Memetic Comp. (2011) 3:245–259
DOI 10.1007/s12293-011-0061-z

REGULAR RESEARCH PAPER

# A social approach for target localization: simulation and implementation in the marXbot robot

**Héctor F. Satizábal · Andres Upegui · Andres Perez-Uribe · Philippe Rétornaz · Francesco Mondada**

Received: 1 December 2010 / Accepted: 26 May 2011 / Published online: 10 June 2011

**Abstract** Foraging is a common benchmark problem in collective robotics in which a robot (the forager) explores a given environment while collecting items for further deposition at specific locations. A typical real-world application of foraging is garbage collection where robots collect garbage for further disposal in pre-defined locations. This work proposes a method to cooperatively perform the task of finding such locations: instead of using local or global localization strategies relying on pre-installed infrastructure, the proposed approach takes advantage of the knowledge gathered by a population about the localization of the targets. In our approach, robots communicate in an intrinsic way the estimation about how near they are from a target; these estimations are used by neighbour robots for estimating their proximity, and for guiding the navigation of the whole population when looking for these specific areas. We performed several tests in a simulator, and we validated our approach on a population of real robots. For the validation tests we used a mobile robot called marXbot. In both cases (i.e., simulation and implementation on real robots), we found that the proposed approach efficiently guides the robots towards the pre-specified targets while allowing the modulation of their speed.

**Keywords** Collective robotics · Target localization · Foraging · marXbot

## 1 Introduction

There is a large amount of real world tasks where a group of robots performs better or more efficiently than a single robot [2]. The approach of using multiple robotic agents to perform a task in a cooperative manner is called *collective robotics*. Collective robotics has been used for a diversity of tasks like object manipulation, obstacle overpassing, and stair climbing. In all cases, collective robotics targets the execution of tasks that, when performed by a single robot, are impossible or inefficient. Researchers argue that by organizing simple robots into cooperating teams, useful tasks may be accomplished otherwise impossible using a single robot [15].

Collective robotics has gained the interest of a large number of researchers in the last decades thanks to the wide range of possibilities of applications that it offers. However, controlling such systems demands the use of coordination strategies taking advantage of the fact of having more than one individual, i.e., the presence of neighbours which can cooperate to ease the execution of the task. The goal is thus to find a strategy that allows a set of robots to, somehow, interact among them in order to find the solution in a more efficient manner than the same set of robots performing the task simultaneously but independently. Including such interaction implies additional costs in terms of robot set-up and computation, like addition of communicating capabilities or attachment mechanisms. In spite of this additional cost, the collective solution must still be more efficient than the individual one.

A common benchmark problem in collective robotics is *foraging*. The term foraging in robotics is used to designate a group of behaviours that mimic the behaviour of foraging in animals. Thus, a forager robot has to navigate in its environment while collecting items and depositing them at specific locations [4,27]. In order to perform this type of task, robots

H. F. Satizábal (✉) · A. Upegui · A. Perez-Uribe
REDS, University of Applied Sciences Western Switzerland,
Yverdon-les-Bains, Switzerland
e-mail: hector-fabio.satizabal-mejia@heig-vd.ch

P. Rétornaz · F. Mondada
LSRO-MOBOTS, Ecole Polytechnique Fédérale de Lausanne,
Lausanne, Switzerland

need to be able to explore their environment in an efficient way, and at any moment, find target locations which are common to the whole population, such as storage places where the collected objects have to be stacked, battery charging stations, or specific sites if a fixed path has been established. Foraging has been largely studied due to its importance[1] in collective robotics, and it has been divided into several states namely: *searching*, *grabbing*, *homing* and *depositing*; and the problem of coordination between robots can exist at every state. In the work we describe here, we are interested in the problem of coordination for the first and last tasks, i.e., searching and homing.

Different approaches have been used in designing control strategies for groups of searching robots. There is a classical approach where a central planning unit coordinates the actions of the population of robots. This unit sends commands according to the state of each unit in order to make them cooperate. The distribution of labours can be hierarchical, and each individual must be capable of replacing the planner unit if it fails due to malfunction [4]. This approach, while being the more intuitive and understandable, is often not scalable and difficult to implement due to the communication requirements of a central coordination, which in addition makes the system less robust.

An alternative to this approach consists on endowing the system with self-organization properties, allowing individual units to cooperate without a central planner. Self-organization is frequently achieved by taking inspiration from biology [12], and in particular from the behaviour of social species. Social species of insects, for instance, are very successful in performing cooperative tasks related to the survival of the species.[2] Their success might come from the fact that social interactions can compensate for individual limitations, both in terms of physical and cognitive capabilities. Indeed, herds and packs allow animals to attack larger prey and increase their chances for survival and mating [14], while organizations and teams facilitate information sharing and problem solving.

Some examples of coordination strategies found in biology are stigmergy in ants [10] and trophallaxis in bees [9]. Stigmergy refers to a specific type of social communication through the modification of the environment, where the result of work of an individual modifies the behaviour of the rest of the population in a unintentional manner [13]. It is a form of self-organization. The construction of nests in termites and other insects are examples of stigmergy in nature. Moreover, trophallaxis is the exchange of fluid food by direct mouth-mouth contacts between members of the population. These exchanges can transfer information about the quality of food source, temperature and water demand. These mechanisms have served as inspiration to develop controllers for collective robotics, and there is a large number of different implementations of these concepts, each one requiring different levels of complexity of the robots, and different types of communication between the units. Schmickl and Crailsheim [23] used trophallaxis as inspiration for creating a decentralized strategy of communication in a swarm of small robots. By using this approach, they manage to create two gradients into the population, and to make the robots employ these gradients in order to navigate towards pre-defined areas. Sugawara et al. [25] employed stigmergy as inspiration to make a population of robots to cooperate in the task of collecting virtual food in a controlled environment. They solved the main problem of stigmergy of modifying the environment with chemical traces by using images projected on the floor where the robots can detect them. Campo et al. [11] employ virtual pheromones to guide the process of path selection in a population of foragers. The pheromones in this case are local messages which are passed between members of a chain of robots.

This paper describes a novel approach for the localization of targets (e.g., searching and homing behaviours) in a population of foragers. The control of the population of robots is performed in a distributed way. In order to test our approach we designed an experimentation scenario where robots have two possible states which are “work” and “search”. In the “work” state robots perform a certain foraging task and are distributed on the arena. In the case of the work presented in this paper, we have a dummy foraging task consisting in navigating on the arena avoiding obstacles. The main interest is in the “search” state, where a robot will try to arrive to a specific target region on the arena. This target region can be a battery charging station, an area for garbage disposal, or the output of a maze. Whatever the robot may search, our hypothesis is that a searching robot can exploit the collective knowledge, given that there may be other robots that can estimate how far they are from the target region, and will somehow help the searching robot to achieve its goal. The proposed target localization avoids the use of global positioning systems, that might be difficult to deploy in unknown or hostile environments, and avoids also the use of odometry, which is sensitive to cumulated errors after large running periods. However, robots need basic communication capabilities[3] in order to disseminate the information they have to the rest of the population. We decided to use light in the visible spectrum as communication channel in order to render the exper-

[1] Foraging is important since it is a metaphor for a broad class of problems integrating exploration, navigation and object identification, manipulation and transport, and robot-robot cooperation. Moreover, many actual or potential real-world applications for robotics are instances of foraging robots, for instance cleaning, harvesting, search and rescue, land-mine clearance or planetary exploration [27].

[2] It has been estimated that one-third of the animal biomass of the Amazon rain forest consists of ants and termites [24].

[3] For the purpose of disseminating information in the population, a communication schema which is local and implicit would be enough.

iments more visual. In this manner, we simplify the process of assessing the behaviour of the robots in a qualitative way, and provide interesting and easily interpretable demonstrations. Our approach for the implementation uses the colour LEDs and omni-directional cameras of the robots as communication link. Thus, each robot encodes in its colour the information of how likely it is to find a target in its proximities, and each robot can "see" what neighbouring robots are showing by using its own camera. This paper complements the initial tests shown on [22] about this approach of communication and coordination, and shows its validation by means of an implementation on real robots. The simulations were performed with the simulator Enki [19], and the tests on the real arena were performed by using 20 marXbot robots [6,7].

The paper is structured as follows. Section 2 introduces the use of topological navigation and the use of landmarks, and describes the use of state communication in coordinating a population of robots. Section 3 describes the simulation framework that was used in order to do the first tests of the target localization strategy, and the robots and sensors implemented. Several results of the simulations are also shown in this section. Section 4 describes the real robot that was used for validating the social-based approach, the results of the tests we performed over the population, and a comparison with the simulation. Section 5 gives some conclusions.

