An Approach to Flocking of Robots Using Minimal Local Sensing and Common Orientation

Iñaki Navarro[1], Álvaro Gutiérrez[2], Fernando Matía[1], and Félix Monasterio-Huelin[2]

[1] Intelligent Control Group, Universidad Politécnica de Madrid,
José Gutiérrez Abascal 2, E-28006 Madrid, Spain
{inaki.navarro,fernando.matia}@upm.es
[2] Departamento de Tecnologías Especiales Aplicadas a la Telecomunicación,
Universidad Politécnica de Madrid,
Avd. Ciudad Universitaria s/n, E-28040 Madrid, Spain
aguti@etsit.upm.es,felix.monasteriohuelin@upm.es

Abstract. A new algorithm for the control of robot flocking is presented. Flocks of mobile robots are created by the use of local control rules in a fully distributed way, using just local information from simple infra-red sensors and global heading information on each robot. Experiments were done to test the algorithm, yielding results in which robots behaved as expected, moving at a reasonable velocity and in a cohesive way. Up to seven robots were used in real experiments and up to fifty in simulation.

Keywords: Distributed Robot Systems, Robot Flocking, Mobile Robot.

1 Introduction

There are many applications in which a multi-robot approach is an advantage compared to single robot systems. Groups of robots moving together can act as sensor arrays, allowing them to locate a desired source in a more effective way. Thus, formations of robots can be very useful in search tasks, especially when spatial pattern of the source is complex, like in odor [1] or sound [2] cases. They are also useful in mapping tasks [3], since measurement redundancy allows robots to build more accurate maps.

One of the basic problems in multi-robot systems is how to make mobile robots move together as a group, behaving as a single entity. This problem is solved by flocking and formations of robots. In flocking problem robots move as a group but the shape and relative positions between the robots are not fixed, allowing robots to move within the group. The first work about artificial flocking was a computer graphic animation of a group of birds [4]. Some characteristic examples of robot flocking in which a theoretical effort is made are [5, 6], while [7] is focused on experiments with real robots. On the other hand, a robot formation can be defined as a group of robots that is able to maintain pre-determined positions and orientations between its members at the same time that it moves as a whole [8, 9].

E. Corchado, A. Abraham, and W. Pedrycz (Eds.): HAIS 2008, LNAI 5271, pp. 616–624, 2008.

A desirable characteristic that flocking of mobile robots may have is scalability in the number of robots. In order to have this scalability, local sensing and communications and a decentralized controller are necessary [10].

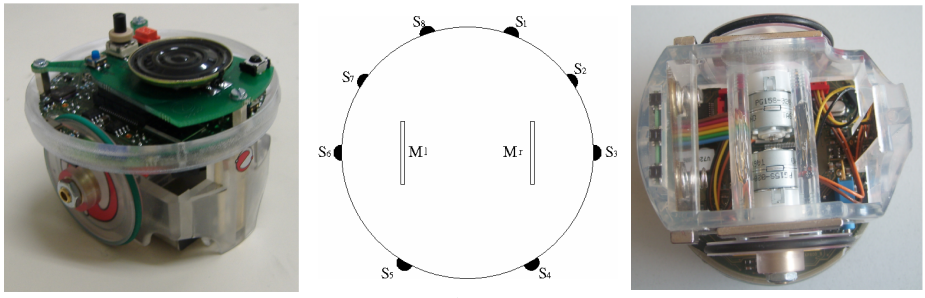
In this paper we propose a distributed and scalable algorithm for the control of mobile robots flocking that uses very simple proximity sensors and information about their own absolute headings. The main advantages are the simplicity of the sensors required for the flocking and the scalability of the algorithm. The platform used to test the designed algorithm is described in Sect. 2. In Sect. 3, the proposed algorithm is explained. Experiments performed to test the algorithm and their results are described and discussed in Sect. 4. Finally the conclusions and future work can be found in Sect. 5.

2 Experimental Test Platform

The proposed algorithm uses local sensing to detect the distance and angle between nearby robots, and information about its own global heading. Thus, the *e-puck* robots, used to test the algorithm, need to have these capabilities. They are small wheeled cylindrical robots, 7*cm* of diameter, equipped with 8 infra-red proximity sensors distributed around their bodies, 3 infra-red ground sensors and a 3-axis accelerometer. Their mobility is ensured by a differential drive system.

