

# Graph-Based Distributed Control for Non-holonomic Vehicles Engaged in a Reconfiguration Task using Local Positioning Information

Riccardo Falconi, Sven Goyal, Jim Pugh and Alcherio Martinoli

**Abstract**—Formation building and keeping among vehicles has been studied for many years, since 1987 with Reynolds’ rules [1]. This paper presents a control algorithm, based on recent work in graph theory, able to reconfigure static formations of non-holonomic vehicles endowed solely with local positioning capabilities. The convergence of our approach is mathematically proven and applied to a realistic robotic platform.

## I. INTRODUCTION

The problem of bringing a multi-agent system to a predefined configuration has been studied extensively. Many methods can be used to drive a swarm of robots into a formation: from a simple Braitenberg controller [2] to the use of potential fields [3], [4], [5].

The problem of driving a multi-agent system to a final common state is known as the *consensus problem* and is based on the idea of using some information from the communication network to drive the system to a final state. As explained in [6], consensus algorithms can be used to achieve formations, i.e. drive all the agents to a predefined configuration characterized by a specific steady state. Many problems related to the *consensus* arise when a limited communication range is considered: in these situations the link between robots can be unstable and drive the system to a different final steady state. A possible solution, introduced in [7], takes into account the possibility of combining potential fields with graph theory.

This work presents a solution to the *consensus problem* taking into account not only kinematic constraints but also those related to local inter-robot positioning information. First, we will introduce the basic concepts in graph theory and the classical solution of the consensus problem in Section II. The classical solution enables us to drive a group of holonomic vehicles having global positioning information to a predefined configuration. In Section III, we explain how to solve the consensus problem extended to non-holonomic vehicles using local positioning information. We also give the proof of the stability of our control algorithm. Finally, in Section IV, we test our approach in Webots [11], a realistic mobile robotic simulator carefully calibrated using real robotic data. Section V concludes this paper and introduces possible future work.

Riccardo Falconi is with the Dipartimento di Elettronica, Informatica e Sistemistica of the Università di Bologna. riccardo.falconi@unibo.it

Sven Goyal, Jim Pugh and Alcherio Martinoli are with the Distributed Intelligent Systems and Algorithms Laboratory of the Ecole Polytechnique Fédérale de Lausanne. svenadrian.goyal@epfl.ch, jim.pugh@epfl.ch, alcherio.martinoli@epfl.ch

## II. BACKGROUND

### A. The Consensus Problem

The *consensus problem* [13] is a well-known and widely studied problem in the field of decentralized control. If we consider agents with a single integrator kinematic model

$$\dot{x}_i = u_i \quad (1)$$

where  $x_i$  is the state of the  $i$ -th robot, the consensus problem for  $N$  agents can be solved with the Laplacian based feedback method, that is based on the algebraic description of the system. The feedback control is in the form

$$\dot{x}_i = u_i = -\mathcal{L}x_i \quad (2)$$

where  $\mathcal{L}$  is the Laplacian matrix of the underlying graph representing the swarm of agents. The corresponding closed loop control is reported in Figure 1, where  $g(x) = -\mathcal{L}$  and  $Bias = 0$ . In particular, the *Bias* value can be changed in order to obtain predefined steady states.

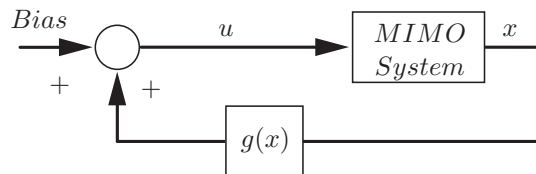


Fig. 1. Closed loop control for the *consensus algorithm*.

## III. FROM HOLONOMIC AGENTS TO REAL ROBOTS

The solution of the consensus problem briefly outlined in Section II-A is based on the assumption that the agents are modelled as holonomic vehicles. Dealing with real robots, we cannot make this assumption because, in reality, robots have some kinematic constraints<sup>1</sup>. A typical example of a real robot model is the differential wheel robot depicted in Figure 2, where the red circle identifies the front side of the robot. The kinematic equations of the  $i$ -th robot are reported in (3).

$$\begin{cases} \dot{x}_i &= u_i \cos(\phi_i) \\ \dot{y}_i &= u_i \sin(\phi_i) \\ \dot{\phi}_i &= \omega_i \end{cases} \quad (3)$$

where  $u_i$  is the linear speed,  $\omega_i$  the rotational speed and  $x_i$ ,  $y_i$  and  $\phi_i$  form the triplet defining the absolute coordinates. Moreover, in our approach, we assume only local data, i.e.

