Wireless communications for distributed navigation in robot swarms

Gianni A. Di Caro, Frederick Ducatelle, and Luca M. Gambardella*

"Dalle Molle" Institute for Artificial Intelligence Studies (IDSIA)
Galleria 2, 6928 Manno, Switzerland, {gianni,frederick,luca}@idsia.ch

Abstract. We consider a swarm of robots equipped with an infrared range and bearing device (Ir-RB) that is able both to make estimates of the relative distance and angle between two robots in line-of-sight (LoS) and to transfer data between them. Through the Ir-RB, the robots create a LoS mobile ad hoc network (LoS MANET). We investigate different ways to implement a swarm-level distributed navigation function exploiting the routing information gathered within the LoS MANET. In the scenario we consider, a number of different events present themselves in different locations. To be serviced, each event requires that a robot with the appropriate skills comes to its location. We present two swarm-level solutions for guiding the navigation of the selected robots towards the events. We use a bio-inspired ad hoc network routing protocol to dynamically find and maintain paths between a robot and an event location in the LoS MANET, and use them to guide the robot to its goal. The performance of the two approaches is studied in a number of network scenarios presenting different density, mobility, and bandwidth availability.

1 Introduction

In parallel with recent and continuous advances in the domain of mobile ad hoc networks (MANETs), the field of *networked robotics* is attracting an increasing interest. The idea is to equip teams or swarms of robots with wireless communication devices and allow them to exchange information to support cooperative activities and fully exploit ensemble capabilities. In this work, we use networked robotics in a situation where a swarm of robots needs to execute tasks in an indoor area. The term "swarm" refers here to a potentially large group of small robots that collaborate using weak coordination mechanisms [8]. The tasks correspond to events that need to be serviced in given locations. Each event can be taken care of by a single robot able to provide a specific set of functionalities. A full solution to this problem involves mechanisms for announcing events, for the allocation of robots to events and for guiding robots to event locations. Here

* This work was supported by the SWARMANOID project, funded by the Future and Emerging Technologies programme (IST-FET) of the European Commission under grant IST-022888.

we focus on robot navigation: how can a robot find an event's location after the event has been advertised and the robot has been selected. An important aspect of the problem under study is the fact that the robots cooperatively and transparently help each other for navigation while they are at the same time involved in a task of their own. This is different from most of previous works, where all robots are involved in solving a single task cooperatively (e.g. collaboratively guide one robot to a destination [12, 11]).

The robots our work is based on are the *foot-bots* [2], which are small mobile robots developed within the *Swarmanoid* project [1]. They move around on a combination of tracks and wheels, and are equipped with three different devices for wireless communication: Wi-Fi, Bluetooth, and an *infrared range and bearing system (Ir-RB)*. The latter one consists of a number of infrared transmitters and receivers placed all around the robot. The system allows both the wireless transmission of data over short distances along *line-of-sight (LoS) paths*, as well as the estimation of the relative *distance* and *angle* to other robots. We make use of the Ir-RB system to create a LoS MANET between the robots of the swarm (in parallel, the Wi-Fi MANET can be used for announcing and allocating the events as well as to transmit the geographic data regarding LoS paths if the Ir-RB bandwidth is insufficient). The core idea to implement a distributed navigation function is to set up a route over the LoS MANET between a robot that wants to serve an event and the event's location. Since each robot in the swarm is involved in its own task, this MANET presents frequent and unexpected topology changes. Therefore, we rely on a bio-inspired adaptive routing algorithm, called *AntHocNet*, which is able to find and maintain routes in the face of high network mobility and has been shown to give efficient and robust performance also in large networks and in cluttered environments [6, 4]. The established route is used for robot navigation. We distinguish two modes of operation: in the first the robot physically follows (robot by robot) the route formed via the wireless connections, while in the other the route is used to make estimates of the relative position of the event, so that the robot can aim to go there independently of the actual data route.

The proposed system has some important advantages. First, thanks to the use of the Ir-RB system, robots can get relative positioning information without a central reference system such as GPS. Second, since robots transparently guide each other via wireless communication and are supported by an adaptive routing algorithm, they do not need to adapt their movements as is done in other approaches (e.g., based on visual communication [11]). At the same time, they can get involved in tasks of their own, improving the possibilities for parallel task solving. Finally, the possibility to base robot navigation on an estimate of the relative location of the event to be serviced, and independent from the actual MANET route, allows to overcome moments of interrupted network connectivity.

