets.

The problem setup for the navigation of a single robot is as follows. The swarm is deployed in a confined area. A robot T of the swarm is assumed to have found a target location (e.g., a location where an event needs to be served, or from where food needs to be transported). It remains static at that location, and announces its presence through periodic messages. We refer to T as a target robot. A robot S needs to navigate to T (e.g., it has specific skills to service the event). All other robots of the swarm are assumed to be involved in tasks of their own, and they do not alter their movements to help S in its navigation task. They do, however, offer support through communication, by forwarding the messages from T over the network formed between the robots. S makes use of these messages to navigate to T, by using network routing information for robot navigation.

For the collective navigation task, two robots T and T' indicate two different target locations (e.g., a nest and a food source), and all other robots navigate back and forth between T and T'. They use the same communication based navigation algorithm described above. We show how the concurrent execution of this behavior by all robots lets the swarm self-organize, and how a collective dynamic structure emerges that supports swarm navigation.

Throughout the paper, we show that our approach leads to *efficient navigation* and finds *shortest paths* in cluttered environments. Moreover, it is *robust*, e.g. with respect to robot failures. Compared to existing pheromone based navigation systems, it avoids the practical problem of how to implement stigmergic communication (see also section VII).

Our navigation system relies on message communication between robots to detect obstacle free paths. To make this possible, we need a wireless communication device that provides only line-of-sight communication, so that communication links can be related to navigable paths. Moreover, we need a device that can link received messages to relative position information (angle and distance) about their sender, so that robots can follow paths detected through communication. We found such device in the form of an infrared range-and-bearing (IrRB) communication system [9], of which im-

This work was partially supported by the SWARMANOID project, funded by the Future and Emerging Technologies programme (IST-FET) of the European Commission under grant IST-022888. The information provided is the sole responsibility of the authors and does not reflect the European Commission's opinion. The European Commission is not responsible for any use that might be made of data appearing in this publication.

This work was also supported by the Swiss National Science Foundation through the National Centre of Competence in Research Robotics.

* F. Ducatelle, G.A. Di Caro and L.M. Gambardella are with IDSIA, USI/SUPSI, Galleria 2, 6928 Manno-Lugano, Switzerland. E-mail: {frederick,gianni,luca}@idsia.ch. F. Ducatelle is corresponding author.

** C. Pinciroli is with IRIDIA, CoDE, ULB, Avenue F. Roosevelt 50, 1050 Brussels, Belgium. E-mail: cpinciro@ulb.ac.be.

*** F. Mondada is with LSRO, EPFL, ME A3 484 (Bâtiment ME), Station 9, 1015 Lausanne, Switzerland. E-mail: francesco.mondada@epfl.ch.

plementations exist for various robots [10], [11], [12]. While most results presented in this paper were obtained through simulation, we present in section VI an implementation of our algorithm on real robots, using the IrRB system.

The rest of this paper is organized as follows. In Section II, we describe the communication aided navigation algorithm. In section III, we evaluate the performance of this algorithm in a scenario where a single robot navigates to a single target. After that, we study the behavior of the system when all robots of the swarm navigate back and forth between two target locations: in section IV we show how the swarm self-organizes and cooperates to get more efficient navigation, and in section V we show that the swarm is able to find shortest paths in cluttered environments. After that, in section VI we describe the implementation of our system on real robots, and in section VII we discuss related work.

II. Communication aided navigation

The navigation system we propose is loosely based on routing algorithms used in mobile ad hoc networks (MANETs). The general idea is that all robots in the swarm maintain a table with navigation information about all known target robots in the environment, similar to how nodes in a MANET maintain routing tables. Each robot periodically broadcasts the content of this table to its neighbors, so that the information spreads throughout the swarm. To navigate to a given target robot T, a searching robot S follows received navigation information, similar to how data packets follow routing information. In what follows, we describe different aspects of this system in detail. We also point out that an older version of this algorithm together with some preliminary results for the navigation of a single robot to a target were described in [13].

The navigation information about a target T present in a robot A's navigation table consists of a sequence number $s(T)$, indicating the relative age of the information, and a distance $d(A,T)$, indicating the distance traveled by the information between T and A. The total size of a robot's navigation table is thus a linear function of the number of targets it knows about. At the start of swarm deployment, all robots have an empty table. When a robot T becomes a target robot (i.e., it discovers a target location and starts announcing it), it puts an entry about itself in its table. In this entry, both the sequence number $s(T)$ and the distance $d(T,T)$ are set to 0. At periodic intervals, robots broadcast the content of their table to neighbors. The size of a robot's broadcast messages equals the size of its navigation table; if bandwidth is limited, robots send updates for a subset of known targets, in a round-robin fashion. When T broadcasts the information about itself, it first increases sequence number $s(T)$ in its table by 1. The distance $d(T,T)$ is broadcast without modification. Another robot A broadcasting information about T does not modify $s(T)$, so that the sequence number marks the relative time when the information left T.