## 2 Searching and homing in a forager robot

As it has been already mentioned, localizing a target (e.g., searching, homing) is one of the essential tasks of a forager robot. When foraging, robots have to be able to find target locations which are common to the whole population in order to collect some resources, or to deposit them in a pre-established position. Finding a target zone is thus a crucial behaviour for a robot being part of a swarm of foragers. Several approaches have been used with this purpose, e.g., omniscient planners, sensing absolute position/orientation, following global beacons, using landmarks, pheromones, beacon chains or contact chains [26]. Global strategies like the use of centralized planners or GPS-like systems are expensive and difficult to implement or unreliable when the number of robots increases or when robots are placed in changing or hostile environments. Conversely, local strategies like the use of local beacons or landmarks, or bio-inspired methods like pheromones, are more easily scalable and allow the implementation of self-organized systems which can adapt to unknown environments.

### 2.1 Landmarks and beacons

The use of landmarks in robot navigation is a widely used approach which has been called *topological navigation* [4]. Robots using this strategy do not use precise measurements of position but have to infer their own location from the perception of known marks in the environment such as doors or intersections in the case of indoor navigation. Topological navigation is common for us since most of the information we use to locate ourselves and target directions are relative to objects in our landscape. Nevertheless, using landmarks is not exclusive to superior animals. Some researchers have taken inspiration from small social animals like insects which employ similar strategies to find their way back to home after exploration journeys. Some species of desert ants, for instance, use visual landmarks in order to return to important places in their environment [18] when other methods like the use of pheromones is not possible. Moreover, other individuals which make part of the population can also be landmarks. Bees, for instance, can use the physical contact with other individuals of the hive in order to regulate the behaviour of foragers [9]. This form of communication, where individuals employ other members of the population as landmarks or beacons for locating a target has also been a source of inspiration for navigation in robotics [21,26].

### 2.2 Social approach of target localization

One of the most important aspects in controlling the coordination of a group of agents is the communication between those agents. Different considerations can be done about the type of communication within the members of the population. The communication can be global if the whole set of agents can receive messages, or local if only neighbour agents can communicate. Moreover, the communication can be explicit or implicit whether the sender is aware of the receiver or not, or even targeted if the sender is able to select the receiver. As it has been already mentioned, local communication schemas are more scalable to large populations, and implicit communication is easier to implement. Indeed, it is not by coincidence that the communication schemas in social species of insects have these properties.

A type of communication in collective robotics having these properties is *state communication* [3]. In state communication robots communicate through their behaviour in explicit or implicit manner. Hence, robots have to be able to interpret the behaviour of other robots by using their sensory capabilities. This type of communication has been successfully used for coordinating tasks in collective robotics [16,17,21] and it has proven to be robust and scalable.

State communication can be used to transfer information about the location of a specific place and hence, a basis of a social approach of target localization in a population of foragers. In this paper, we present a novel approach for finding a common target location based on the knowledge gathered by a population of robots. It supposes the existence of a set of robots performing a foraging task. A group of robots is thus distributed in the environment while searching for some kind of resource, and at any moment, any individual has to find a specific place which is common for the whole population e.g., a charging station, or a depot where gathered objects have to be stacked. Robots are not provided with their positions when looking for the targets, instead, each individual has to use imprecise nearness estimations of neighbours which are transferred through state communication. The physical means that are used for establishing the communication link are not relevant for testing the validity of our approach. What is important is that the communication must be local and implicit.

For the sake of easily visualising what is happening in the population, we decided to use visible light as physical layer for establishing the communication. In order to share the knowledge with its neighbours, every robot can display a colour, and that colour reflects an internal state of the robot which is directly related to the certitude of being near the target. Each robot has limited vision which allows it to detect other robots as well as obstacles. Thus, if a robot needs to go to a specific place, it has to follow robots showing the colour that was assigned to this place. These coloured robots act as moving beacons to guide other members of the population to the specific goals. Once arrived to the target, the robot must update its colour in order to cooperate with the rest of the population serving as beacon for other robots while linearly decreasing its colour. Moreover, robots were programmed to copy a proportion of the colour of other robots, and as a consequence, an emerging colour gradient is formed in the population. This behaviour improves the dissemination of information through the robots, facilitating the task of looking for a target. Any robot in the population behaves as a mobile beacon, and cooperates with the execution of the task by guiding other robots to the target, even if the exact position of the target is unknown. The details of the implementation of the social strategy are shown in Sects. 3.2 and 4.3.

Note that this approach does not exactly mimic stigmergy given that the robots do not produce any trace in the environment (as it happens with the pheromones of ants); and it does not exactly mimic trophallaxis given that the robots do not lose any of the information they have when they communicate (as in the case of bees for instance).

### 2.3 The proposed foraging task

We performed several experiments in order to validate our social-based approach of target localization. In our foraging task, robots will only perform the *searching* and *homing* parts of foraging, and the components of *grabbing* and *depositing* are not present. In our approach robots may be in one of two possible states: *working* or *searching*. In the *working* state, robots freely (i.e., randomly) navigate on the arena while avoiding obstacles (i.e. walls and other robots). In the *searching* state, robots must look for one of the targets that are distributed in the arena. Robots that happens to find the targets must change their colour in order to share this information with other members of the population. Furthermore, every robot in the population must copy the $R$ and $B$ colour components of the robots within its field of view having the highest values for these components. Robots in the *searching* state should follow the direction of the colour gradient that was created by the population of robots in order to head in the direction of the target.

In order to validate our approach, we placed two targets in opposite corners of the arena, and set the state of some of the robots in the population to *searching*. These robots repeatedly go from one target to the other, while at the same time we measured the time each robot spent in its trajectory.

## 3 Simulations

This section shows some of the experiments we carried out in simulation. Some of these experiments where already introduced on [22], and others are completely new experiments that were designed in order to further test our approach.

### 3.1 Simulation setup

We used a simulator called Enki [19] to perform the first tests of the proposed strategy. Enki is an open source 2D robot simulator written in C++ which provides collision and limited physics support for robots evolving on a flat surface.

#### *3.1.1 The simulated arena*

The flat space where robots evolve is a square arena limited by dark gray[4] walls of 15 cm height. There are two simulated RFID tags located at opposite corners of the arena. Each tag is programmed to be detected at a maximal distance of 21.5 cm. These areas are shown by the two light grey circles located at the top-left corner for tag number 1, and bottom-right corner for tag number 2. This setup is shown in Fig. 1.

#### *3.1.2 The simulated robots*

The robots implemented on Enki were programmed to simulate a real robot called marXbot [6,7]. This robot has two wheels[5] for locomotion, $[RGB]$ LED for displaying colours, omni-directional camera, infra-red bumpers, rotating distance sensor scanner, and a RFID tag detector. The simulated omni-directional camera is a vector of 180 pixels that

[4] R = 30%, G = 30%, B = 30%.

[5] The actual robot has a combination of tracks and wheels called treels.

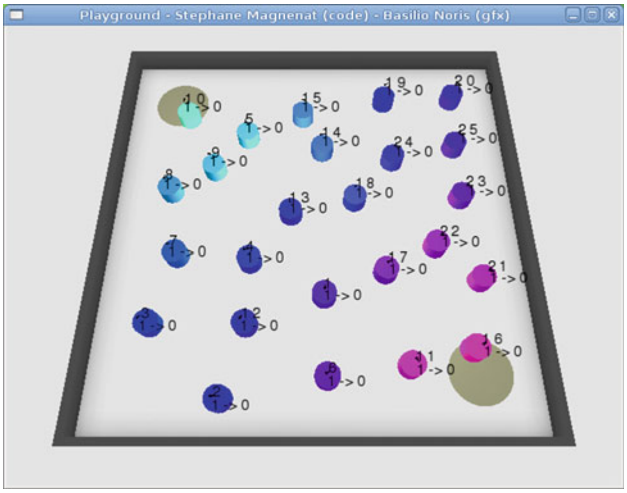


**Fig. 1** Arena where the simulated experiments evolved. The *numbered cylinders* are the mobile robots and the two *grey circles* represent the zones from where RFID tags can be detected. Tag 1 is placed in the *top-left corner*, and tag 2 is placed in the *bottom-right corner*

detects the colours of the objects around the robot (i.e., one pixel of the linear image covers 2 degrees of field of view). A picture of the actual robot is shown in Fig. 9a.