The infra-red sensors are used to estimate the distance to the neighboring robots, while the angle to the nearby robot is approximated by the direction of the infra-red sensor. Every obstacle seen by each sensor is considered as a neighboring robot, so no real obstacles are placed or considered in the experiments. The range of the infra-red sensors is about 12*cm*. A picture of the *e-puck* robot and the distribution of its infra-red sensors can be seen in Fig. 1.

In order to know its heading in a global coordinate system, robots move in a vertical plane, using accelerometer sensors to obtain a global orientation. A magnetic cubic extension is added to the bottom of the robots, permitting the robots to move attached to a metallic wall. A picture of the robot with the magnets is shown in Fig. 1c. This modification allows us to design a virtual compass using the 3-axis



(a) (b) (c)

Fig. 1. (a) The *e-puck* robot. (b) Distribution of the infra-red sensors. (c) Magnets placed at the bottom of the robot.

accelerometer thanks to the gravity force. Robots sense the gravity on x and y accelerometer's axis, giving $\tan(y/x)$ the global heading. A preliminary calibration is needed on every robot to overcome with the accelerometer bias.

Communication between robots is necessary in some parts of the algorithm. It is implemented through bluetooth and a central computer. *E-pucks* transmit information to a computer that is redirected to the robots

Webots simulator [11] was used initially to test the algorithm, using a realistic model of the robots. It allowed us to perform the experiments in a fast way, tuning the different parameters easily, and using up to 50 robots. A compass was added to the *e-puck* model in order to know the own heading in global coordinates.

3 Algorithm

The developed algorithm is fully distributed among the robots, allowing the robots to move as a group in a common pre-defined direction, by just using local information from the infra-red proximity sensors and gobal orientation.

Each robot reacts to every object detected by its infra-red sensors, being attracted or repelled depending on the measured distance. This makes that the robots try to maintain a desired distance between them. Each robot generates a virtual velocity $\boldsymbol{V}_{aggregation}$ as the sum of the reactions to nearby robots:

$$V_{aggregation} = \sum V_i \tag{1}$$

The magnitude and angle of $\boldsymbol{V}_i$ are defined as follows:

$$|V_i| = \begin{cases} K_1(desiredDist - dist_i), & if\ dist_i \le desiredDist \\ K_2(dist_i - desiredDist), & if\ desiredDist < dist_i \le \max Dist \\ 0, & if\ \max Dist < distSensor_i \end{cases} \tag{2}$$

$$\arg(V_i) = \begin{cases} angle_i, & if\ dist_i \le desiredDist \\ angle_i + \pi, & if\ desiredDist < dist_i \le \max Dist \\ 0, & if\ \max Dist < distSensor_i \end{cases} \tag{3}$$

where *desiredDist* is the desired distance that robots are supposed to maintain between them; $dist_i$ is the measured distance by a sensor; *maxDist* is a threshold indicating that every measure above should not be considered as a robot; and $angle_i$ is the position in radians of the sensor, that represents an approximation of the direction of the detected nearby robot. K_1 and K_2 are the adjusting parameters of the proportional controller.

In order to move in the pre-defined desired direction, each robot reacts generating another desired virtual velocity $V_{desiredDirection}$, defined by its magnitude and angle as follows:

$$|V_{desiredDirection}| = K_3 \tag{4}$$

$$\arg(V_{desiredDirection}) = desiredDirection - myHeading \tag{5}$$

where *desiredDirection* is the desired common direction of movement, and *myHeading* is the robot heading expressed in the same coordinate system as *desiredDirection*.

As a result from this virtual velocity $\boldsymbol{V}_{desiredDirection}$, robots would move in the same direction and approximately at the same speed. In order to make the robots to move together as a group in the same direction and maintaining the desired distance between them, both virtual velocities are added resulting in the final total virtual velocity:

$$V_{total=}V_{aggregation} + V_{desiredDirection} \tag{6}$$

Since robots used in the experiments are not purely holonomic, the total virtual velocity must be translated in the appropriate motor speeds, approximating the desired velocity by making use of the *Low Level Controller* described in next the section.