A robot B receiving a broadcast from A processes the entries for all targets T in the message, reading the received sequence number $s'(T)$ and distance $d'(A,T)$. On the basis of $d'(A,T)$, it calculates a new estimate for its own distance to T, $d'(B,T)$, by adding the distance $d(B,A)$ between itself and A (as measured at message reception with the IrRB communication system). Like this, $d'(B,T)$ reflects the distance traveled by the information from T to B. This is an upper bound of the shortest obstacle free path for robot navigation from B to T, since the communication works only in line-of-sight. Then, B compares the new values, $s'(T)$ and $d'(B,T)$, to the information about T in its own table, $s(T)$ and $d(B,T)$. The new information is considered better if either $s'(T) > s(T)$ (the new information is more recent), or $s'(T) = s(T)$ and $d'(B,T) < d(B,T)$ (the new information indicates a shorter path). In that case, the information in the table is replaced by the new information.

If B moves around without receiving new updates about T for a while, the distance $d(B,T)$ in its table can quickly loose its value as an estimate of the shortest obstacle free path between B and T. Therefore, as B is moving, it measures its moved distance through odometry, and adds this to $d(B,T)$. This way, $d(B,T)$ grows and remains a measure of the distance traveled by the navigation information. Note that the direction of B's movement is not taken into account, so that $d(B,T)$ is not necessarily the shortest distance to T. But it is an upper bound of the shortest obstacle-free path (since B per definition moved over an obstacle-free path). Using this mechanism, the navigation system can work in sparsely connected swarms: navigation information can bridge gaps in network connectivity by traveling on board of moving robots (as is common in the area of delay tolerant networks).

When a searching robot S receives information about its target T from a robot A, it stores $s(T)$ and $d(A,T)$, as well as the relative position of A at the moment the information was received (as measured through the IrRB system). Using odometry, it goes to this position. If S receives new information about T from a robot B, $s'(T)$ and $d'(B,T)$, it compares this to the old information from A, and it starts to move towards B's location if $s'(T) > s(T)$ (the new information is more recent), or if $s'(T) = s(T)$ and $d'(B,T) < d(A,T)$ (B is closer to T than A). If S reaches the position it is moving to without getting new information about T, it can either wait there, or start doing random movements until new information is received. We refer to the former strategy as "navigation with stopping", and to the latter as "navigation with random"; we investigate both strategies in section III. These strategies also define the behavior of S in case it gets disconnected from the swarm. When S receives a message directly from T, it goes straight to T. Finally, we point out that we let searchers approach locations they move to from the right, so that two searchers moving towards each other do not collide.

III. Navigation of a single robot

In this section, we investigate the working of the navigation algorithm when a single robot needs to find a single target. The other robots of the swarm perform random movements. The goal is to show to what extent the robots of the swarm can give support for each other's navigation without adapting their movements for this.

All tests presented here and in the next two sections are executed using a simulated model of the foot-bot, a small ground robot developed within the project Swarmanoid (`http://www.swarmanoid.org`) on the basis of the marXbot platform [12]. It has a diameter of 15 cm and is 20 cm high. Its IrRB system has a capacity of one message of 10 bytes per robot at each timestep of 0.1 s (so robots can broadcast an update every 0.1 s). Its maximum range is limited to 3 m here. We use the ARGoS simulator [14], which contains a reliable physics model of the foot-bot.

In a first set of tests we use an uncluttered closed arena of 20x20 m^2. The robots are placed in the arena according to a uniform random distribution. One of the robots is a target and remains static. A second robot searches this target. The remaining robots move according to a random direction model: choose a direction θ uniformly from $]-\pi, \pi]$, turn towards θ, choose a time t from an exponential distribution with fixed average (set to 10 s here), move forward for this time t, and then repeat this process. We use a forward speed of 0.15 m/s, both for the searching and the randomly moving robots. All robots have an obstacle avoidance mechanism, based on short range infrared proximity sensors, which makes them turn away from each other and from walls. We vary the number of robots in the swarm, from 2 (0 randomly moving robots) up to 92 (90 randomly moving). For each data point, we make 500 independent test runs (this high number is needed because the random initial positions of searcher and target induce a high variance). We measure the time between the start of each test and the moment the searching robot comes in range of the target.

The results are shown in figure 1. We compare the two variations of the navigation system presented in section II, "navigation with stopping" and "navigation with random", which differ in the strategy used by the searching robot when it does not have any navigation information. The results show a large difference in performance between both strategies for low numbers of robots. This is because the communication network is sparse, and navigation information spreads slowly from the target. In the extreme case with 0 randomly moving robots, navigation with stopping can never reach the target. Navigation with random, on the other hand, does find the target, through random search. The expected time for a randomly moving agent to find a static target is normally referred to as the "hitting time" [15]. In the context of DTNs, analytical models have been developed to calculate the hitting time for various mobility models [15]. In our case, those models can be used to calculate an upper bound for the expected time needed to reach the target when using navigation with random. For high numbers of robots, the difference between both strategies decreases: the network gets better connected, and the searcher rarely falls without information. For the highest numbers of robots, performance gets close to the time needed to cross the expected straight line distance between the searcher's initial position and the target. This is indicated in figure 1 as "Delay navigation shortest path". This gives a lower bound for the expected navigation time. It is interesting to note the graceful degradation of the system's performance as the number of robots goes down. This indicates that the navigation system is robust with respect to failure or loss of robots in the swarm.