## 3.2 Implementation of the social approach in simulation

The simulated robots were programmed to avoid obstacles (i.e., walls and other robots), to detect two specific zones (i.e., the targets), and to communicate their state in order to collaborate with the population in the task of foraging.

### *3.2.1 Detection of targets and state communication*

Two simulated RFID tags where placed in two opposite corners of the arena, in a way that robots can detect them by using their RFID detectors when approaching to a certain distance (i.e., 21.5 cm). These RFID tags represent the targets the robots have to detect when navigating. Additionally, when a robot finds a target (i.e., it detects the RFID tag), it can communicate this information to the rest of the population by showing one of two colours. Hence, for the simulations we present here, when a robot detects the target number 1 it sets the *green* component of its colour to 100%, and in the other case, when a robot detects the target number 2 the robot sets the *red* component of its colour to 100%. Showing a colour (which must be known by the whole population) when a robot finds a target allows other robots in the proximities of the target to know its location. Once the robot leaves the target zone it linearly decreases all the components of its colour.

Additionally, robots can further share their knowledge by copying the colour of neighbouring robots. The idea with this behaviour is that the colour of the robots indicate how certain a robot is of the information it has. Hence, robots can follow the gradient of colour in order to get to a target. This gradient was created by simulating a limit on the maximal range of vision of the omni-directional camera. After some tests, this maximal range was set to ~50 cm as shown in Fig. 2.

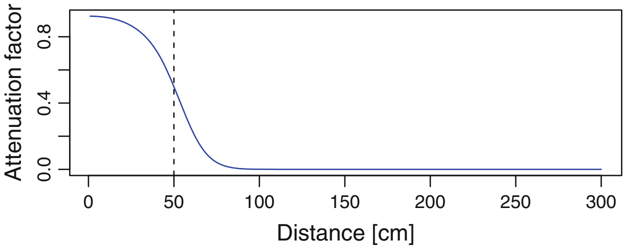


**Fig. 2** Attenuation of the colours detected by the simulated omni-directional camera with respect to the distance to the object

### *3.2.2 Navigation*

Navigation was performed in a pure reactive manner as in Braitenberg vehicles [8], and the integration of sensor information was based on a strategy called *motor schema-based navigation* [1]. Hence, 24 infra-red sensors were used as bumpers, and the 180 pixels of the omni-directional linear camera were employed in order to detect mid-range and distant obstacles and colours. The steer direction **S** was thus calculated by adding 4 components:

- Random (**r**): A vector pointing to a random direction.
- Bumpers (**b**): The vector pointing in the direction where there are no obstacles detected by the bumpers.
- Free Area (**f**): The vector pointing in the direction where there are no obstacles detected by the camera.
- Attraction to Landmark (**t**): The vector pointing in the direction where there are objects having the colour associated to the target.
- Repulsion (**nt**): The vector pointing in the direction where there are no objects having a colour associated to a different target

$$S_x = r_x + b_x + f_x + t_x + nt_x \tag{1}$$
$$S_y = r_y + b_y + f_y + t_y + nt_y \tag{2}$$

Component **r** was added to allow the robot to get out from corners, or other places where sensory information is symmetric, and to guarantee that every run of the experiment is different.

Each one of the aforementioned components was calculated as the dot product of the vector having the response of the sensor group, and the vector of positions of each individual sensor. Let us take the bumpers as an example. We build two vectors here, a vector **bumper** with the signals of the 24

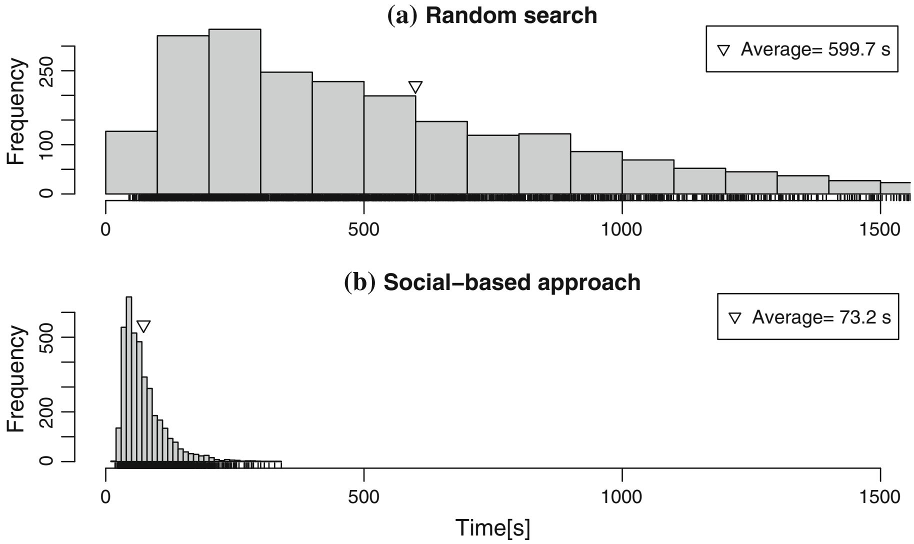


**Fig. 3** Histograms of the data collected on the two experiments. **a** Random search. **b** Social-based target localization

infra-red sensors, and a vector **A** compiling the angle $\alpha$ of each sensor. The vector **b** summarizing the activation of the 24 bumpers, and pointing in the direction where there is an obstacle[6] is calculated as follows:

$$b_x = \mathbf{bumper} \cdot \cos \mathbf{A} \tag{3}$$

$$b_y = \mathbf{bumper} \cdot \sin \mathbf{A} \tag{4}$$

Furthermore, in the case of the omni-directional camera some masks were applied to the image in order to eliminate the influence of walls when calculating the **t** and **nt** components.

### 3.3 Tests on simulation

This section shows the results of some simulated tests with two different configurations of the arena.

#### *3.3.1 Comparison with random search*

The results in this subsection were obtained by using the setup described in Sect. 3.1, with a square arena of 300 cm of side length, and by placing two targets in opposite corners of the arena at positions (40, 260) cm and (260, 40) cm. In order to test our simulation setup, we measured the time a robot spends in going from one target to the other by 1. using a random walking strategy, and 2. using our social-based approach. The random walking strategy was implemented by setting to zero the components **t** and **nt** of the steering vector. Figure 3 summarizes the measures we made for 1000 single trips in each case.

As it can be seen in Fig. 3, the time needed for finding a target is dramatically reduced when robots employ the social-based approach.

In order to characterize our approach, we performed a set of tests modifying the attraction between robots and we measured the time they spent in finding the targets. For doing so, we multiplied the component **t** of the steering vector in Eqs. 1 and 2 by a parameter of attraction $k$. The resulting steer vector is shown in Eqs. 5 and 6. By changing the attraction between robots we also modify the dynamics of the population, making the navigation less or more fluid. We also explored the efficiency of our approach under the presence of different amounts of robots performing the task of searching within the population. The results of these tests are depicted in Fig. 4. The population size was always kept constant.

$$S_x = r_x + b_x + f_x + k \cdot t_x + nt_x \tag{5}$$

$$S_y = r_y + b_y + f_y + k \cdot t_y + nt_y \tag{6}$$

As it can be seen in Fig. 4, the parameter $k$ can be used to control the speed of the robots and thus to control the time the robots spend in finding a target. From the same figure, it can also be observed that the average time for finding a target is lower when few robots are searching targets.

#### *3.3.2 Comparison with a GPS-like controller with similar complexity*

The results in this section were obtained by using the slightly modified arena shown in Fig. 5a. This arena has a rectangular form of 300 cm × 400 cm side length, and two targets located at positions (60, 40) cm and (340, 40) cm.

The goal of these experiments is to compare our social approach against another method driving the robots to the

[6] If there are several obstacles this method returns the direction where there are more detections.