3.1 Low Level Controller

The *Low Level Controller* (LLC) is designed to translate the virtual velocity in mobile robots with differential drive configuration. The controller is inspired by a similar one described by Hsu [12], but allowing backwards movement. The angle (θ) and magnitude ($|\mathbf{V}|$) of the virtual velocity are the inputs for getting the angular and linear velocities:

$$V_{linear} = K_4 |V| \cos(\theta) \tag{7}$$

$$V_{angular} = \begin{cases} K_5(\theta + \pi), & if\ \theta < -\pi/2 \\ K_5\theta, & if\ \pi/2 > \theta > -\pi/2 \\ K_5(\theta - \pi), & if\ \theta > \pi/2 \end{cases} \tag{8}$$

From (7) it can be seen that the linear velocity (V_{linear}) is proportional to the magnitude of the virtual velocity, and it is multiplied by $cos(\theta)$. This makes velocity maximum when $\theta = 0$ or $\theta = -\pi/2$, and minimum ($V_{linear} = 0$) when $\theta = \pi/2$ or $\theta = -\pi/2$. This is natural since a desired movement towards $\theta = \pi/2$ or $\theta = -\pi/2$ is not possible because of the kinematics of the robot. The robot will move forwards when $-\pi/2 > \theta > \pi/2$ and backwards otherwise. The angular velocity ($V_{angular}$) is proportional to θ for $-\pi/2 > \theta > \pi/2$, when moving forwards, and proportional to $\theta + \pi$ when moving backwards. Thus, the robot will reach the desired heading that depends on if it is moving forward or backwards.

The sum of the linear and angular velocities is translated to motor speeds taking into account the kinematics of differential drive robot as follows:

$$s_{motor-right} = V_{linear} + B * V_{angular} \tag{9}$$

$$s_{motor-left} = V_{linear} - B * V_{angular} \tag{10}$$

where B is half the distance between the two wheels, $s_{motor\text{-}right}$ the speed of the right motor and $s_{motor\text{-}left}$ the speed of the left motor.

By applying the LLC to $\boldsymbol{V}_{total}$ in all the robots, it results in a flocking of the robots towards the pre-defined direction.

3.2 Search for Lost Flock Algorithm

Since robots' sensor range is just 12*cm*, eventually a robot may stop detecting any neighboring robot and might not follow the flock. In order to overcome with that problem, a simple algorithm to look for the group of robots has been designed and implemented.

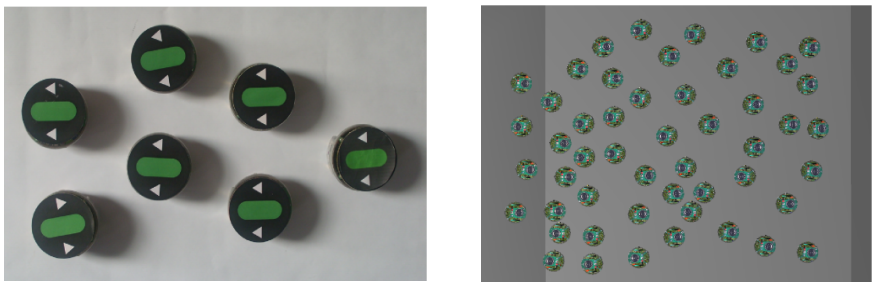
When a robot loses the flock it first orientates in the direction of the last seen robot. After that it moves during few seconds in that direction. If the flock is still not found, the lost robot moves in the direction that the flock is moving, this is, *desiredDirection*. If after a certain time the flock is not found the robot consider itself as completely lost and stops. When the robot finds the flock it quits the searching algorithm and continues with the general control algorithm.

This recovery algorithm works quite well, usually during the first seconds the robot finds the flock, partially because it moves 50% faster than the flock.

4 Experimental Results

In order to test the quality and scalability of the algorithm, experiments have been done both in simulation and with real robots. In simulation, experiments have been repeated with 7 and 50 robots, while in the real environment just 7 of them were used.