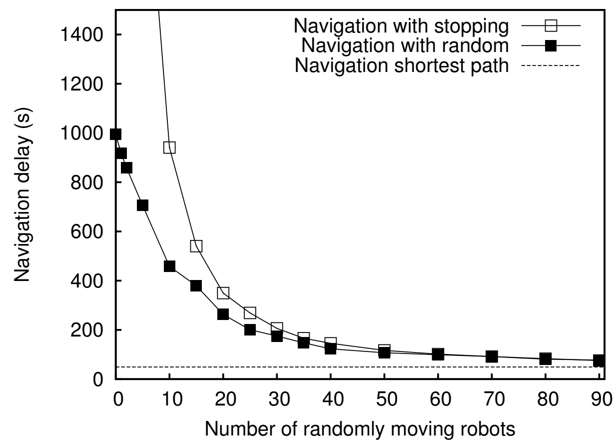

Fig. 1. Experimental results in an uncluttered environment. See main text for explanation.

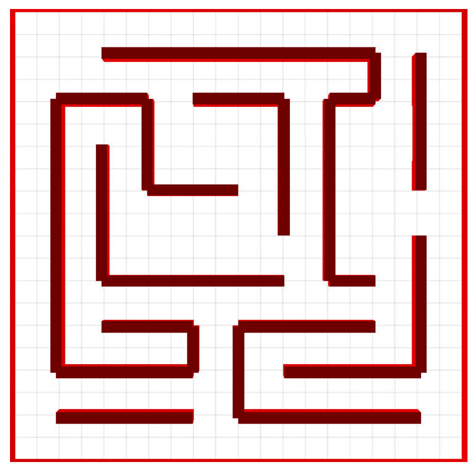

Fig. 2. Layout of the maze for our experiments. The area is 20x20 m^2.

In a second set of experiments, we test the system in a cluttered environment. Since the algorithm looks for obstacle free paths (see section II), it should be able to deal with such situations. We use again an arena of 20x20 m^2, in which now obstacles are placed to form a maze, as shown in figure 2. Again, we deploy the swarm according to a uniform random distribution, and we measure the time needed for the searcher to reach the target. The results are shown in figure 3. The searcher needs a lot more time to reach the target. Also, a larger swarm is needed to bring this delay down, and the system has more difficulties to reach the time required to travel over the shortest path. Nevertheless, we get the same trends in performance, and with a large enough swarm, the system guides a searching robot to its target efficiently.

Clearly, the performance of the system depends also on the movement patterns of the robots of the swarm. We

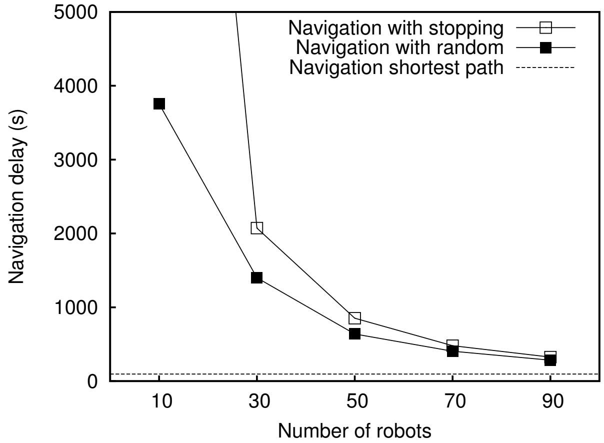

Fig. 3. Experimental results in the maze environment of figure 2. See main text for explanation.

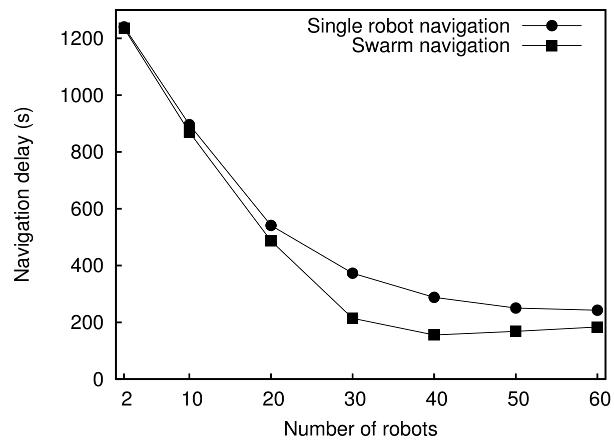

Fig. 5. Experimental results for swarm navigation. See main text for details.