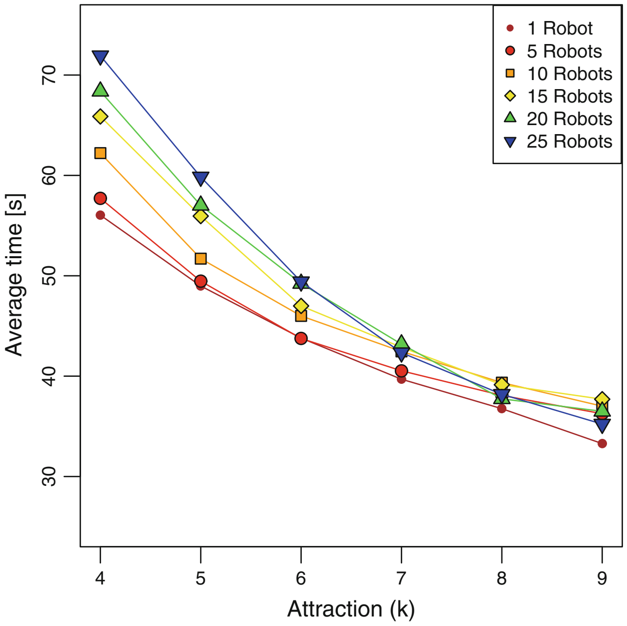


**Fig. 4** Average time spent by a robot with respect to the value of attraction in the case of using the information gathered by the population about the location of the targets. Each *point* in the image summarizes a set of 1000 single trips performed by the robots

targets with a similar complexity. Maps and global coordination strategies are thus not allowed.

Given that in the social approach each robot relies on the direction to move, we decided to compare that against a GPS-like approach where each robot knows the location of the targets and moves in that direction. Therefore, we replaced the components **t** and **nt** of the steering vector by a single vector **a** always pointing in the direction of the desired target. The resulting steering formulae are shown in Eqs. 7 and 8.

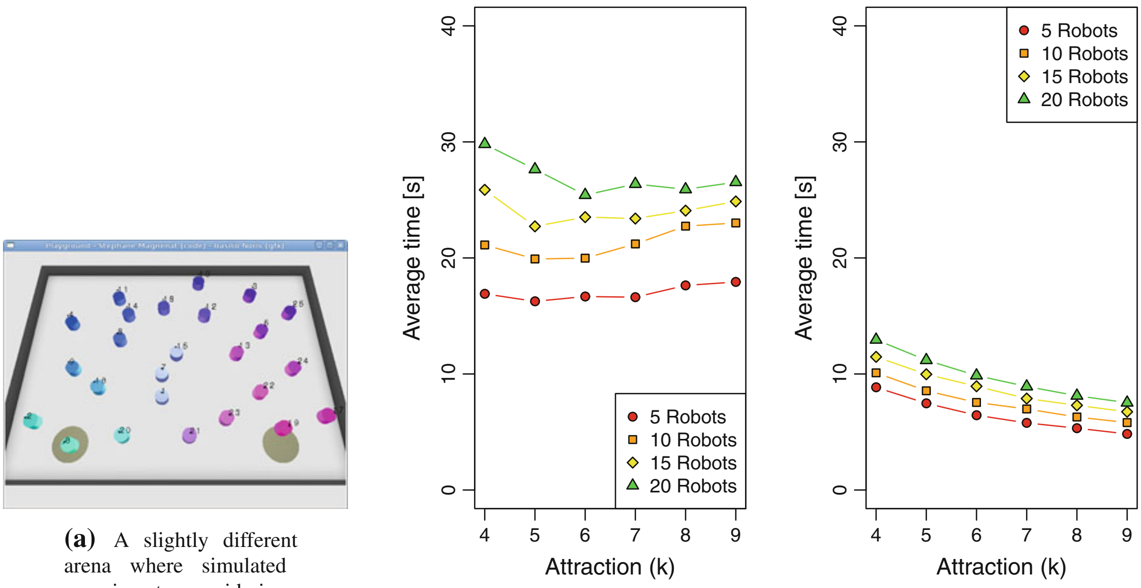
$$S_x = r_x + b_x + f_x + a_x \tag{7}$$

$$S_y = r_y + b_y + f_y + a_y \tag{8}$$

Figure 5 shows the average time a robot spends in going from one target to the other for both strategies, different values of $k$, and different amounts of robots searching at the same time. After comparing Fig. 5b, c it can be seen that average time in the case of the GPS-like strategy is lower that in the case of the social strategy. This result can be seen as obvious given that there are no obstacles in the trajectory of the robots, and thus knowing the position of the target gives a major advantage to the GPS-like approach.

We added an obstacle between the two targets to make this comparison more interesting. In the presence of an obstacle robots have to first avoid the obstacle in order to get to the target. With this experiment we want to observe how the robots behave with these two simple strategies when obstacles of different lengths are placed in the arena. Figure 6 shows the obstacles of length 50, 100, 150 and 200 cm that were put in the arena to perform the tests.

Figure 7 shows the average time for robots using the social approach when going from one target to the other with different values of $k$, different amounts of robots searching at the same time, and different lengths of obstacles. As it can

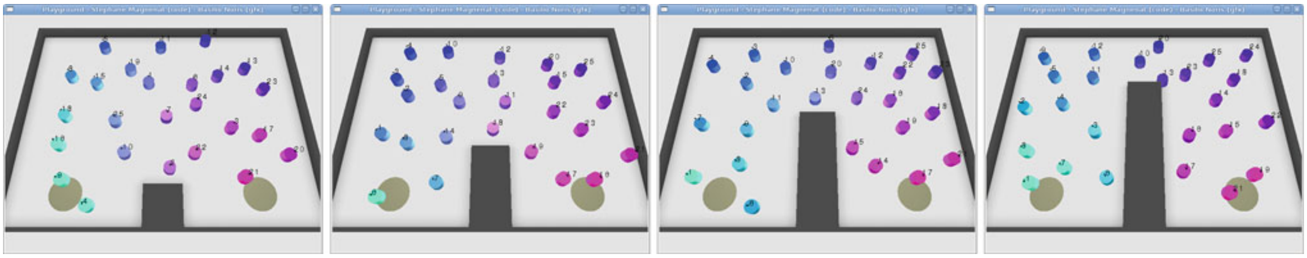


**(a)** A slightly different arena where simulated experiments considering an obstacle evolved.

**(b)** Average time of the social strategy without obstacle.

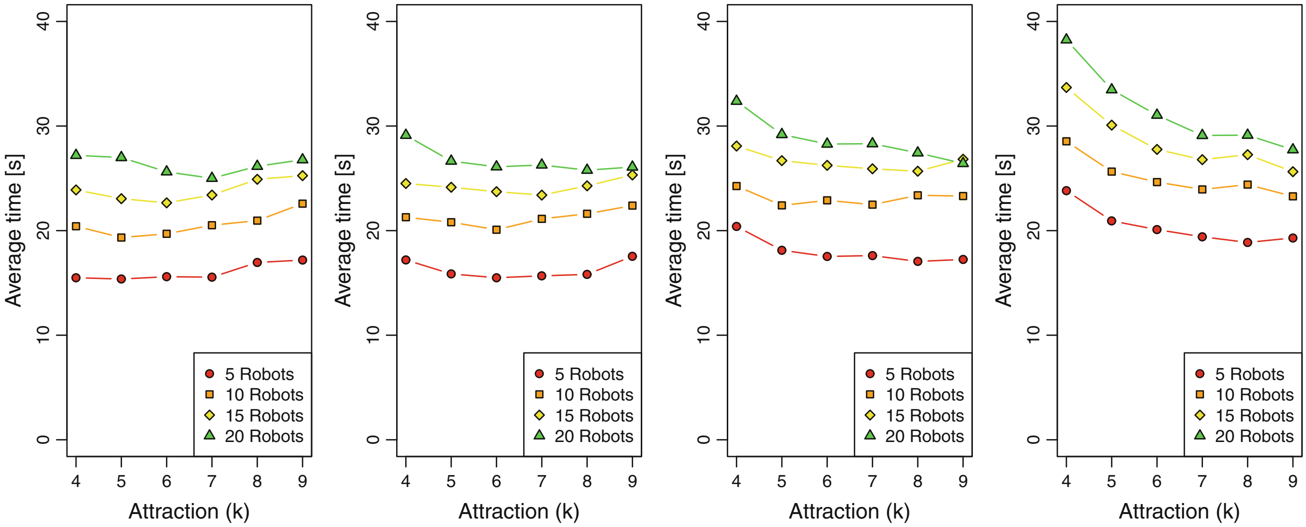
**(c)** Average time of the GPS-like strategy without obstacle.

**Fig. 5** **a** Tag 1 is placed in the *bottom-left* corner, and tag 2 is placed in the *bottom-right* corner. **b**, **c** Average time a searching robot spends in going from one target to the other without the presence of obstacles in the environment. Different values of $k$ and different amounts of robots searching for targets at the same time were tested



**(a)** Obstacle of 50 cm. **(b)** Obstacle of 100 cm. **(c)** Obstacle of 150 cm. **(d)** Obstacle of 200 cm.

**Fig. 6** Four different lengths of obstacles in the modified arena



**(a)** Obstacle of 50 cm. **(b)** Obstacle of 100 cm. **(c)** Obstacle of 150 cm. **(d)** Obstacle of 200 cm.

**Fig. 7** Average time for robots using the social approach for target localization. We tested different values of $k$, different amounts of robots performing the searching task at the same time, and different lengths of obstacle between the targets

be seen from that figure, the average time increases with the length of the obstacle between the targets. This result is coherent since robots have to transit longer paths when longer obstacles are put between the targets.