<sup>1</sup>Given our platform of interest, the Khepera III robot [14], characterized by a very low mass, we neglect dynamic constraints.

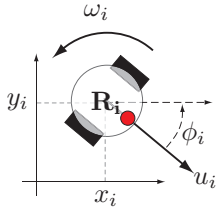


Fig. 2. Kinematic model of a non-holonomic wheeled robot. The red circle identifies the front side.

using for the  $i$ -th robot a range and bearing module to detect the position of the other robots within a given range. This introduces another limitation: a robot cannot have the global position and orientation of all its neighbors. Furthermore, occlusions between robots can happen and the positioning network can become unstable. The MIMO system in Figure 1 for a group of non-holonomic vehicles becomes:

$$\begin{cases} \dot{x}_1 = u_1 \cos(\phi_1) \\ \dot{y}_1 = u_1 \sin(\phi_1) \\ \dot{\phi}_1 = \omega_1 \\ \vdots \\ \dot{x}_N = u_N \cos(\phi_N) \\ \dot{y}_N = u_N \sin(\phi_N) \\ \dot{\phi}_N = \omega_N \end{cases} \quad (4)$$

Since the new system is nonlinear, we have to modify the control loop:

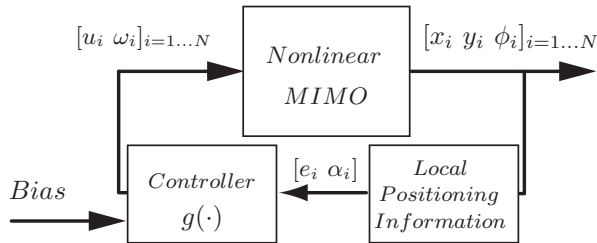


Fig. 3. Control loop for the non-holonomic vehicles.

If we fix the *Bias* value, our goal is to find a control function  $g(\cdot)$  to make the system converge towards a predefined configuration.

#### A. Range and Bearing

A hardware extension board for the Khepera III robot [14] has been developed in [15] to enable robots to find the position of the other robots. The robot has a diameter of 12 cm, making it appropriate for multi-robot indoor experiments. Figure 4 shows the sixteen evenly-spaced infrared Light Emitting Diodes (LEDs) that this platform uses. This range and bearing board has also the ability to broadcast low bit rate communication packets using the IR emitter.

#### B. Control Algorithm

The stability of a swarm of non-holonomic vehicles has been already considered in [16]. However, in these papers the convergence is proved taking into account some data

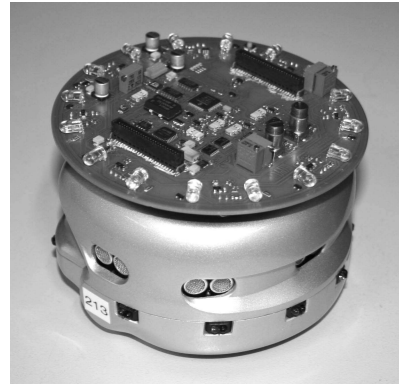


Fig. 4. The range and bearing board developed at EPFL on a Khepera III robot.

that are not accessible to our robots, i.e. the global position and orientation of each robot. In this section a new control algorithm based only on local sensing is introduced, and a proof of convergence and stability is given using a Lyapunov function. With reference to the  $\mathbf{R}_i$  robot in Figure 5, we define the following:

*Definition 1:* For a robot  $\mathbf{R}_i$  with  $\Delta_i \neq 0$  ( $\Delta_i$  is the number of robots connected to the robot  $\mathbf{R}_i$ ), we define:

$$\begin{aligned} \bar{e}_{x,i} &= \frac{1}{\Delta_i+1} \sum_{j=1}^{\Delta_i} [-\mathcal{L}_{i,j} \cdot e_{i,j} \cdot \cos(\alpha_{i,j})] \\ \bar{e}_{y,i} &= \frac{1}{\Delta_i+1} \sum_{j=1}^{\Delta_i} [-\mathcal{L}_{i,j} \cdot e_{i,j} \cdot \sin(\alpha_{i,j})] \end{aligned} \quad (5)$$

where  $e_{i,j}$  is the Euclidean distance between  $\mathbf{R}_i$  and  $\mathbf{R}_j$ , and  $\alpha_{i,j}$  is the azimuth  $\mathbf{R}_j$  with respect to  $\mathbf{R}_i$ . The *global relative error* is defined as  $c_i = [\bar{e}_i, \bar{\alpha}_i]^T$ , where:

$$\bar{e}_i = \sqrt{\bar{e}_{x,i}^2 + \bar{e}_{y,i}^2} \quad (6)$$

$$\bar{\alpha}_i = \text{atan2}(\bar{e}_{y,i}, \bar{e}_{x,i}) \quad (7)$$

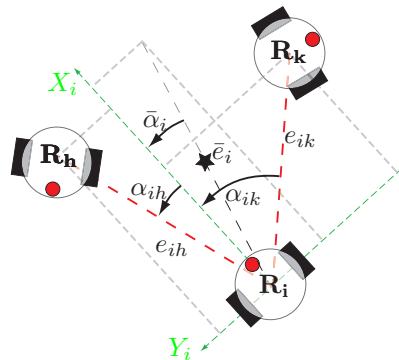


Fig. 5. Definition of the relative position of the centroid of the group (black star) with respect to the robot  $\mathbf{R}_i$ .

The *global relative error*  $c_i$  identifies, with respect to reference frame of the robot  $\mathbf{R}_i$ , a point in the plane that corresponds to the centroid of the group. The demonstration that proves that all robots point to the same centroid is omitted due to space limitations. Our goal is now to define a control vector  $[u_1, \omega_1, \dots, u_N, \omega_N]^T$  that stabilizes the

system. The control law for the  $i$ -th robot is given by:

$$\begin{cases} u_i &= K_1 \cdot \bar{e}_i \cdot \cos(\bar{\alpha}_i) \\ \omega_i &= K_1 \cdot \sin(\bar{\alpha}_i) \cdot \cos(\bar{\alpha}_i) + K_2 \cdot \bar{\alpha}_i \end{cases} \quad (8)$$

*Theorem 1:* Assume that the communication graph  $G$  of a given group of robot is connected. Then the decentralized control law (8) will stabilize the system to a final common value.

**Proof :** Consider the candidate Lyapunov function:

$$V(\bar{c}) = V_1(c_1) + V_2(c_2) + \dots + V_n(c_N) = \sum_{i=1}^N V_i(c_i)$$

where:

$$V_i(c_i) = \frac{1}{2} \bar{e}_i^2 + \frac{1}{2} \bar{\alpha}_i^2 \quad (9)$$

and  $\bar{c} = [c_1, \dots, c_N]^T$ . As the candidate Lyapunov function is quadratic in the relative range and bearing with respect to a calculated mean point, it is

$$\begin{cases} V_i(c_i) > 0, \forall c_i \neq 0 \\ V_i(c_i) = 0, \forall c_i = 0 \end{cases}$$

We will consider the stabilization of the generic  $i$ -th robot. Deriving (9) we obtain,

$$\dot{V}_i(c_i) = \bar{e}_i \dot{\bar{e}}_i + \bar{\alpha}_i \dot{\bar{\alpha}}_i \quad (10)$$

As explained in [17], we obtain for the  $i$ -th robot:

$$\begin{cases} \dot{\bar{e}}_i &= -u_i \cos(\bar{\alpha}_i) \\ \dot{\bar{\alpha}}_i &= -\omega_i + \frac{u_i \sin(\bar{\alpha}_i)}{\bar{e}_i} \end{cases} \quad (11)$$

and we can rewrite  $\dot{V}_i$  as:

$$\dot{V}_i = -u_i \bar{e}_i \cos(\bar{\alpha}_i) + \bar{\alpha}_i \left( -\omega_i + \frac{u_i \sin(\bar{\alpha}_i)}{\bar{e}_i} \right) \quad (12)$$

Now, if we apply the control law in (8),

$$\dot{V}_i = -K_1 (\bar{e}_i \cos(\bar{\alpha}_i))^2 - K_2 (\bar{\alpha}_i)^2 \quad (13)$$

as  $K_1, K_2 > 0$ , follows that:

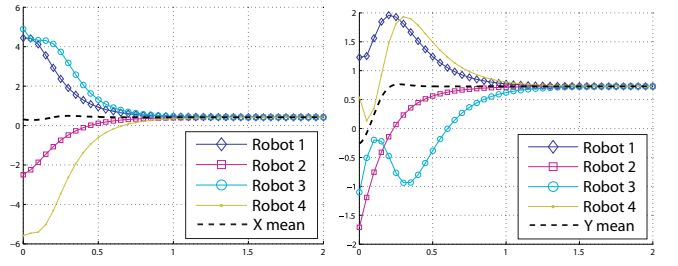
$$\begin{cases} \dot{V}_i(c_i) < 0, \forall c_i \neq 0 \\ \dot{V}_i(c_i) = 0, \forall c_i = 0 \end{cases}$$

which prove that the system converges. ■

In Figure 6 the behavior of a group of four robots is depicted. Note that the mean value (dotted line) is not static but changes dynamically due to the nonlinearities of the control law.