Three types of experiments were done in simulation. In *Type 1* experiments, robots move in an unbounded arena without borders always in the same direction. In *Type 2* experiments, the arena has borders marked on the floor that robots are able to detect. When one robot detects that it has arrived to the border it communicates that it has reached the limit to the rest of the flock and the direction of movement is inverted. In *Type 3* experiments, robots move in an unbounded arena but they change their desired direction of movement progressively with time, making a complete turn in 50*s*. The aim of this experiment is to prove that the algorithm might be used in combination with a higher level algorithm to make more complicate tasks and not only move in a linear direction. With real robots just *Type 2* experiments were done.



(a) (b)

Fig. 2. (a) Flock of 7 real *e-puck* robots. (b) Flock of 50 *e-puck* robots in simulation.

Thirty experiments of 180*s* were done for each type of experiments and number of robots in simulation and reality. In the experiments with real robots, positions and headings of every robot were stored every 100*ms* using a tracking software tool [13].

Three parameters were measured to analyze the performance of the algorithm: *i) group velocity*, this is, the velocity of the center of mass of the group; *ii)* the *area* given by the convex hull [14], this is, the area of the minimum convex polygon that contains all the robots; and *iii)* the *polarization* that is a measure of how well aligned are the headings of the robots.

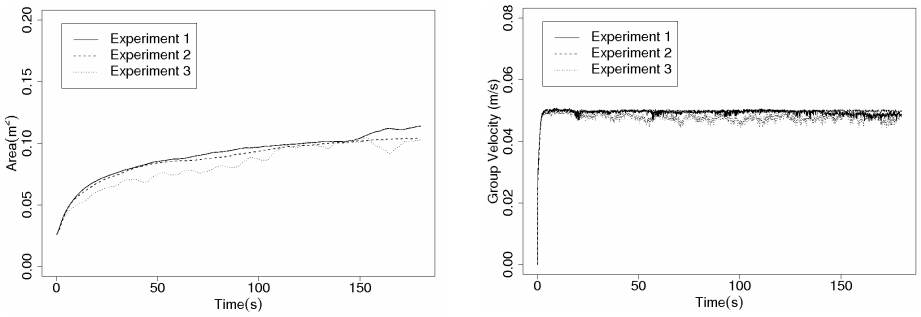
The area measurement is used to identify if the area of the group grows too much, that will indicate that the flock is being split in small groups. Polarization $P(G)$ of a group of robots G is defined as the mean vector angle deviation between the group heading and each individual heading [15]. If all robots are aligned, then $P(G)=0$. Conversely, if headings are evenly distributed, $P(G)=2\pi$. Lastly, if headings are random, i.e. drawn from a uniform distribution, then $P(G)=\pi$ in average. Like robots in the experiment are able to move backwards and forwards, the polarization measure is $P(G)=\pi$ for robots evenly distributed and $P(G)=\pi/2$ for random uniform distribution.

All the systematic experiments done worked well since robots were able to move as a whole in the pre-defined direction, and at the expected group velocities. In Fig. 2, two pictures of flocks in simulation and of real robots can be seen.

4.1 Results in Simulation

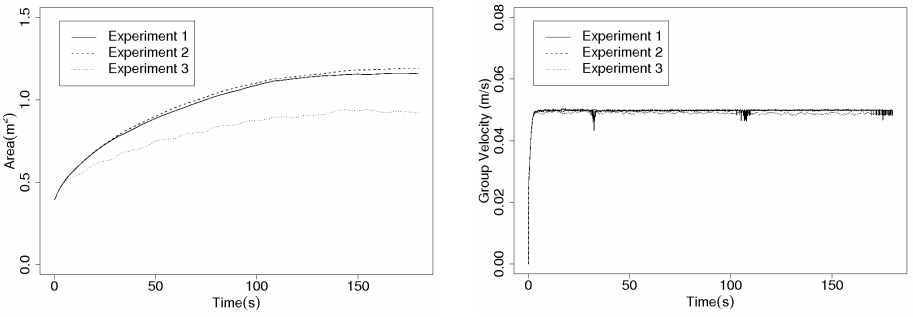
Figure 3a shows how the area of a group of 7 simulated robots evolves on average for the three different types of experiments. As it is observed in the graph, robots start increasing the area and after a period of time (t=120*s*.) they reach a stable plateau of about $0.1m^2$. The group responds as a unique individual, moving on the environment at constant velocity of 0.05*m/s*, after the first seconds in which the flock is created. Small oscillations are observed on its steady-state as shown in Fig. 3b.