Figure 8 shows the average time for robots using the GPS-like approach when going from one target to the other with different values of $k$, different amounts of robots searching at the same time, and different lengths of obstacles. As in the previous case, the average navigation time is positively affected by the length of the obstacle between the target. However, the GPS-like approach seems to be more influenced by the presence of big obstacles than the social approach.

We pushed even further the comparison between both approaches by putting an very large obstacle which covers almost the whole arena letting just a little corridor for the robots to pass through. Figures 7d and 8d show the results of this last comparison. As it can be seen in the figure, the social approach manages to drive the robots to the targets even when operating in an environment which has more the form of a labyrinth than an open space. Instead, with the GPS-like approach, robots spent much more time to get to the targets in this environment.

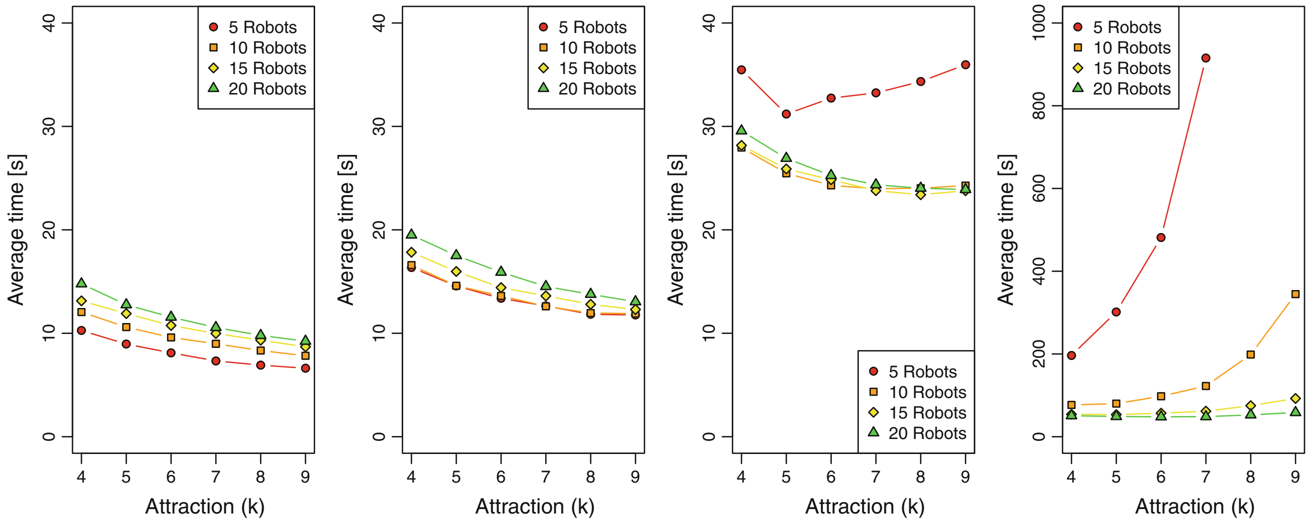
## 4 Validation on the marXbot robot

A set of the above described experiments was implemented on a real robotic platform in order to validate our approach handling real physical constraints. For doing so, we used a collective robotic set-up with 20 robots communicating their state through visual signals: emitted by RGB LEDs and perceived by omni-directional cameras. This section describes the robotic platform and the experimental set-up.

### 4.1 The marXbot robot

The marXbot [6,7] is a small mobile robot[7] designed at the EPFL with the support of the FP6 European project PERPLEXUS. This robot offers a reliable hardware platform to perform experiments in collective robotics. It has

[7] 17 cm diameter, 17 cm height, ∼1000 g weight.



**(a)** Obstacle of 50 cm. **(b)** Obstacle of 100 cm. **(c)** Obstacle of 150 cm. **(d)** Obstacle of 200 cm.

**Fig. 8** Average time for robots using the GPS-like approach for target localization. We tested different values of *k*, different amounts of robots performing the searching task at the same time, and different lengths of obstacle between the targets

a modular hardware and software architecture with a real operative system which allows high flexibility and robustness. The ground mobility is achieved by a combination of tracks and wheels that gives good mobility in even and uneven terrains. About 60 sensors ensure a perception of the environment sufficient to perform a large variety of tasks and a collaboration among several individuals. Moreover, the marXbot robot is modular to allow different configurations depending on the task to be achieved.

The robot structure is based on a modular concept at all levels: mechanical, electronics and software. Mechanical modularity is achieved by stacking modules on top of another following a well-defined mechanical specification. Figure 9b shows the modules composing one of the basic configurations of the marXbot. From Fig. 9b one can distinguish a common element: two central electrical connectors and four fixations holes around them ensuring the mechanical and electrical interface between all the modules. The modularity of the electronics is achieved by:

- Sharing, on the extension connectors:
  - The raw battery power supply.
  - Some control signals such as the power enable and the reset signal.
  - Two communication buses: CAN and $I^2C$.
- Selecting different functionalities for the several modules in such a way that they result as independent as possible.
- Providing each module with its own local processing power.
- Supporting on each module the ASEBA architecture described in [20].

Figure 9a shows a picture of the actual robot in one of its possible configurations with five modules.

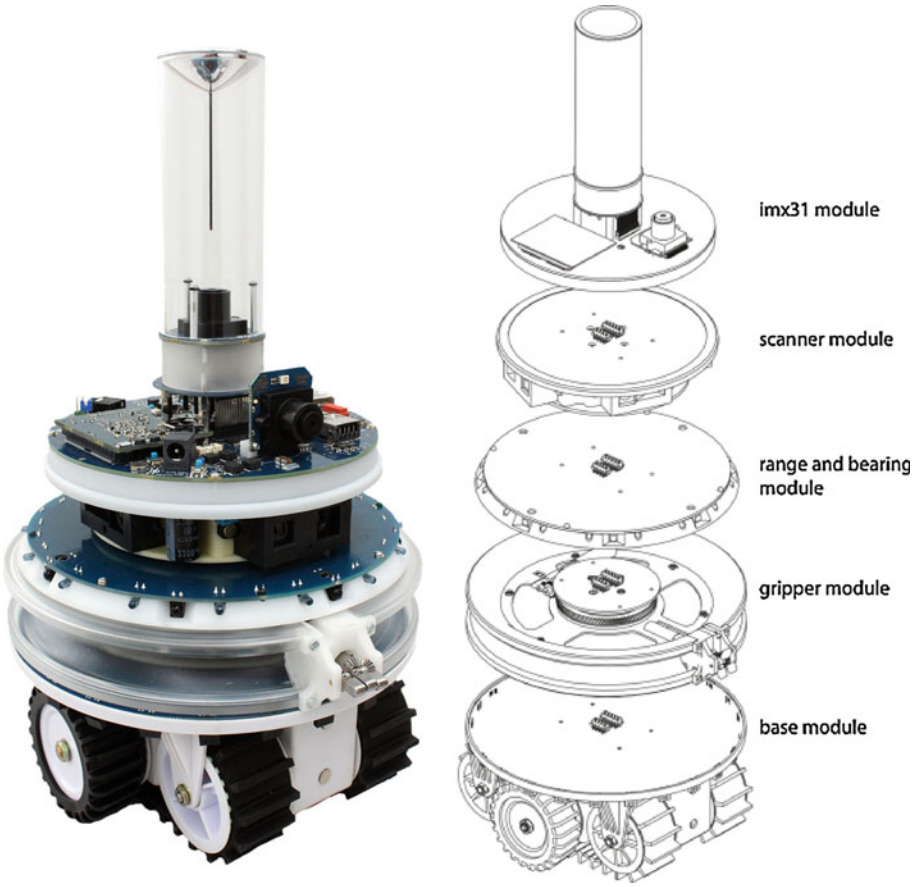
The base module contains the power supply of the whole robot. The marXbot is powered by a 3.7 V, 10 Ah Lithium-Polymer battery which is hot-swappable. The hot-swapping capability is provided by a super-capacitor which maintains the power supply of the robot for ∼10 s during battery exchange. The casing in the base module embeds two individual battery cells, the slipping contacts for hot-swapping, and the protection electronics. This electronics provides charging and over-current protection, but also monitoring, allowing the robots to know the current battery capacity and the number of recharge cycles it went through.