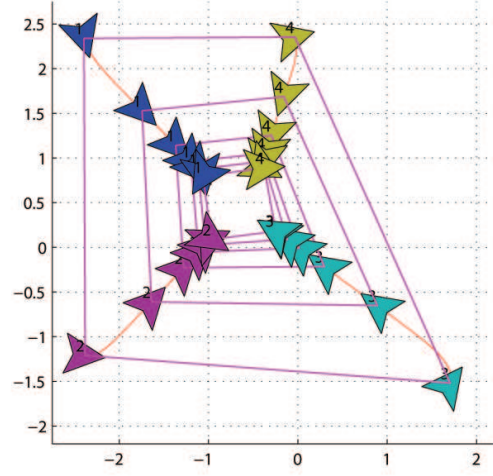
In [6] it is clearly explained how to use the *Laplacian approach* to achieve a formation control: setting the *Bias* input in Figure 1 to a value other than zero, we can drive the system to a predefined final state. Because of the nonlinearities of our system, the *Bias* input is defined as a matrix

$$B(i, j) = \begin{cases} b_{i,j}, & \forall r_j \in \mathcal{N}_i \\ 0, & \text{otherwise} \end{cases}$$

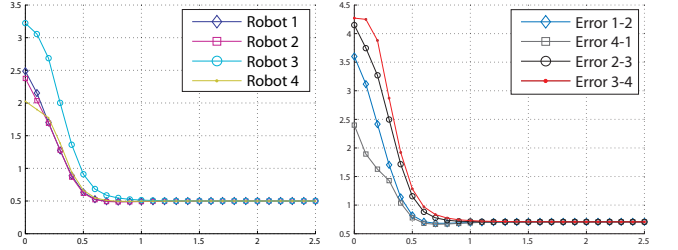


(a) Evolution of the  $X$  coordinates. (b) Evolution of the  $Y$  coordinates.

Fig. 6. Four non-holonomic agents solving the consensus problem.



(a) 2D trajectories of the four robots converging to a square.



(b) Evolution of the distances to the mean point. (c) Evolution of the distances between the robots.

Fig. 7. Four non-holonomic robots converge to a square formation.

where  $b_{i,j}$  is the desired distance between the  $i$ -th and the  $j$ -th robot. As our control law is based on local positioning, it is easy to modify (5) to:

$$\begin{aligned} \bar{e}_{x,i} &= \frac{1}{\Delta_i+1} \sum_{j=1}^{\Delta_i} [-\mathcal{L}_{i,j} \cdot (e_{i,j} - b_{i,j}) \cdot \cos(\alpha_{i,j})] \\ \bar{e}_{y,i} &= \frac{1}{\Delta_i+1} \sum_{j=1}^{\Delta_i} [-\mathcal{L}_{i,j} \cdot (e_{i,j} - b_{i,j}) \cdot \sin(\alpha_{i,j})] \end{aligned} \quad (14)$$

that drives our system to the equilibrium state when all the distances between the robots match the ones specified in the *Bias* matrix (assuming a feasible *Bias* matrix). The symmetry of  $B$  follows from the definition of the *Edge set* of an undirected graph. As an example, we consider 4 robots stabilized in a square configuration with side equal to  $\frac{1}{\sqrt{2}}$  and a diagonal equal to 1. As long as the graph is connected, we assume that it is complete and the corresponding Laplacian

matrix is:

$$\mathcal{L} = \begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix} \quad (15)$$

and the matrix  $B$  is:

$$B = \begin{bmatrix} 0 & \frac{1}{\sqrt{2}} & 1 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 1 \\ 1 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 1 & \frac{1}{\sqrt{2}} & 0 \end{bmatrix} \quad (16)$$

The behavior of the group is depicted in Figure 7.

### C. Additional Optimizations

In the previous section, we proved, that assuming perfect sensors and environment, robots will converge into the specified formation. Unfortunately, the transition to the real world is not as straight forward and we need additional recipes to achieve good performances. The idea presented in the following subsection represent a local optimization technique that we used in the experiments presented in Section IV.