This paper is organized as follows. In Section 2, we describe the robots for which we developed this work. Then, in Section 3, we present our communication aided robot navigation system. After that, in Section 4 we present and discuss the results of our experiments.

2 The robots and their communication interfaces

A foot-bot is about 15 cm wide and long and 20 cm high. It moves on the ground making use of a combination of tracks and wheels. For *basic obstacle avoidance and navigation*, the foot-bot has 24 short-range infrared sensors all around, 8 infrared sensors directed at the floor, and a rotating platform with a long-range infrared sensor (maximum range 150 cm, precision 2 cm), which gives one measurement per second of obstacles all around with a step size of 2^o. For *communication*, the foot-bot has Wi-Fi and Bluetooth as well as the mentioned Ir-RB system, which is made of 26 infrared emitters and 16 receivers, placed around and on top of the foot-bot. Based on the quality of the received signals, it calculates an estimate of the relative bearing and range to other robots in LoS equipped with the same system. The maximum range of the system is about 3 m, and the precision is 20% for range estimates and 30^o for bearing estimates. The system also allows LoS communication with a nominal bandwidth of 40 kbps.[1] Finally, it also contains a number of other features, which are not relevant for the work presented here. For complete details of the foot-bot, as well as of the other robots developed in the Swarmanoid project the reader can refer to [2].

The foot-bots are derived from a previous robot called the *s-bot* [10]. Since the foot-bots are currently not yet available our work is based on *simulation*, whereby the simulator has been derived from an extensively tested s-bot simulator that has been refined and extended to include the new foot-bot features [3].

3 Use of LoS routing paths for robot navigation

The main idea in our approach is to use an ad hoc routing protocol to dynamically set up and maintain data routes in the LoS MANET between the event and the robot that can serve it. We assume that each event is represented by a robot that remains static at the event location and does all the communications for the event. The route information in the LoS MANET is then used directly or indirectly to guide robot navigation towards the event location.

3.1 AntHocNet, the MANET routing algorithm

To establish routes in the MANET, we make use of *AntHocNet*, a MANET routing algorithm based on ideas from *Ant Colony Optimization (ACO)* [5]. Here we give an high level overview of the algorithm. For more details, see [6].

In AntHocNet, a node s requiring a route to a destination d sends out a *reactive forward ant*. This is a control packet that has as a task to find a path to d. At any node i in the network, the reactive forward ant can be locally broadcast or unicast, depending on whether or not i has routing information available for

[1] These are the values that have been used for the simulation experiments at the time this paper was written. However, in the meantime, the hardware design has been improved, such that the latest Ir-RB system provides a much better precision and performance.

d. When a node receives multiple copies of the same ant, it forwards only the first one it receives, and discards all subsequent copies. Once the ant reaches d, it is returned to its source node s, following the same path it came over. On its way back, the ant measures the quality of the path and sets up routing information towards the destination. Path quality is measured based on the signal strength of the wireless links along the path. Routing information takes the form of next hop pointers associated to relative goodness values based on the path quality measurements. These goodness values are called *pheromone values*, in accordance to the ACO inspiration of the algorithm, and are stored in routing tables.

Once the route is set up, the source node s starts a *route maintenance and improvement process*, in order to continuously adapt the existing route to the changes in the dynamic network and find new and better routes when possible. This process is based on two subprocesses: pheromone diffusion and proactive ant sampling. The aim of *pheromone diffusion* is to spread out pheromone information that was placed by the ants. Nodes periodically broadcast messages containing the best pheromone information they have available. Neighboring nodes receiving these messages can then derive new pheromone for themselves (using *information bootstrapping*, similar to Bellman-Ford updating schemes), and further forward it in their own periodic broadcasts. This way, a field of diffused pheromone is set up that points out possible routes towards the destination. However, since this information is based on the use of periodic (low-frequency) broadcast messages, it can temporarily contain erroneous information. This is where the second process, *proactive ant sampling*, comes in. At constant intervals, node s sends out *proactive forward ants*. Like reactive forward ants, these are control packets that try to find a route towards the destination. They follow the pheromone spread through the diffusion process. When they reach the destination, they travel back to the source setting up a route indicated by pheromone. This way, they update and validate the information found through the pheromone diffusion process and find new routes in the changing MANET.

In case of a route failure, AntHocNet sends link failure notification messages. When a node i perceives that a link has failed on an existing route, it broadcasts to its neighbors a message indicating the destination it has lost the route to. A neighbor receiving this message updates its routing information accordingly. If it observes the loss of a route due to this update, it sends out its own notification.