Moreover, the base module contains two tracks and two wheels that give mobility to the robot. This combination of tracks and wheels is called *treels*. Treels are powered by two 2 W motor, each one associated to a rubber track and a wheel. These motors are driven by dedicated electronic boards situated on each side of the battery (one for each motor). With this configuration, the marXbot can reach a maximum speed of ∼30 cm/s.

The base of the robot also provides some sensing capabilities. There are infra-red sensors distributed around the robot on the main printed circuit which act as virtual bumpers and ground detectors. Those sensors have a range of some centimetres, 24 are directed outside the robot and 8 are directed to the ground. In addition, 4 contact ground sensors are placed under the lowest part of the robot and mounted on the two vertical printed circuits.

The base of the marXbot also embeds a RFID reader and writer with an antenna situated on the bottom of the robot, close to the ground. This reader can read all types of RFID

**Fig. 9** The marXbot in a configuration including five modules



**(a)** Actual image of the marXbot.

**(b)** Exploded CAD view of the marXbot.

chips in the HF 13.56 MHz range. By using this reader, RFID chips can be placed on the ground to identify specific zones.

Finally, to better perceive the orientation of the robot body in all-terrain conditions, the marXbot base includes 3 axis accelerometers and gyroscopes.

The top module includes two cameras, a LED beacon, an imx.31 processor and some peripherals, such as a WiFi board and a flash card reader. The two cameras are mounted in different configurations: a front camera and an omni-directional camera on top of the board. This last specific feature makes that this module can only be placed on the top of the robot. The imx.31 processor runs LINUX and access standard peripherals such as WiFi, USB or flash storage.

## 4.2 Experimentation setup

Our implementation makes use of a population of 20 marXbots on an arena. We employed the RGB led and the omni-directional camera to implement the *state communication* (see Sect. 2) between robots.

### *4.2.1 The arena*

The arena we employed for our experiments includes:

- A working area of 6.4 m × 4 m, 2.8 m high. Only a portion of this working area has been used for the experiments, as illustrated in Fig. 10.
- White walls made of highly reflective tissue for optimal diffusion of light.
- An homogeneous lightning system with a spectrum with low infra-red components.
- One black/white 5 Mpixels camera placed on the ceiling to track the position of the robots.
- Tracking software based on the `fidtrack` library [5]. In order to track a robot, a fiducial was placed over the omni-directional camera on top of the robot.

Figure 10 shows 20 marXbot robots on the arena during our experiments. We distributed the robots in a square of 2 m of side length, and we placed some RFID chips in opposite corners of the arena. These RFID tags represent the targets that the robots have to search.

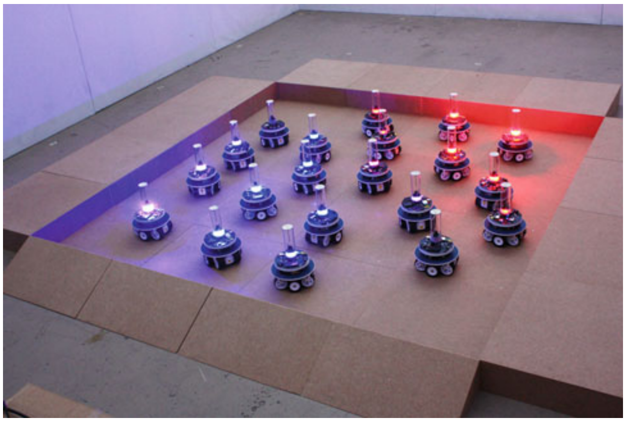


**Fig. 10** Example of arena configuration with twenty marXbot robots

### 4.3 Implementation of the social approach on the marXbot

We tried to keep both implementations of our approach (i.e., in the marXbot as well as in the case of simulation) as accurate as possible. However, the implementation in the actual robot does not match perfectly the simulation. The idea was not to simulate every detail of the robot but to exhibit the same high-level behaviour in both cases. For this reason, some constants we employed in our algorithms are not the same.

#### *4.3.1 Detection of targets*

The marXbot robot has a RFID reader which we employed to detect the target areas. Once the robot is located over one of the RFID chips, it changes the colour of its LED beacon in order to inform the other members of the population about the location of the target. If a robot detects the target number 1, it sets the *blue*[8] component of its colour to 100%, instead, if a robot detects the target number 2, it sets the *red* component of its colour to 100%.

#### *4.3.2 Omni-directional camera*

Colour calibration had to be done before processing the images from the camera since we decided to propagate the information through the population by using the RGB components of the LED beacon. In our experimental set-up, robots share information by encoding it in the value of the R and B components of the colour of their LED beacon. Robots receiving this information have to decode the colour by extracting the right $R$ and $B$ components that were sent. After some preliminary tests we found that the $R'G'B'$ components detected by the camera do not keep the same proportions than the components sent by the LED. Instead, if a sender robot sets the red component of its LED beacon to 100%, the receiver robots detect a blob with the three components $RGB$ (i.e., components $G$ and $B$ are present even if the sender does not send them). In order to get the original values sent by the sender, receiving robots have to make a transformation of the three received components consisting in projecting the received components into the actual $R'G'B'$ axes of the camera. Figure 11(before transformation) shows the intensity of the components detected by the camera when a sender robot emits each one of the $RGB$ components individually.

After some algebraic manipulations, we managed to find the $R'G'B'$ axes of the camera in which each one of the detected components have to be projected in order to detect the original $RGB$ components that were sent. Equations 9, 10 and 11 show how each one of these components were obtained.

$$R = 1.09r - 0.26g - 0.02b \tag{9}$$

$$G = -0.24r + 1.19g - 0.16b \tag{10}$$

$$B = -0.06r - 0.33g + 1.06b \tag{11}$$

This calibration process was done once, before starting the experiments. Figure 11 shows the resulting components after the transformation.

Moreover, the images obtained from the omni-directional camera have to be analysed in order to extract the location and the colours of neighbouring robots. The goal of processing these images is to obtain the direction where there are other robots showing the red and blue components, and to detect the most predominant colours in the proximities in order to copy them. Figure 12 shows an example of an image obtained with the omni-directional camera. In order to ease the task of detecting the LED beacons of other robots, we reduced the integration time of the camera in a way that only the robots showing colours (i.e., LED beacon turned on) could be visible. The algorithm for image processing performs two steps:

1. Blob detection. The first step after capturing the image from the omni-directional camera is to detect the neighbouring robots. In order to do so, we performed a blob detection on the image and we compute the position of blobs of light which represent the LED beacons of robots.
2. Transformation of the $R'G'B'$ components. Once the location of neighbouring robots is detected, we transform their colour as it was aforementioned in order to find the $RGB$ components that were sent. Then, we keep

[8] In the case of the real robot we employed the blue component instead of the green component we used in simulation. The reason for this choice is related to the camera of the marXbot which detects better the red and blue components.