1) *Data Broadcasting*: When navigating through a terrain, robots often encounter obstacles that can prevent point-to-point communication. To make the system more robust and redundant we added the ability for the robots to broadcast their relative positioning information. It has the additional advantage that the underlying graph, representing the swarm, does not need to be complete, but only connected. Broadcasting virtually recreates the missing connections. Figure 8 depicts a typical situation where broadcasting is useful. In this figure, the line-of-sight between robot  $\mathbf{R}_0$  and robot  $\mathbf{R}_2$  is blocked by a wall. Fortunately, robot  $\mathbf{R}_1$  can be used as a relay to compute the missing information about the position of  $\mathbf{R}_2$  with respect to  $\mathbf{R}_0$ . The idea is to have  $\mathbf{R}_1$  broadcasts its relative positioning information (in our case  $\alpha_{10}$ ,  $e_{10}$ ,  $\alpha_{12}$  and  $e_{12}$ , where  $e_{ij}$  is the distance sensed by  $\mathbf{R}_i$  to  $\mathbf{R}_j$ ). This information enables  $\mathbf{R}_0$  to compute an estimation of  $\alpha_{02}$  and  $e_{02}$  with the following equations:

$$\begin{cases} \beta_{02} &= \pi + \alpha_{01} - \alpha_{10} + \alpha_{12} \\ e_{x,02} &= e_{01} \cos(\alpha_{01}) + e_{12} \cos(\beta_{02}) \\ e_{y,02} &= e_{01} \sin(\alpha_{01}) + e_{12} \sin(\beta_{02}) \\ e_{02} &= \sqrt{e_{x,02}^2 + e_{y,02}^2} \\ \alpha_{02} &= \text{atan2}(e_{y,02}, e_{x,02}) \end{cases}$$

The data broadcasted by the range and bearing board is grouped into a data packet. Figure 9 shows how the packet sent by  $\mathbf{R}_0$  and  $\mathbf{R}_1$  in our previous example looks like. This data packet not only contains the measurements ( $\alpha_{ij}$  and  $e_{ij}$ , for all  $j$  such that  $\mathbf{R}_i$  is sensing  $\mathbf{R}_j$ ), but also contains a *hop count*  $h_{ij}$  that is incremented each time a robot broadcasts positioning data that it is not directly measuring. In Figure 8, the *hop count*  $h_{02}$  will be one, meaning that  $\mathbf{R}_0$  used one relay to compute  $\alpha_{02}$  and  $e_{02}$ . The data broadcasted can be ignored if the *hop count* is too high ( $h_{ij} > h_{\max}$ ).

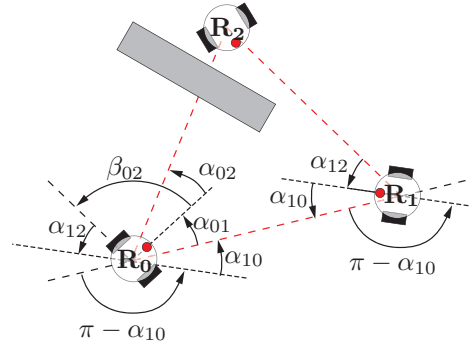


Fig. 8. Example of a blocked line-of-sight where broadcasting is important.

0 (Sender ID)	1 (Sender ID)
1	0
$\alpha_{01}$	$\alpha_{10}$
$e_{01}$	$e_{10}$
0 (= $h_{01}$ )	0 (= $h_{10}$ )
2	2
$\alpha_{02}$	$\alpha_{12}$
$e_{02}$	$e_{12}$
1 (= $h_{02}$ )	0 (= $h_{12}$ )

(a) Data packet sent by  $\mathbf{R}_0$ .

(b) Data packet sent by  $\mathbf{R}_1$ .

Fig. 9. Broadcasted data packet format.

## IV. EXPERIMENTS

Experiments were conducted using simulated Khepera III robots [14]. All sensors and actuators of the simulated robotic platform, including the range and bearing module, were calibrated to match reality. In particular, the range and bearing module suffers from a 10% noise ratio in the estimation of the distance and a 0.1 radians noise in the relative angle.

### A. Experimental Setup

At the beginning of each simulation run, four robots and either zero or ten obstacles are randomly placed in a 3 x 3 m area in the middle of a 4 x 4 m arena. Figure 10 shows an example of initial and final positions for the robotic nodes. Obstacles are represented by cylinders with a 10 cm radius.

The goal of the four robots is to converge to a square formation where the diagonal of the square is one meter. Thus, they use the *Bias* matrix  $B$  (16). The controller of the robot has  $K_1 = 25000$  and  $K_2 = 50000$ . It is important to note that if at any time  $\Delta_i = 0$ , the robot  $\mathbf{R}_i$  will start to move randomly to acquire the positioning information again.