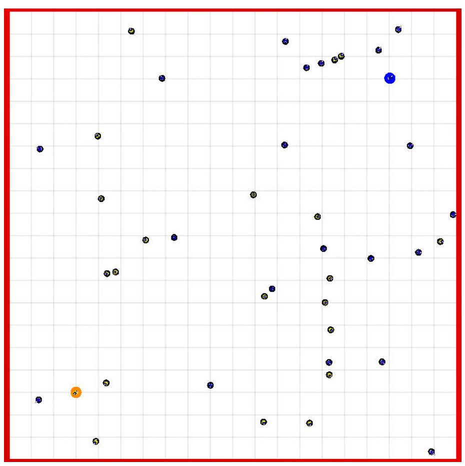

Fig. 4. Setup for swarm navigation experiments. The area is 20x20 m^2. The target robots are located in the top-right and bottom-left corner (indicated with large orange and blue disks).

have tested other random mobility patterns, such as random waypoint [16], with similar results (not shown here for lack of space) as those with random direction. In future work, we plan to use also other movement patterns, e.g., to simulate the movement of robots in a warehouse or along production lines. We do not investigate this further here though, as we want to concentrate on the effects of self-organization and cooperation, presented in the remaining of this paper.

IV. Swarm navigation

We study how a swarm of robots can use the communication based navigation algorithm to move back and forth between two target locations. As pointed out in section I, this is a common task in swarm robotics. To follow swarm terminology, we refer to the two target locations as nest and food source. We use only the "navigation with random" strategy, as this gives the best performance. We show how local behavior based on our navigation algorithm lets the swarm self-organize and show coordinated global behavior.

We first investigate the swarm's behavior in an uncluttered arena, as shown in figure 4. Two robots, indicating nest and food source, are placed in opposite corners, at about 20 m from each other. All other robots are placed according to a uniform random distribution. Half of these robots initially go to the food source, the other half to the nest. A robot that has reached its target (i.e., food source or nest) starts moving towards the other target. A robot is said to have reached a target when it comes within 0.5 m of it. We vary the total number of searching robots in the swarm from 2 up to 60. We perform 50 independent tests of 5000 s for each setup. We measure the average time needed by robots to move from one target to the other. We compare to experiments with the same numbers of robots, but where only one robot is going back and forth between nest and food source, while the other robots are moving according to the random direction mobility model (as in the experiments of section III).

The results are shown in figure 5. The scenario where all robots navigate back and forth is referred to as "swarm navigation", and the other as "single robot navigation". In both scenarios, performance improves as the number of robots increases, since navigation information spreads more easily in densely connected swarms However, for the swarm navigation scenario, the performance improves faster (with 30 robots, navigation delay of swarm navigation is about half of that of single robot navigation). This is due to cooperation. When a robot moving towards the food source meets a robot navigating towards the nest, they can give each other navigation information about their respective targets. Moreover, if a group of robots moving towards the same target are in communication range from each other, new information received by any of them spreads throughout the group, and they simultaneously move in the same direction. These two effects make robots form clusters moving in opposite directions. When there are enough robots, such clusters can become large enough to cover the whole distance between nest and food source. At that point, the swarm organizes into a stable structure, which we refer to as a *dynamic chain*. Figure 6 shows a snapshot after 300 s of a typical run of

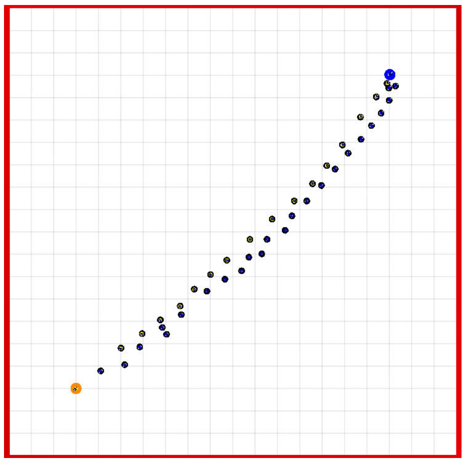

Fig. 6. Swarm navigation after 300 s of simulation: a self-organized dynamic structure has formed.

swarm navigation with 40 robots, illustrating this behavior. It is this behavior which causes the strong improvement in performance between 20 and 30 robots in figure 5. For larger swarms (50 and 60 robots), congestion of robots near target locations leads to a decrease in performance.

The dynamic chain is an example of emergent self-organized behavior: the swarm shows organization at the global level that emerges from local interactions between individual robots. In what follows, we investigate when this self-organization arises and how stable it is. To do this, we first need a measure for self-organization. Several authors use entropy to measure self-organization in the context of swarm robotics [17], [18]. If X is a random variable which can take M different states, its entropy $H(X)$ is defined as

$$H(X) = -\sum_{i=1}^{M} p_i \log_2(p_i), \tag{1}$$

where p_i is the probability that X is in state i (here, we refer to Shannon's information entropy [19]). Strictly speaking, this is a measure for order (or disorder), rather than self-organization: the more a system is ordered, the more you can find it in a limited subset of its possible states, and the lower the entropy. Some authors criticize measuring self-organization as a mere increase in order, and propose other measures [20]. For us, however, it is sufficient to measure whether there is increased order in the behavior of the robots, so we stick with entropy.