Results with 50 robots are similar to the ones with 7 robots as it can be seen in Fig. 4. Robots reach its steady-state area at t=140*s*, which is maintained during the rest of the experiment. There is no difference in the group velocity between experiments with 7 and 50 robots.



(a) (b)

Fig. 3. Results for a group of 7 simulated robots for the three different experiments: (a) area on average (30 experiments), (b) group velocity on average (30 experiments)



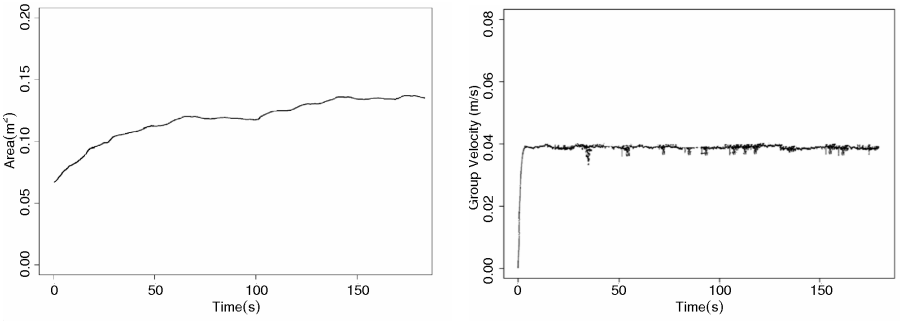
(a) (b)

Fig. 4. Results for a group of 50 simulated robots for the three different experiments: (a) area on average (30 experiments), (b) group velocity on average (30 experiments)

In both group sizes, polarization starts at a approximate value of $\pi/2 rad$ converging to a value of $0.05 rad$ in t=$10s$. Polarization reaches its minimum at the same time as the group velocity is maximum.

4.2 Results with Physical Robots

Experiments with real robots were all of them of *Type 2*. As it can be seen in Fig. 5, their performance is similar to the one in the simulation case, with very small differences in the velocity and area values. There is a first phase when robots try to adopt similar headings, that lasts to t=$15s$, when the group reduces its polarization value down to $0.07 rad$. After this phase, robots keep moving at constant velocity while maintaining the orientation. Group velocity is reduced from what expected from simulation to $0.04 m/s$. This decrement might be due to the magnets which may be introducing friction in the movement, but also to the dynamics derived of moving in the vertical plane. Area increases in similar way as the one in simulation, up to about $0.13 m^2$.



(a) (b)

Fig. 5. Results for a group of 7 real robots for the three different experiments: (a) area on average (30 experiments), (b) group velocity on average (30 experiments)

5 Conclusions and Future Work

The presented algorithm works well according to the carried out experiments. A flock of mobile robot results from the local interactions between the robots, moving at the desired velocity and in a cohesive way. It was proved that the algorithm works with real robots, using simple real infra-red sensors and global heading provided by the *on-board compass*, while in [7] the sensors need to be emulated. Experiments in simulation with 50 robots show the scalability on the number of robots of the algorithm, which represent an advantage compared to other implementations like [9]. Both group velocity and polarization have reasonable values with real robots and in simulation. In addition, the absence of any robot leader and the use of many robots make the flock tolerant to the failure of any of its robots. The good performance of *Type 3* experiments shows that the algorithm could be used for tasks that would need of turns of the flock.

The use of a real compass on each robot, instead of the virtual one using accelerometers, would make the experiments easier since the use of magnets and vertical movement was a limiting factor in the velocity and smoothness of the movements. In addition, if obstacles need to be avoided, a system like the one presented in [16] to detect nearby robots and differentiate from other objects, will be necessary.