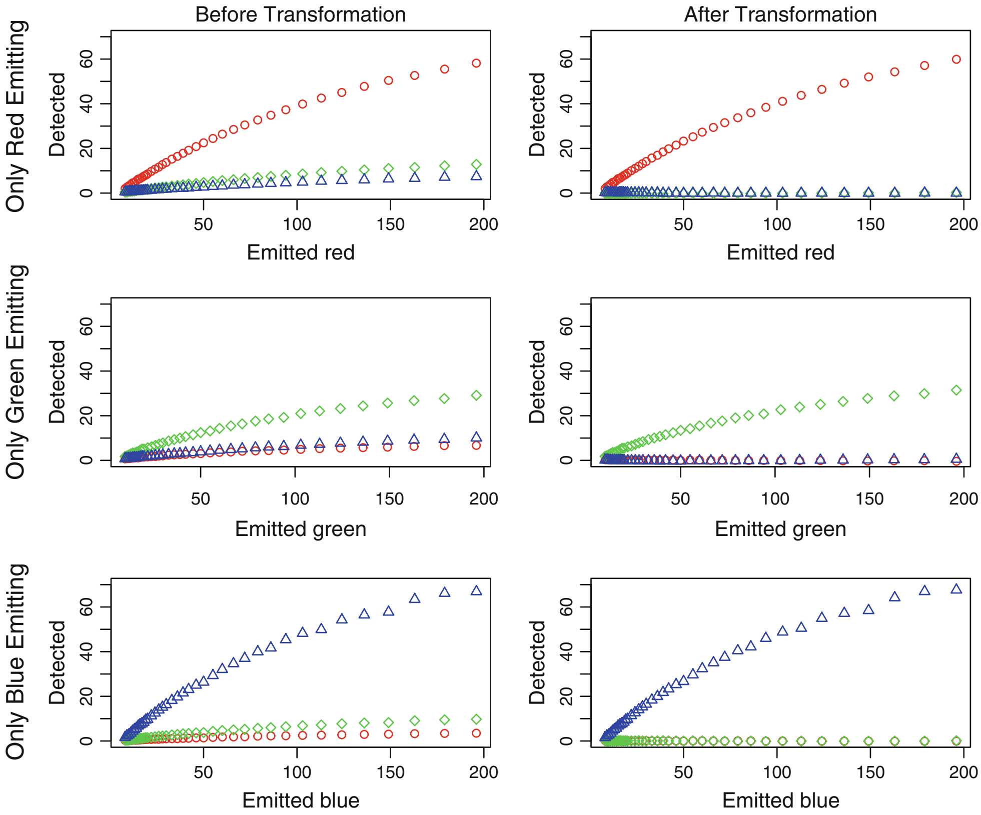


Fig. 11 Each one of the three components of colour detected by the omni-directional camera, before and after transformation

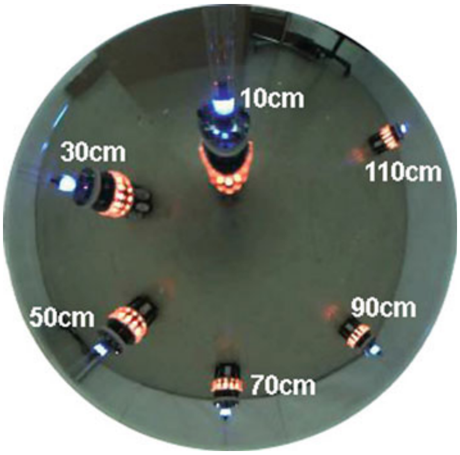


Fig. 12 Example of an image captured with the omni-directional camera. LED beacons are *blue* for all robots

the maximum value of the detected components in order to emit this colour by using the LED beacon.[9]

Note that the use of the RGB code may limit the number of targets to search to a maximum of 3, since the signal used for coding the estimation of the distance to each target must be orthogonal respective to the signals used for other targets. However, the use of colours for state transmission can be considered as a proof of concept of a larger approach. If we consider the use of RF signals with N different frequencies, each frequency can be used for coding the estimation for each target, providing the possibility of guiding robots towards N different targets, accepting that the chosen RF frequencies will be orthogonal and that there will not be harmonics interference between them.

### *4.3.3 Navigation*

As in the case of simulation, navigation was implemented in a very simple reactive manner. We computed the direction to move by using the same approach showed in Eqs. 1 and 2 in Sect. 3.2.2. Component **b**[10] of the steering vector was computed by using the lectures from virtual bumpers, and component **t**[11] of the steering vector was calculated by using information from the omni-directional camera. In order to go to the target while avoiding obstacles the final steering

[9] No attenuation of the components was done as it was done in the case of simulation. In the experiments with the actual marXbot this attenuation is intrinsically done by the camera and the algorithm of blob detection.

[10] The vector pointing in the direction where there are no obstacles.

[11] The vector pointing in the direction where it is more likely to find a target.

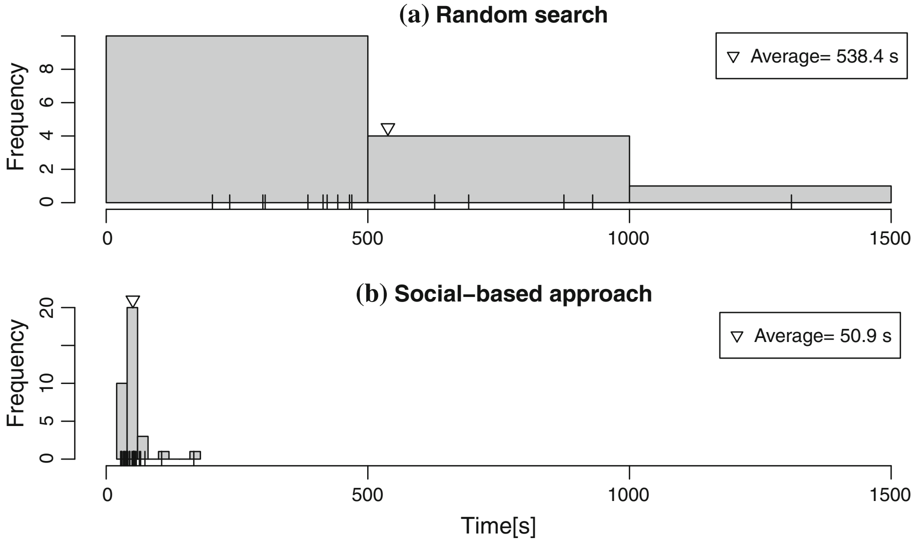


**Fig. 13** Histograms of the data collected from the two experiments made with the population of marXbots. **a** Random search. **b** Social-based target localization

vector is computed by adding vectors **b** and **t**, as shown in Eqs. 12 and 13.

$$S_x = r_x + b_x + t_x \tag{12}$$

$$S_y = r_y + b_y + t_y \tag{13}$$

### 4.4 Tests on a population of marXbots

We implemented the social-based approach in a population of 20 marXbots evolving on the arena described in Sect. 4.2.1. Figure 13 shows the resulting single-trip times after 30 minutes of collecting data.

A direct quantitative comparison of the results obtained in simulation (Fig. 3) and in the real robot (Fig. 13) may make few sense since real physical constraints are always very difficult to model in an accurate manner. However, we can use both figures in order to establish interesting qualitative equivalences between simulated and real experiments. In both cases we can observe a Poisson-like distribution for both random and social based approach. In the real robot implementation the distribution may become less evident because of the few amount of runs which are due to the practical difficulty of performing lots of real measures, but the representation of the results of the different runs (at the bottom of the distribution) helps to guess the response trend. Another interesting parallel is the performance comparison between the social and random approaches; the socially driven robot can find the target around 100 faster than the random search in both cases: simulation and real implementation.

### 4.5 Tracking a robot

Moreover, we put a fiducial over one of the robots that was using the social-based approach in order to track its position during the experiment. The robot was tracked by using the camera on the ceiling of the arena as described in Sect. 4.2.1. Tracking the position of robots allowed us to have a better idea of the trajectories they made in order to get to the targets. Figure 14 shows a 2D histogram of the position of the tracked robot. In this picture, the color of each one of the boxes in the grid indicates the amount of times the robot was in that position. As it can be seen in the histogram, the searching robot has a strong tendency to go trough the diagonal of the arena, or in other words, to follow the gradient that was made by the other members of the population.

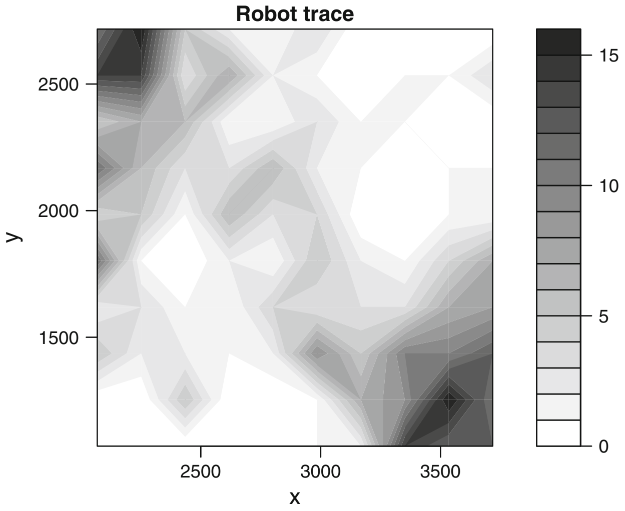


**Fig. 14** 2D histogram of the position of a searching marXbot. *Colours* indicate the amount of times the robot was found in [$x$, $y$] position

## 5 Conclusions

Robots performing a cooperative task in an unknown environment eventually need to find common target areas. Since these key zones are common for every robot, we proposed to guide the navigation of each robot by using the knowledge about the location of the targets that is disseminated in the whole population. The proposed coordination scheme is distributed and uses state communication in an intrinsic way, i.e., robots transmit information about their internal state, but they are not aware of whether other robots receive this information or not. This approach simplifies the communication and makes the system more robust. Moreover, the proposed strategy for target localization avoids the use of global positioning systems, that might be difficult to deploy in unknown or hostile environments, and avoids also the use of odometry, which is sensitive to cumulated errors after large running periods. Additionally, the fact of being a distributed scheme makes it very robust and scalable.