To calculate the entropy $H(X)$, we need a discrete variable X that characterizes the swarm's behavior. In [21], [18] the authors use the orientation of the robots, discretized into four bins; the entropy based on this variable indicates to what extent the robots face the same direction. In our case, this measure can be used (once the chain is formed, robots face in similar directions), but it is quite noisy, especially when there is congestion (robots turn to avoid each other). What we really want to measure is whether the robots move in a low number of connected clusters; whether there is order in

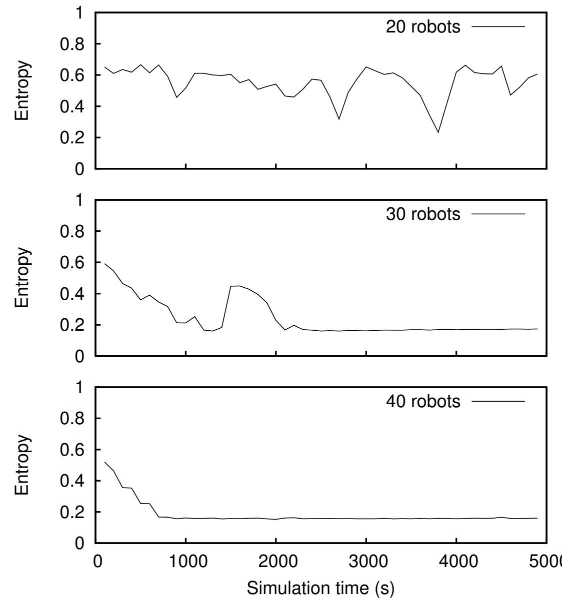

Fig. 7. Evolution of $S(R)$ over the course of an example test run for 20, 30 and 40 robots

their physical locations. To do this, we turn to hierarchic social entropy [22], which proposes an entropy measure for a group of robots characterized by a multi-dimensional variable. In our case, this multi-dimensional variable will be the location coordinates of each robot. The idea behind hierarchic social entropy is to first cluster the robots using hierarchic clustering based on a distance threshold h: a robot is added to a cluster if it is within distance h from all robots in the cluster. The division of robots into clusters gives a discrete variable X on the basis of which entropy is calculated (the clusters form the M different states for X, and the probabilities p_i are defined by the number of robots in each cluster). Obviously, X depends on the threshold h: if $h = 0$, each robot is in a cluster of its own, and entropy is maximal, while if $h = \infty$, all robots fall in a single cluster, and entropy is 0. Therefore, the notation $H(R, h)$ is used to refer to the entropy of a group of robots R using clustering distance h. The hierarchic social entropy $S(R)$ is then defined by integrating $H(R, h)$ over all values of h:

$$S(R) = \int_{0}^{\infty} H(R, h) dh. \tag{2}$$

We use $S(R)$ based on the location coordinates of the robots to analyze the behavior of the swarm. Compared to the definition of S(R) in [22], we introduce one change, related to the clustering: we use single linkage clustering, which means that a robot is added to a cluster if it is within distance h from any robot of that cluster. Single linkage clustering can find long stretched clusters [23], which enables it to detect the chaining behavior of the swarm. In figure 7, we show the evolution of $S(R)$ over the course of example test runs with 20, 30 and 40 robots; we calculate $S(R)$ at every timestep of 0.1 s, and average it per 100 s of simulation. When the

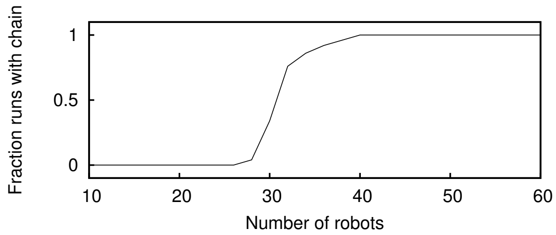

Fig. 8. Fraction of test runs in which a stable dynamic chain forms.

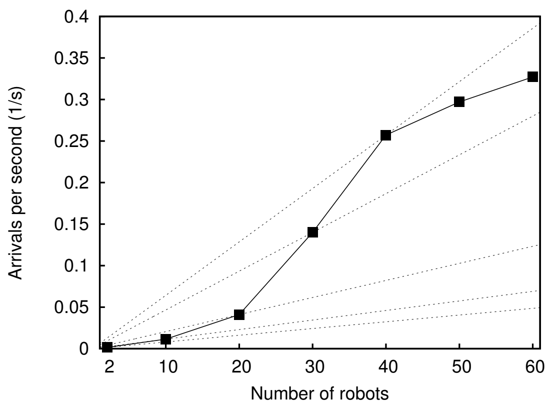

Fig. 9. Frequency with which the targets are reached by robots.

robots of the swarm move close together, there is a drop in entropy. When the dynamic chain forms, entropy stays low for an extended amount of time. All runs with 20 and 40 robots display patterns similar to the ones shown here: for 20 robots, the chain never forms, while for 40 robots it forms quickly and remains for the whole duration of the simulation. With 30 robots, varying patterns have been observed. In some runs, including the example here, the chain forms after a while. In other runs, it does not form. Interestingly, when it does form, it usually stays for the whole test duration. This suggests that the chain is stable with respect to perturbations.