Acknowledgments. Authors want to acknowledge the ASL, LIS and SWIS laboratories of the Ecole Polytechnique Fédérale de Lausanne (EPFL) for the design of the *e-puck* robot, and Michele Leidi and Tarek Baaboura of EPFL for their help and ideas about the magnets on the *e-puck*. I. Navarro and F. Matía are partially sponsored by Spanish Ministry of Science (ROBONAUTA project DPI2007-66846-C02-01). I. Navarro is sponsored by Madrid Region with a Ph.D. grant (FPI-2005).

References

1. Hayes, A.T., Martinoli, A., Goodman, R.M.: Distributed Odor Source Localization. IEEE Sensors Journal 2(3), 260–271 (2002)
2. Pugh, J., Martinoli, A.: Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization. In: IEEE Swarm Intelligence Symposium, Piscataway, NJ, pp. 332–339. IEEE Press, Los Alamitos (2007)
3. Howard, A.: Multi-robot mapping using manifold representations. In: IEEE International Conference on Robotics and Automation, Piscataway, NJ, vol. 4, pp. 4198–4203. IEEE Press, Los Alamitos (2004)
4. Reynolds, C.W.: Flocks, herds, and schools: A distributed behavioral model. Computer Graphics 21(4), 25–34 (1987)
5. Tanner, H., Jadbabaie, A., Pappas, G.: Stable flocking of mobile agents, part i: fixed topology. In: Proceedings. 42nd IEEE Conference on Decision and Control, Piscataway, NJ, vol. 2, pp. 2010–2015. IEEE Press, Los Alamitos (2003)
6. Canepa, D., Potop-Butucaru, M.G.: Stabilizing Flocking Via Leader Election in Robot Networks. In: Stabilization, Safety, and Security of Distributed Systems, pp. 52–66. Springer, Heidelberg (2007)
7. Hayes, A.T., Dormiani-Tabatabaei, P.: Self-organized flocking with agent failure: Off-line optimization and demonstration with real robots. In: IEEE International Conference on Robotics Automation, Piscataway, NJ, pp. 3900–3905. IEEE Press, Los Alamitos (2002)

8. Fredslund, J., Mataric, M.J.: A general algorithm for robot formations using local sensing and minimal communication. IEEE Transactions on Robotics and Automation, Special Issue on Multi Robot Systems 18, 837–846 (2002)
9. Balch, T., Hybinette, M.: Social potentials for scalable multi-robot formations. In: IEEE International Conference on Robotics and Automation, Piscataway, NJ, vol. 1, pp. 73–80. IEEE Press, Los Alamitos (2000)
10. Sahin, E.: Swarm robotics: From sources of inspiration to domains of application. In: Sahin, E., Spears, W. (eds.) Swarm Robotics 2004. LNCS, vol. 3342, pp. 10–20. Springer, Heidelberg (2005)
11. Michel, O.: Webots: Professional mobile robot simulation. Journal of Advanced Robotics Systems 1(1), 39–42 (2004)
12. Hsu, H., Liu, A.: Multiagent-based multi-team formation control for mobile robots. Journal of Intelligent and Robotic Systems 42, 337–360 (2005)
13. Correll, N., Sempo, G., de Lopez, M.Y., Halloy, J., Deneubourg, J.-L., Martinoli, A.: SwisTrack: A Tracking Tool for Multi-UnitRobotic and Biological Systems. In: IEEE/RSJ International Conference onIntelligent Robots and Systems (IROS), Piscataway, NJ, pp. 2185–2191. IEEE Press, Los Alamitos (2006)
14. Graham, R.L.: An efficient algorithm for determining the convex hull of a finite planar set. J.-Info.-Proc.-Lett. 1(4), 132–133 (1972)
15. Viscido, S.V., Parrish, J.K., Grunbaum, D.: The effect of population size and number of influential neighbors on the emergent properties of fish schools. Ecological Modelling 183, 347–363 (2005)
16. Pugh, J., Martinoli, A.: Relative Localization and Communication Module for Small-Scale Multi-Robot Systems. In: IEEE International Conference on Robotics and Automation, Piscataway, NJ, pp. 188–193. IEEE Press, Los Alamitos (2006)