### B. Results

The position of each robot is monitored during a run. After 1000 runs, distances to the center of mass of the robots are computed to assess the convergence of the algorithm. Since the robots need to converge to a square configuration with a diagonal of one meter, their distances to the center of mass should converge to half a meter. Two main scenarios are tested, one without obstacles and one with ten obstacles.



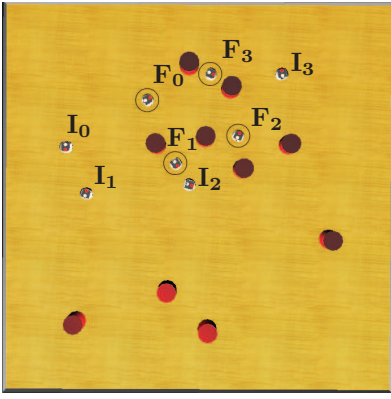


Fig. 10. Example of an experiment with ten obstacles and four robots. Obstacles are in red, robots are white.  $I_i$  and  $F_i$  denote the initial and final position of robot  $R_i$  respectively.

Each scenario is subdivided into two test cases: perfect links between robots (no packet loss) and intermittent connectivity. During the intermittent connectivity, or unstable connectivity, positioning and communication links are unstable and can go up or down with constant probability at a rate corresponding to a Poisson distribution. We chose a mean time constant for the Poisson process of 10 s. Furthermore we analyze the usefulness of the broadcasting algorithm by varying  $h_{\max}$ , the maximal hop count.

Figure 11(a) shows the evolution of the distance to the center of mass with perfect connectivity and no obstacle present. We can see that, even though there are no obstacles, point-to-point communication can still break when another robot passes in between two other robots. Hence, a two hop broadcasting enables a faster convergence of the algorithm. This figure also shows that the approach used converges.

Figure 11(b) shows that, even though the communication links can easily go down, the system remains stable. Since a robot starts to move randomly when it loses all its communication links, the figure shows a mean of the distance to the center of mass a little higher than half a meter and large standard deviations. Again it is worth noting that a two hops broadcasting achieves similar performances to the full broadcasting and we can safely have the robots not send data with two hops or more, thus saving energy.

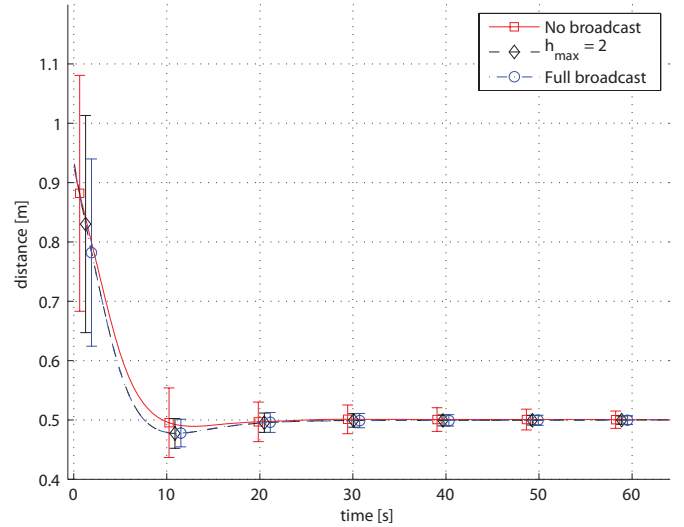
Figure 12(a) and Figure 12(b) really demonstrate the importance of broadcasting in real - not trivial - environments. The figures show that the control law used enables convergence even in more challenging scenarios. Again two hops broadcasting is achieving identical performances to the full broadcasting.

### C. Discussion

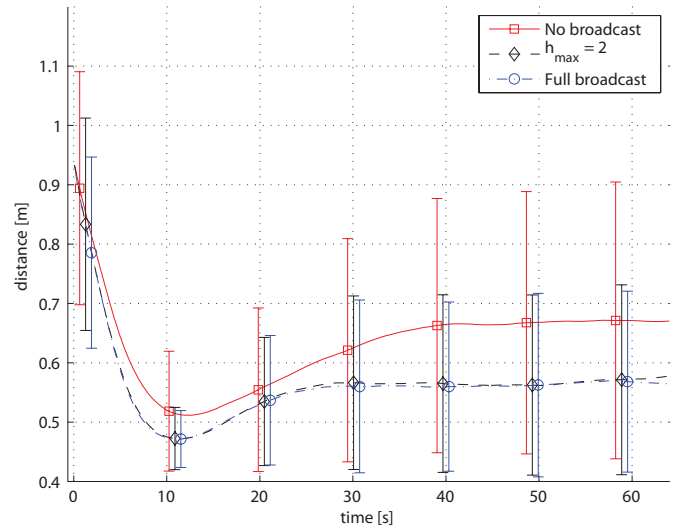
The results obtained during the experiments show two interesting properties:

- the convergence and stability of our approach and
- the usefulness of ad-hoc optimizations in difficult environments.