3.2 Network routing and robot navigation

When a robot wants to serve a particular event, it uses the AntHocNet routing algorithm to set up a data route to the robot signaling the event. Once a route is set up, it amounts to a set of nodes (robots) connecting the service robot to the event in the LoS MANET, together with the (noisy) information of their relative *(distance, angle)* location. We foresee two possible ways of using this LoS routing information. The first is that the robot physically follows the data route in the network robot by robot. The other is that the full route information serves to calculate an estimate of the relative location of the event, and the robot moves directly in that direction.

Locally following the routing path. In this approach, once the service robot has established a data route in the LoS MANET, it starts moving towards the estimated location of the nearest robot (next hop) in the data route. While moving, the robot continuously tries to re-sample the route in order to get a more up-to-date estimate of the route towards the event, and, more in particular, of the location of the next hop, which it is moving to. If the route is lost at any time (e.g. due loss of connectivity), the robot remains static and repeatedly tries to establish a new route. According to this behavior, all robots get continuous measurements of the relative distance and angle to each of their neighbors in the LoS network. Since these measurements are affected by precision errors, each robot aggregates them using moving averages:

$$\begin{aligned} \hat{d}_i^j(t) &= \gamma \hat{d}_i^j(t-1) + (1-\gamma) d_i^j(t) \\ \hat{\alpha}_i^j(t) &= \gamma \hat{\alpha}_i^j(t-1) + (1-\gamma) \alpha_i^j(t), \end{aligned} \tag{1}$$

where $\hat{d}_i^j(t)$ is robot i's estimate at time t for the distance to neighbor robot j, and $\hat{\alpha}_i^j(t)$ is i's estimate of the angle towards j with respect to its own orientation. $d_i^j(t)$ is the new measurement for the distance received by i at time t, and $\alpha_i^j(t)$ is the new measurement for the angle. $\gamma \in [0, 1[$ defines how quickly the local estimate is adapted to new measurements (we use $\gamma = 0.7$ in the experiments).

Having the robot physically follow the data route has a number of advantages and disadvantages. A first advantage is that it is a simple process. A second advantage is that it provides obstacle-free paths. This is because routes are composed of LoS links, that is, of a feasible path to the event. A disadvantage is that the robot can have difficulties following the path when it changes often and abruptly. This can happen when the robots move a lot or in the presence of obstacles. Another disadvantage is that the robot does not know where to move when there is no route available. This leads to low performance in cases of intermittent network connectivity, e.g. when there are few robots around or when obstacles block signals. A final disadvantage is that the path followed by the robot can be substantially longer than the shortest path, especially when the shortest path in the LoS MANET does not correspond to the geographic shortest path (e.g., this can easily happen when robot density is low [13]).

Following path-level estimates of destination location. In this approach the constructed data route is used to give the searching robot an estimate of the relative distance and angle to the event location, so that it can move there directly without following the route. According to AntHocNet's behavior, to refresh the data route, the service robot periodically sends proactive forward ants towards the destination node (the event robot), which are then sent back to set up a new data route. On their way back, we let these ants gather the locally maintained estimates of the distance $\hat{d}_i^j(t)$ and angle $\hat{\alpha}_i^j(t)$ to each next hop and previous hop (see Equation 1) and combine them geometrically to make an estimation of the relative distance $D_i^n(t)$ and angle $A_i^n(t)$ to the event location n. We represent the path followed by the ant as $P = (1, 2, \ldots, n-1, n)$, whereby node 1 is the searching robot and node n is the event location (so the ant travels from n to 1). At any node $i < n$ on this path, $D_i^n(t)$ and $A_i^n(t)$ are incrementally calculated as follows, where the common index t is dropped for notational clarity:

$$
\begin{aligned}
D_i^n &= \begin{cases} \hat{d}_i^n & \text{if } i = n-1, \\ \sqrt{(\hat{d}_i^{i+1})^2 + (D_{i+1}^n)^2 - 2\hat{d}_i^{i+1} D_{i+1}^n \cos\left(A_{i+1}^n - \hat{\alpha}_i^{i+1}\right)} & \text{if } i < n-1. \end{cases} \\
A_i^n &= \begin{cases} \hat{\alpha}_i^n & \text{if } i = n-1, \\ \arccos\left[\frac{(\hat{d}_i^{i+1})^2 + (D_i^n)^2 - (D_{i+1}^n)^2}{2\hat{d}_i^{i+1} D_i^n}\right] & \text{if } i < n-1. \end{cases}
\end{aligned}
\tag{2}
$$