In figure 8, we study the stability of the chain. For increasing numbers of robots, we perform each time 50 test runs, and measure in which fraction of those runs a stable dynamic chain appears. We consider the chain stable if for the last 1000 s of the test $S(R)$ remains below 0.2. The graph shows a clear phase transition around 30 robots: with less robots, the system never self-organizes, with more it always does. Such phase transitions are typical for self-organizing systems in physics, and have also been observed in swarm robotics [21]. They indicate that within a given range of a control parameter, the self-organizing behavior is robust and takes place independently from perturbations in the system.

Finally, in figure 9, we show how frequently the targets are reached by robots. This indicates how many items the swarm could transport between the two locations. Increasing the swarm size, one could expect a sub-linear performance improvement, because more robots can transport proportionally more items (linear improvement), but there is also increased congestion. In our system, increased swarm size also gives more cooperation, which leads to a super-linear increase in performance between 10 and 40 robots (dotted lines in the figure illustrate for each swarm size the extrapolated performance in case of linear improvement). For more robots, congestion makes the performance growth decrease.

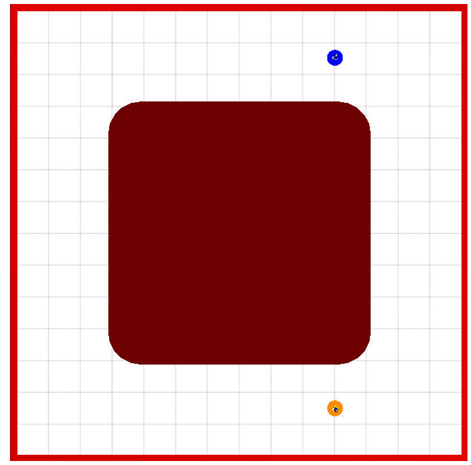

Fig. 10. Test setup for multi-path environment. The area is 14x14 m^2.

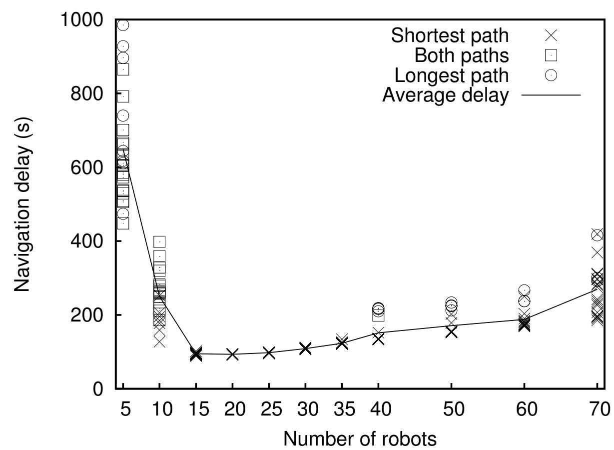

Fig. 11. Navigation delay versus number of robots in the scenario of figure 10. The choice of path in each test is shown by the point symbols.

V. Shortest Path Finding

An interesting question is what happens when there are multiple paths between the targets, e.g. due to the presence of obstacles. We study the setup of figure 10, where there is a short path of length $d_s = 15$ and a long path of length $d_l = 27$. We vary the swarm size from 5 to 70 robots, and perform 25 tests of 5000 s for each size. We measure the average time needed for a robot to navigate between the targets. We also observe at each time step how many robots are located on the short path versus on the long path, and combine this per test to calculate the percentage of robots using the short path, p_s. If $p_s > 66\%$, we say the swarm uses the short path, if $p_s < 33\%$ it uses the long path, and otherwise it uses both.

Figure 11 shows the result of each individual test, as well as the average per swarm size. For swarm size 5, the robots use both paths, with a slight perference for the long path. This is because navigation information is scarce, and robots mainly move randomly, leading to a uniform distribution; since there is more space on the long path, we find more robots there. Starting from 10 robots, there is a preference for the short path, which becomes very strong from 15 robots. In general, using the short path leads to lower delay, except for 70 robots, where congestion plays a major role.

The preference for the short path is explained as follows. Robots go towards navigation information with high sequence number and short distance (see section II). If we assume a homogeneous initial distribution of robots, information travels equally fast around both sides of the obstacle, and the same number of robots are attracted towards the left and right of the obstacle: $0.5(d_s + d_l)\rho$, where ρ is the robot density in robots/m. Both paths receive the same number of robots, but on the short path this leads to a higher density. This means that information can spread faster there, giving more chances to attract robots.

Starting from 15 robots, also the dynamic chain plays a role. It makes the swarm navigation lock onto one of the paths, so that we rarely observe the use of both paths. Between 15 and 30 robots, there are enough robots to form the chain over the short path, but not over the long path. This makes the swarm always choose the short path. Starting from 35 robots, the chain can also be formed over the long path (verified in separate tests not shown here). While the robots' general preference for the short path normally makes the chain form there, fluctuations in the robots' initial distribution lets the chain occasionally choose the long path. Such amplification of fluctuations is a typical phenomenon in self-organizing systems in nature [24]. We also conducted tests moving the targets so as to reduce the difference between d_s and d_l (swarm size 50). This led to proportional changes in the number of runs choosing the short path.