Using a minimum of two hops for broadcasting enabled a fast convergence of the distributed robotic system to the



(a) Average and standard deviations of the distances to the center of mass depending on time without obstacles and using a perfect connectivity.



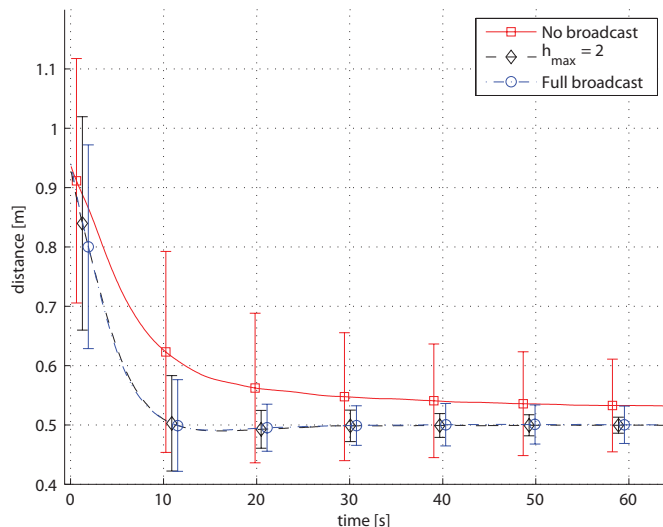
(b) Average and standard deviations of the distances to the center of mass depending on time without obstacles and using an unstable connectivity.

Fig. 11. Average and standard deviations of the distances to the center of mass depending on time without obstacles.

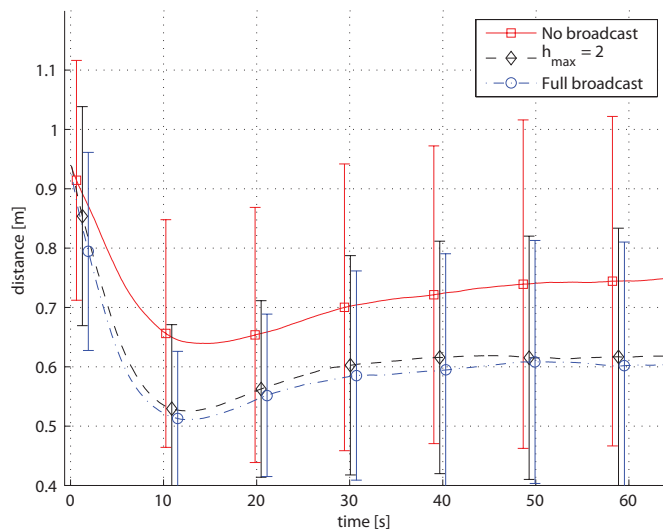
predefined square formation. Broadcasting implicitly adds communication links and makes the communication network more stable; it proved to be particularly efficient when the connectivity was unstable or when obstacles could occlude lines-of-sight.

During some experiments, we observed that the robot team could stabilize to a wrong configuration due to the presence of local minima. To overcome this issue, each robot is able to change the order of the rows and columns of its *Bias* matrix.

Since our approach is fully decentralized, the system is fault-tolerant and the number of robots can easily be increased. On the other hand, if there are more robots, occlusions happen more frequently. To maintain the group connected, we need to increase the maximal number of hops and thus the



(a) Average and standard deviations of the distances to the center of mass depending on time with obstacles and using a perfect connectivity.



(b) Average and standard deviations of the distances to the center of mass depending on time with obstacles and using an unstable connectivity.

Fig. 12. Average and standard deviations of the distances to the center of mass depending on time with obstacles.

propagated positioning information becomes less accurate.

## V. CONCLUSIONS

In this paper, we demonstrated that we could drive non-holonomic robots to a specific formation using only local positioning information. We proved mathematically that our approach converges. In particular, before dealing with the formation problem, we solved the *consensus problem* for kinematically constrained vehicles. We also tested the robustness of our control under challenging conditions such as obstacle field arenas and unstable communication links. The next step of this work will include systematic validation between simulated and real robots. In particular, we will focus on real-world sensor and actuator limitations.

## ACKNOWLEDGEMENT

Sven Goyal and Jim Pugh have received partial support for this work from the Swiss National Science Foundation (Contract Nr. PP002-116913).