Once the searching robot has received a first estimate of the distance D_1^n and angle A_1^n towards the event location, it starts moving in straight line towards its goal. As the robot is going, the routing algorithm keeps sending proactive ants regularly, making new estimates available. Having a continuous stream of new estimates is important to overcome errors in previous estimates and to keep an updated view of the event location. Errors in the estimate stem from two main causes. First of all, the event location estimate is based on a composition of local distance and angle estimates along the links of the paths, each of which contains some error, and therefore the total estimate has an error that increases with the number of hops. Hence, at large distances, the event location estimate only offers a rough guideline for the robot's movements, while at smaller distances, the estimate becomes more accurate. The second source of errors is due to the robot's own movements. As the robot is going, it needs to adapt the location estimates according to its own rotations and displacements, using feedback from the local odometry. This causes the estimate to gradually become less reliable. Therefore, the periodic sending of proactive ants is needed to keep renewing it.

Using estimates of the relative event location has the advantage that the robot is not directly dependent on the LoS route itself or on its persistence for its movements. It is sufficient to get a new estimate from time to time to update the global estimate. Figure 1 shows the sample of a typical behavior in the error of the distance estimate in the basic experimental setting (see next section). The error shows large fluctuations at the beginning, that is at large distances, but also a generic decreasing trend and a rapid convergence to zero when approaching the event location. On the other hand, this approach cannot guarantee an obstacle free path, and can run into problems in presence of many obstacles scattered along the straight line between the searching robot and the event location.

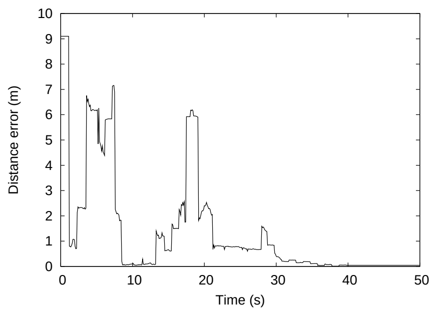

Fig. 1. Typical evolution of the error in the distance estimate for follow-estimate.

4 Experimental results

All tests are done using the Swarmanoid simulator [3]. For each test scenario we execute 30 independent runs. We report the average with 95% confidence interval of the time needed for the robot to reach the event and of the distance traveled compared to the straight line distance. We compare three different navigation behaviors: the two proposed ones (from now on referred to as *Follow route* and *Follow estimate*) and a *sweeping behavior*, used as a reference of the performance that is possible when no communication is used. In this behavior, the robot knows its location in the room at all times. It goes to the room corner that is closest to its start location and then starts scanning the room in straight lines parallel to one of the room walls, until it finds the destination. Moving steps in the direction of the other room walls are proportional to the radio range. We use an open space room of 10×10 m^2. The service and the event robot are located in opposite corners. The other robots move according to the *random waypoint mobility model (RWP)* [9] in order to simulate the fact that they are involved in tasks of their own and their movements are independent from the task of guiding the searching robot. They choose a random destination, move to it, pause for some time and then choose a new random destination. This model fits well the movements of robots that service sequences of events. All robots are equipped with a minimal obstacle avoidance mechanism. We report experiments to study the effect of varying the number of robots and their speed, that equals to study the effect of varying network density and its topological changing rate. We also study the effect of changing the proactive ant send interval and that of increasing the fraction of packet losses during communications. Results concerning the effect of the presence of obstacles and of multiple events can be found in [7].

Effect of scaling the number of robots. We investigate the influence of the number of robots on the ability of a searching robot to find an event location. We vary the number of robots from 10 up to 50. The speed of the robots is 0.15 m/s, the pause time of the RWP model is 6 s. The results are shown in figure 2. As can be seen from the graphs, a lower number of robots makes the task difficult

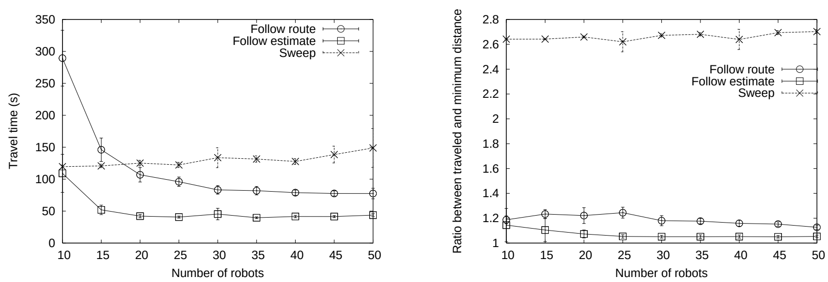

Fig. 2. Results for tests with increasing numbers of robots.