We also mention that also in uncluttered environments the chain tends to choose short (i.e., straight) paths. This is because any bend in the chain is worked away by robots trying to move directly to the best navigation information. Therefore, we can expect that the communication based navigation algorithm lets a robot swarm move efficiently in a wide variety of different environments. However, one issue is congestion: in all tests we found that very large swarms have reduced performance due to increased congestion.

VI. Implementation on real robots

We implemented the communication based navigation system on real foot-bots [12]. Since this is the robot used as model in the simulation experiments, the robot characteristics (IrRB capacity, robot speed, etc.) are the same as described in section III. We have not yet performed extensive tests like the ones we did in simulation, but we made some long experiments with varying conditions, in order to qualitatively confirm results of this paper. Videos can be seen in on-line supporting material at `www.idsia.ch/~frederick/swarm_navigation/online_material.html`.

Fig. 12. Foot-bots moving in a dynamic chain between two targets

We performed four different tests. In the first two, robots were deployed in a room of 12.7 x 3.4 m^2. Two robots were placed on either end of the room to serve as targets. One robot moved back and forth between the targets while all other robots moved randomly. We started with 14 randomly moving robots, and gradually reduced this number by removing robots during the experiment. In the first experiment, we used "navigation with stopping", and in the second "navigation with random" (see section II). The experiments show how both strategies let the searching robot move smoothly between the targets. For low swarm sizes, navigation with stopping performs worse than navigation with random, as in simulation. In the third experiment, we used navigation with random in a more complex setup, where one target was placed outside the room, and the robot needed to navigate through a door. This added complexity did not give problems for the system. In the fourth experiment, we let a swarm of 15 robots move back and forth between two targets in the room. As in section IV, a dynamic chain forms. A snapshot is shown in figure 12. We gradually reduced the swarm size, and observed that the chain was stable until about 6 robots. When we increased the swarm size again, the chain restored itself. We also moved the targets around to see the chain adapt, and observed how it always focused on a straight path. We point out that during all tests there were robot failures. This never affected the swarm's behavior noticeably, showing its robustness. Unfortunately, we have not yet had the opportunity to perform real robot experiments for the shortest path behavior.

VII. Related work

Several works address communication aided navigation of a single robot to a target. Many of these use a static network of communicating sensor nodes to guide a mobile robot [25], [26]. Some use mobile robots to deploy the static nodes [27], or to fill gaps in the sensor network [28]. The approach closest to ours is [29], where a navigating robot gets support to move around obstacles from a few dedicated explorer robots, using line-of-sight communication.

For the task of navigating a swarm of robots between two targets, most work is based on indirect stigmergic communication, inspired by ant behavior [3], [5], [6], [7],

[8]. An important problem for such approaches is how to physically implement the pheromone used by ants to mark the navigation trail. A common solution is to mark the trail with a chain of robots [3], [6]. Compared to our system, this has the disadvantage that some of the robots remain static and cannot take part in navigation. Moreover, the system is vulnerable to failures of robots in the chain, making it less robust. In [4], a method based on direct (rather than indirect) communication is proposed: robots exchange landmark trails to help each other navigate. Finally, in [18], the authors address the swarm navigation problem with neuro-evolution. Interestingly, they find a swarm level behavior that is similar to our dynamic chain, though based on very different individual robot behavior (using visual feedback, robots turn around in local dynamic chains; these chains merge and grow and may eventually include the targets). However, this behaviour was not designed to generalize to environments that are radically different from the uncluttered scenario in which it was developed.

VIII. Conclusions and future work

We have presented a navigation system for robotic swarms. It is a simple and flexible algorithm that can be used in different contexts. We have first shown how it allows robots of a swarm to guide a single robot to a target, without the need to adapt their own movements. Then, we have investigated how the system can be used for collective swarm navigation between two targets, a common task in swarm robotics. We have shown that cooperation improves navigation performance, and that when enough robots are present, the swarm self-organizes into a dynamic structure that supports efficient navigation and is robust to perturbances and robot failures. Moreover, we have shown that swarm navigation has a preference for short paths, similar to pheromone mediated navigation in ant colonies. In tests with real robots, we have shown the feasibility of the approach.

In future work, we will first investigate more complex scenarios. We will study single robot navigation with different, realistic robot movement patterns, and test the dynamic chain behavior in complex cluttered environments. Also, we will perform extensive tests with real robots to confirm all results from simulation. After that, we will integrate this system in other scenarios of swarm deployment, e.g., where the swarm performs tasks in service of humans. Many such scenarios require navigation. Moreover, the swarm communication we use for navigation can be extended to carry more information, e.g. for task allocation, planning, etc.

References

[1] M. Dorigo and E. Sahin, "Guest editorial: Swarm robotics," *Autonomous Robotics*, vol. 17, no. 2–3, 2004.

[2] T. De Wolf and T. Holvoet, "Emergence vs. self-organisation: Different concepts but promising when combined," in *Engineering Self-Organising Systems*, 2005.