## ADDITIONAL MATERIAL

A video showing four simulated Khepera III robots in the experimental arena is available on <http://www5.epfl.ch/swis/page34198.html>.

## REFERENCES

- [1] C. Reynolds, *Flocks, Birds and schools: a Distributed Behavioral model*, Computer Graphics, Vol. 21, 1987, pp 25-34.
- [2] V. Braitenberg, *Vehicles, Experiments in Synthetic Psychology*, The MIT Press, 1984.
- [3] V. Gazi, *Swarm Aggregations Using Artificial Potentials and Sliding-Mode Control*, IEEE Transaction on Robotics, Vol. 21, No.6 December 2005, pp 1208-1214.
- [4] N. E. Leonard, E. Fiorelli, *Virtual Leaders, Artificial Potentials and Coordinated Control of Groups*, 40th IEEE Conference on Decision and Control, 2001, pp 2968-2973.
- [5] M. C. De Gennaro, A. Jadbabaie, *Formation Control for a Cooperative Multi-Agent System using Decentralized Navigation Functions*, 2006 American Control Conference, 2006, pp 1346-1351.
- [6] M. Egerstedt, *Graph-Theoretic Methods for Multi-Agent Coordination*. ROBOMAT, 2007.
- [7] M. Ji, M. Egerstedt, *Distributed Coordination Control of Multi-agent System while Preserving Connectedness*, IEEE Transactions on Robotics, Vol. 23, No.4, August 2007, pp. 693-703.
- [8] D. V. Dimarogonas, K. J. Kyriakopoulos, *On the State Agreement Problem for Multiple Unicycles*, 2006 American Control Conference, 2006, pp 2016-2021.
- [9] D. V. Dimarogonas, K. J. Kyriakopoulos, *On the Rendezvous Problem for Multiple Nonholonomic Agents*, IEEE Transaction on Automatic Control, Vol.52, No.5, May 2007, pp 916-922.
- [10] B. Lei, W. Li, F. Zhang, *Stable Flocking Algorithm for Multi-Robot Systems Formation Control*, 2008 IEEE Congress on Evolutionary Computation, 2008, pp 1544-1549.
- [11] O. Michel, *Webots: Professional mobile robot simulation*, Journal of Advanced Robotic Systems, Vol.1, No.1, 2004, pp. 39-42.
- [12] C. Godsil, G. Royle, *Algebraic Graph Theory*, Springer, 2001.
- [13] R. Olfati-Saber, J. A. Fax, R. M. Murray, *Consensus and Cooperation in Networked Multi-Agent Systems*, Proceedings of the IEEE, Vol. 95, No. 1, January 2007, pp 215-233.
- [14] T. Lochmatter, X. Raemy, L. Matthey, S. Indra and A. Martinoli, *A Comparison of Casting and Spiraling Algorithms for Odor Source Localization in Laminar Flow*, 2008 IEEE International Conference on Robotics and Automation, 2008, pp 1138-1143.
- [15] J. Pugh, X. Raemy, C. Favre, R. Falconi and A. Martinoli, *A Fast On-Board Relative Positioning Module for Multi-Robot Systems*, IEEE Transactions on Mechatronics, focus section on Mechatronics and Multi-robot systems, 2009. To appear. Available online at: <http://infoscience.epfl.ch/record/131138>.
- [16] Y. Liang, H. Lee, *Decentralized Formation Control and Obstacle Avoidance for Multiple Robots with Nonholonomic Constraints*, 2006 American Control Conference, 2006, pp 5596-5601.
- [17] M. Aicardi, G. Casalino, A. Bicchi, A. Balestrino, *Closed Loop Steering of Unicycle-like Vehicle via Lyapunov Techniques*, IEEE Robotics and Automation Magazine, March 1995, pp 27-35.
- [18] A. Jadbabaie, J. Lin, A. S. Morse, *Coordination of groups of mobile autonomous agents using nearest neighbor rules*, IEEE Transactions on Automatic Control, Vol. 48, 2003, pp. 988-1001.
- [19] J. P. Desai, J. P. Ostrowski, V. Kumar, *Modeling and Control of Formations of Nonholonomic Mobile Robots*, IEEE Transactions on Robotics and Automations, Vol. 17, No.6, December 2001, pp. 905-908.
- [20] P. Ogren, M. Egerstedt, X. Hu, *A Control Lyapunov Function Approach to Multiagent Coordination*, IEEE Transaction on Robotics and Automation, Vol.18, No.5, October 2002, pp 847-851.