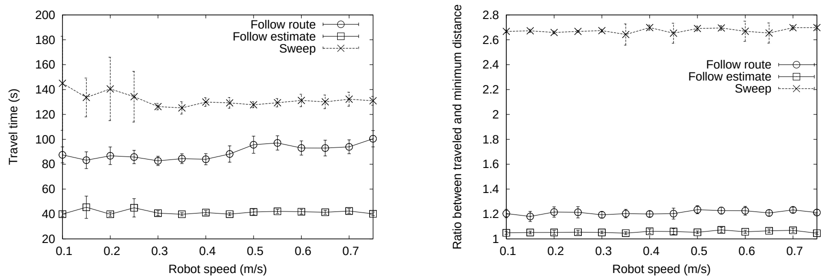

Fig. 3. Results for tests with increasing robot speed.

for the communication based behaviors. This is because there is limited network connectivity and the task to establish and maintain a stable route is therefore difficult. This affects especially *Follow route*, which depends on the constant availability of a route: for this behavior, there is a strong increase in the travel time for the searching robot. For *Follow estimate*, the increase of the travel time is only visible for the lowest number of robots. Interestingly, the travel distance is not much affected for either of the behaviors. Overall, *Follow estimate* needs less time and produces shorter paths than *Follow route*. The *Sweep behavior* needs much more time and produces much longer travel distances, especially as the number of robots increases. This indicates the general usefulness of using telecommunications for navigation when a large group of robots is available.

Effect of robot speed. The speed of the searching robot is set to 0.35 m/s. For the other robots, the speed is varied from 0.1 m/s up to 0.75 m/s. The total number of robots is fixed to 30. The results for increasing movement speed are shown in Figure 3. It is interesting to see that the speed of the robots has little influence on the performance. For *Follow route* there is a small increase in the robot travel time, but no noticeable effect on the traveled distance. For *Follow estimate*, there is no noticeable effect for either measure. The higher vulnerability of the *Follow route* behavior is to be expected, as high node mobility leads to higher variability and more frequent disconnections of the LoS MANET route, so that it becomes difficult to follow it hop by hop. The robustness of our approach with respect to robot speed is an important advantage for its deployment.

Effect of the ant send interval. We evaluate the effect of changing the time interval in the periodic transmission of proactive forward ants. This parameter defines the frequency of route updates and hence also the load on the MANET. In previous tests, we used intervals of 1 s, here we run tests up to 10 s. The results are shown in Table 1. *Follow route* is unaffected by the ant send interval, while *Follow estimate* shows a slight decrease of performance with increasing intervals. This is because *Follow estimate* needs a continuous stream of ants to keep reducing the error on its estimate of the event location, while *Follow route*

Table 1. Effect of changing the send interval of proactive ants.

Proactive ant send interval	1	2	3	4	5	6	7	8	9	10
Follow route: Distance ratio	1.24	1.21	1.20	1.21	1.22	1.20	1.12	1.20	1.21	1.22
Follow route: Travel time	96	96	95	94	93	93	90	90	95	93
Follow estimate: Distance ratio	1.05	1.06	1.09	1.10	1.10	1.10	1.10	1.14	1.15	1.14
Follow estimate: Travel time	41	41	42	42	43	43	43	44	45	45

relies only on the presence of the route and can therefore more easily function with a lower frequency of proactive ants.

Effect of the packet losses. In the Swarmanoid simulator, the MAC and PHY layers are simulated with a probabilistic model based on two parameters defining the probabilities of packet loss, θ, and of signal interference (both set to 0 in the previous experiments). Here, we increase θ from 0 up to the extreme value of 0.5. An increase in θ results in the decrease of the ability of AntHocNet to gather up-to-date routing information. The results for the case of 30 robots are shown in Figure 4. *Follow estimate* shows no noticeable effect in travel time. In terms of distance, the average value is stationary but the variance gets larger and larger. In fact, in presence of large packet losses, the robot can be forced to rely on poor estimates for relatively long time periods, especially at the beginning of the search, when paths are longer. Therefore, the initial fluctuations shown in Figure 1 may become more severe in some scenarios. *Follow route* depends critically on the continual availability of route updates. Accordingly, its time performance shows a rapid degradation for $\theta \geq 0.3$. The distance performance is not much affected since the robot stops in absence of new route information.