[3] B. B. Werger and M. J. Mataric, "Robotic food chains: Externalization of state and program for minimal-agent foraging," in *Proc. of the 4th Int. Conf. on the Simulation of Adaptive Behavior (SAB)*, 1996.

[4] R. Vaughan, K. Støy, G. Sukhatme, and M. Matarić, "Lost: Localization-space trails for robot teams," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, 2002.

[5] K. Sugawara, T. Kazama, and T. Watanabe, "Foraging behavior of interacting robots with virtual pheromone," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2004.

[6] S. Nouyan and M. Dorigo, "Chain based path formation in swarms of robots," in *Proc. of the 5th Int. Workshop on Ant Algorithms and Swarm Intelligence*, 2006.

[7] S. Garnier, F. Tache, M. Combe, A. Grimal, and G. Theraulaz, "Alice in pheromone land: An experimental setup for the study of ant-like robots," in *Proc. of the IEEE Swarm Intelligence Symp. (SIS)*, 2007.

[8] F. Ducatelle, G. Di Caro, and L. Gambardella, "Self-organized cooperation between robotic swarms," *Swarm Intelligence*, 2011, to appear.

[9] J. Pugh and A. Martinoli, "Relative localization and communication module for small-scale multi-robot systems," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2006.

[10] A. Gutiérrez, A. Campo, M. Dorigo, D. Amor, L. Magdalena, and F. Monasterio-Huelin, "An open localization and local communication embodied sensor," *Sensors*, vol. 8, 2008.

[11] J. Roberts, T. Stirling, J. Zufferey, and D. Floreano, "2.5d infrared range and bearing system for collective robotics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009.

[12] M. Bonani, V. Longchamp, S. Magnenat, P. Rtornaz, D. Burnier, G. Roulet, F. Vaussard, H. Bleuler, and F. Mondada, "The marXbot, a miniature mobile robot opening new perspectives for the collective-robotic research," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[13] F. Ducatelle, A. Foerster, G. Di Caro, and L. Gambardella, "Supporting navigation in multi-robot systems through delay tolerant network communication," in *Proc. of the IFAC Workshop on Networked Robotics (NetRob)*, 2009.

[14] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. D. Caro, F. Ducatelle, T. Stirling, A. Gutiérrez, L. Gambardella, and M. Dorigo, "Argos: a pluggable, multi-physics engine simulator for heterogeneous swarm robotics," IRIDIA, Tech. Rep. TR/IRIDIA/2010-026, 2010.

[15] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Performance analysis of mobility-assisted routing," in *Proc. of the 7th ACM int. symp. on mobile ad hoc networking and computing (MobiHoc)*, 2006.

[16] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, 1996.

[17] G. Baldassarre, D. Parisi, and S. Nolfi, "Measuring coordination as entropy decrease in groups of linked simulated robots." in *Proc. of the Int. Conf. on Complex Systems (ICCS)*, 2004.

[18] V. Sperati, V. Trianni, and S. Nolfi, "Self-organized path formation in a swarm of robots," *Swarm Intelligence*, 2011, to Appear.

[19] C. Shannon, *The Mathematical Theory of Communication*. University of Illinois Press, 1949.

[20] C. Shalizi, K. Shalizi, and R. Haslinger, "Quantifying self-organization with optimal predictors," *Physical Review Letters*, vol. 93, no. 11, 2004.

[21] G. Baldassarre, "Self-organization as phase transition in decentralized groups of robots: A study based on boltzmann entropy," in *Advances in Applied Self-organizing Systems*, 2008.

[22] T. Balch, "Hierarchic social entropy: An information theoretic measure of robot group diversity," *Autonomous Robots*, vol. 8, no. 3, 2000.

[23] F. Kuiper and L. Fisher, "A monte carlo comparison for six clustering procedures," *Biometrics*, vol. 31, 1975.

[24] I. Prigogine and I. Stengers, *Order out of chaos*. Bantam, 1984.

[25] M. Batalin, G. Sukhatme, and M. Hattig, "Mobile robot navigation using a sensor network," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2004.

[26] K. O'Hara and T. Balch, "Pervasive sensor-less networks for cooperative multi-robot tasks," in *Proc. of the 7th Int. Symp. on Distributed Autonomous Robot Systems (DARS)*, 2004.

[27] M. Batalin and G. Sukhatme, "Coverage, exploration and deployment by a mobile robot and communication network," *Telecommunication Systems Journal,* Special Issue on Wireless Sensor Networks, vol. 26, no. 2, 2004.

[28] U. Witkowski, M. El-Habbal, S. Herbrechtsmeier, A. Tanoto, J. Penders, L. Alboul, and V. Gazi, "Ad-hoc network communication infrastructure for multi-robot systems in disaster scenarios," in *Proceedings of the IARP/EURON Workshop on Robotics for Risky Interventions and Surveillance of the Environment (RISE).* (published online), 2008.

[29] A. Sgorbissa and R. C. Arkin, "Local navigation strategies for a team of robots," *Robotica*, vol. 21, 2003.