We used a robotic simulator to test our approach, and we verified the correct behaviour of the robots by measuring the time a robot spent in sequentially travelling between two targets. For comparing[12] purposes we also provide measures where, instead of using our social-based approach, the robots freely navigate in a random way. Moreover, some tests concerning robustness and scalability were performed. A parameter $k$ was added in order to modulate the attraction between robots when approaching a target. When we changed $k$ the average time a robot spends in finding a target changed and therefore, the trajectories performed by the robots become more or less smooth. The amount of robots looking for targets was also changed during simulations. We found that there is a weak relationship between robot performance and the amount of robots performing the task; and that this relationship is even weaker when parameter $k$ is higher. We also performed some comparisons against another approach for guiding the robots having a similar degree of complexity. Global coordination and maps of the arena were thus avoided. Therefore, we implemented a GPS-like approach were each robot knows the location of the target and uses this information to guide its navigation. We compared both strategies i.e., social and GPS-like, in an slightly modified arena with an obstacle between the targets. Our results showed that the social approach is more robust to the presence of longer obstacles in the environment.

We surmise that the advantages of the social approach rely on the fact that there are no fixed maps of the environment giving the position of the targets. Instead, this knowledge is distributed in the population, and is thus dynamically updated as the robots navigate through the environment. The whole population benefit from the local contributions of each member of it, which globally produces a gradient that is useful to guide any robot to a target. However, this intrinsic map of the environment is only present if there is a population of robots collaborating. A single robot (the worst case) is bound to navigate randomly through the environment without any indication of the location of the targets given by neighbouring robots.

Last but not least, we validated our social-based approach in a population of real robots. In our experiments we employed a mobile robot designed at EPFL called the marXbot. These robots have remarkable capabilities such as locomotion using tracks and wheels; a large set of sensors including an omni-directional camera and infra-red bumpers; and a RGB LED beacon that was used to disseminate the gradient of knowledge through the population. Even if both implementations (i.e., simulation and actual marXbot) of the strategy were not exactly the same, we corroborated that in both cases the behaviour exhibited by the population of robots remained the same, the performance curves kept the same trend, and that they are equivalent in terms of the time travelled by the robots when looking for a target.

**Acknowledgments** The authors would like to thank all the members of the Perplexus project for their valuable work, and their colleagues at the REDS and the MOBOTS groups for their support. This work is funded by the FET programme IST-STREP of the European Community, under grant IST-034632 (PERPLEXUS).

## References

1. Arkin RC (1992) Cooperation without communication: multiagent schema-based robot navigation. J Robot Syst 9(3):351–364
2. Arkin RC, Bekey GA (eds) (1997) Robot colonies. Kluwer, Norwell
3. Balch T, Arkin RC (1994) Communication in reactive multiagent robotic systems. Auton Robots 1(1):27–52
4. Bekey GA (2005) Autonomous robots: from biological inspiration to implementation and control (Intelligent Robotics and Autonomous Agents). The MIT Press, Cambridge
5. Bencina R, Kaltenbrunner M, Jorda S (2005) Improved topological fiducial tracking in the reactivision system. In: IEEE computer society conference on computer vision and pattern recognition-workshops, 2005. CVPR workshops. IEEE, p 99
6. Bonani M, Baaboura T, Retornaz P, Vaussard F, Magnenat S, Burnier D, Longchamp V, Mondada F (2009) The marxbot—a modular all-terrain experimentation robot. http://mobots.epfl.ch/marxbot.html
7. Bonani M, Longchamp V, Magnenat S, Rtornaz P, Burnier D, Roulet G, Vaussard F, Bleuler H, Mondada F (2010) The MarXbot, a miniature mobile robot opening new perspectives for the collective-robotic research. In: 2010 IEEE/RSJ international conference on intelligent robots and systems (IROS 2010). http://mobots.epfl.ch/
8. Braitenberg V (1984) Vehicles: experiments in synthetic psychology. The MIT Press, Cambridge
9. Camazine S, Crailsheim K, Hrassnigg N, Robinson GE, Leonhard B, Kropiunigg H (1998) Protein trophallaxis and the regulation of pollen foraging by honey bees (*Apis mellifera* L.). Apidologie 29(1–2):113–126
10. Camazine S, Deneubourg JL, Franks NR, Sneyd J, Theraulaz G, Bonabeau E (2001) Self-organization in biological systems. Princeton University Press, Princeton

[12] A more detailed comparison can be found in [22].

11. Campo A, Gutiérrez A, Nouyan S, Pinciroli C, Longchamp V, Garnier S, Dorigo M (2010) Artificial pheromone for path selection by a foraging swarm of robots. Biol Cybernet 103:339–352. doi:10.1007/s00422-010-0402-x
12. Deneubourg JL, Goss S (1989) Collective patterns and decision making. Ethol Ecol Evol 1:295–311
13. Floreano D, Mattiussi C (2008) Bio-inspired artificial intelligence: theories, methods, and technologies. The MIT Press, Cambridge
14. Gadagkar R (1997) Survival strategies: cooperation and conflict in animal societies. Harvard University Press, USA
15. Ijspeert A, Martinoli A, Billard A, Gambardella LM (2001) Collaboration through the exploitation of local interactions in autonomous collective robotics: the stick pulling experiment. Auton Robots 11(2):149–171
16. Kuniyoshi Y, Kita N, Rougeaux S, Sakane S, Ishii M, Kakikua M (1994) Cooperation by observation: the framework and basic task patterns. In: IEEE international conference on robotics and automation, 1994. Proceedings 1994, vol 1, pp 767–774
17. Kuniyoshi Y, Rickki J, Ishii M, Rougeaux S, Kita N, Sakane S, Kakikura M (1994) Vision-based behaviors for multi-robot cooperation. In: Proceedings of the IEEE/RSJ/GI international conference on intelligent robots and systems '94, vol 2. 'Advanced robotic systems and the real world', IROS '94, pp 925–932
18. Lambrinos D, Roggendorf T, Pfeifer R (2001) Insect strategies of visual homing in mobile robots. In: Biorobotics—methods and applications. AAAI Press, pp 37–66
19. Magnenat S, Waibel M, Beyeler A (2009) Enki—an open source fast 2d robot simulator. http://home.gna.org/enki/
20. Magnenat S, Rtornaz P, Bonani M, Longchamp V, Mondada F (2010) ASEBA: a modular architecture for event-based control of complex robots. IEEE/ASME transactions on mechatronics PP(99):1–9. doi:10.1109/TMECH.2010.2042722. http://www.ieee-asme-mechatronics.org/
21. Nouyan S, Gross R, Dorigo M, Bonani M, Mondada F (2005) Group transport along a robot chain in a self-organised robot colony. In: Proceedings of the 9th international conference on intelligent autonomous systems, IOS. IOS Press, pp 433–442
22. Satizábal HF, Upegui A, Pérez-Uribe A (2010) Social target localization in a population of foragers. In: González JR, Pelta DA, Cruz C, Terrazas G, Krasnogor N (eds) Studies in computational intelligence, vol 284. NICSO. Springer, Berlin, pp 13–24
23. Schmickl T, Crailsheim K (2006) Trophallaxis among swarm-robots: a biologically inspired strategy for swarm robotics. In: The first IEEE/RAS-EMBS international conference on biomedical robotics and biomechatronics, 2006. BioRob 2006, pp 377–382. doi:10.1109/BIOROB.2006.1639116
24. Smith JM, Szathmary E (2000) The origins of life: from the birth of life to the origin of language. Oxford University Press, USA
25. Sugawara K, Kazama T, Watanabe T (2004) Foraging behavior of interacting robots with virtual pheromone. In: Proceedings 2004 IEEE/RSJ international conference on intelligent robots and systems, 2004 (IROS 2004), vol 3, pp 3074–3079. doi:10.1109/IROS.2004.1389878
26. Werger B, Mataric MJ (1996) Robotic "food" chains: externalization of state and program for minimal-agent foraging. In: Proceedings of 4th internationl conference simulation of adaptive behavior: from animals to animats, vol 4. The MIT Press, pp 625–634
27. Winfield A (2009) Towards an engineering science of robot foraging. Distrib Auton Robot Syst 8:185–192