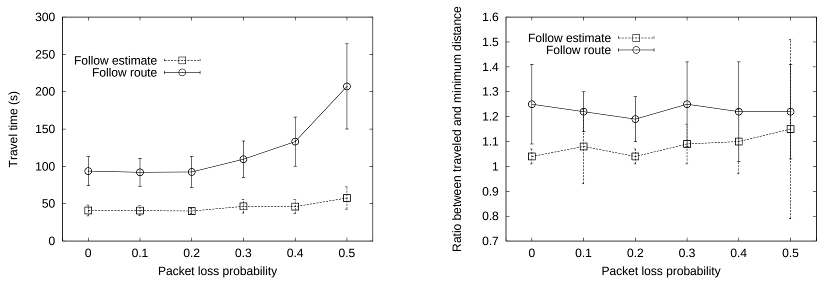

Fig. 4. Results for tests with increasing probability of packet losses.

5 Conclusions and future work

We have investigated the use of LoS wireless networking to support cooperative navigation in a swarm of mobile robots. We have based our work on the *foot-bots*, small robots developed in the Swarmanoid project [1], that are equipped with a LoS infrared device that can be used to both transmit data and return the relative distance and angle between two robots. We have proposed two solutions based on the use of a bio-inspired ad hoc network routing protocol to discover and maintain data routes between a robot and an event location in the LoS network. The established route is used for robot navigation in two different ways: in the first, the robot physically follows the route formed via the wireless connections, in the other the route is used to make estimates of the relative position of the event. We ran a number of simulation experiments varying the number of robots and their speed, and studying the sensitivity of the algorithms to variations in the number of control packets used to gather routing information and in the probability of packet losses. Both algorithms showed an overall robust behavior, with the approach based on location estimates outperforming the other. In the future, we will consider also the other types of robots forming the 3D Swarmanoid and integrate the navigation function in a task allocation architecture.

References

1. Swarmanoid: Towards humanoid robotic swarms. `http://www.swarmanoid.org`.
2. Hardware design. Internal Deliverable D4 of IST-FET Project *Swarmanoid* funded by the European Commission under the FP6 Framework, 2007.
3. Simulator prototype. Internal Deliverable D7 of IST-FET Project *Swarmanoid* funded by the European Commission under the FP6 Framework, 2008.
4. G.A. Di Caro, F. Ducatelle, and L. M. Gambardella. A simulation study of routing performance in realistic urban scenarios for manets. In *Proc. of the 6th Int. Workshop on Ant Algorithms and Swarm Intelligence*. Springer, 2008.
5. M. Dorigo, G. Di Caro, and L. M. Gambardella. Ant algorithms for distributed discrete optimization. *Artificial Life*, 5(2):137–172, 1999.
6. F. Ducatelle, G. Di Caro, and L. M. Gambardella. Using ant agents to combine reactive and proactive strategies for routing in mobile ad hoc networks. *International Journal of Computational Intelligence and Applications (IJCIA)*, 5(2), 2005.
7. F. Ducatelle, G.A. Di Caro, and L. M. Gambardella. Robot navigation in a networked swarm. In *Proceedings of the International Conference on Intelligent Robotics and Applications (ICIRA)*, volume 5314 of *LNAI*. Springer, 2008.
8. A. Farinelli, L. Iocchi, and D. Nardi. Multirobot systems: a classification focused on coordination. *IEEE Trans. on Sys., Man, and Cyb., Part B*, 34(5), 2004.
9. D. B. Johnson and D. A. Maltz. *Mobile Computing*, chapter Dynamic Source Routing in Ad Hoc Wireless Networks. Kluwer, 1996.
10. F. Mondada, G. Pettinaro, A. Guignard, I. Kwee, D. Floreano, J.-L. Deneubourg, S. Nolfi, L.M. Gambardella, and M. Dorigo. SWARM-BOT: a new distributed robotic concept. *Autonomous Robots*, 17(2–3), 2004.
11. 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.
12. D. Payton, M. Daily, R. Estowski, M. Howard, and C. Lee. Pheromone robotics. *Autonomous Robots*, 11(3), November 2001.
13. T.C.H. Sit, Z. Liu, M.H. Ang Jr., and W.K.G. Seah. Multi-robot mobility enhanced hop-count based localization in ad hoc networks. *Robotics and Autonomous Systems*, 55(3):244–252, March 2007.