Distributed orientation agreement in a group of robots

Iñaki Navarro · Fernando Matía

Abstract In this article, a method for the agreement of a set of robots on a common reference orientation based on a distributed consensus algorithm is described. It only needs that robots detect the relative positions of their neighbors and communicate with them. Two different consensus algorithms based on the exchange of information are proposed, tested and analyzed. Systematic experiments were carried out in simulation and with real robots in order to test the method. Experimental results show that the robots are able to agree on the reference orientation under certain conditions. Scalability with an increasing number of robots was tested successfully in simulation with up to 49 robots. Experiments with real robots succeeded proving that the proposed method works in reality.

Keywords Multi-robot · Consensus · Decentralized control · Distributed systems

1 Introduction

A new method for a set of robots to agree in a distributed way on a common reference orientation is presented, explained and analyzed in this article. Robots share and maintain a common heading that is not related to any global reference and it is just a result of their initial reference values. They do not make use of any global information or compasses, the reference orientation is just computed by using a relative positioning system and wireless communications among neighboring robots. Sharing a common reference orientation involves agreement on the directions and orientations of both axes of a coordinate system, but it does not imply agreement on the origin.

Sharing a common reference orientation can be quite useful in many collective movement and formations of robots algorithms. The first work on flocking by Reynolds (1987), makes use of local rules between neighboring robots, forcing them to match their velocities. This implies that each agent must know the relative heading of its nearby mates, so having a common reference orientation can be useful. If robots share a common reference they can exchange their relative headings. This can be also helpful in many other flocking and formation algorithms (Carling et al. 2003; Chuang et al. 2007; Navarro et al. 2008, 2009), which require that the relative heading to neighboring robots is known. In addition, some formation architectures need of a common reference orientation (Martinson and Payton 2005; Lee and Chong 2006).

One of the most straight forward solutions in robotics, in order to share a reference orientation, is to use magnetic compasses. As stated by Borenstein et al. (1996), the magnetic field of the earth can not be well measured near power lines or steel structures due to their influence. Another study of the sources of error in a specific compass concludes that some of them, like vibration, can only be partially filtered, while others, like external interferences from the robot and other electromagnetic sources are more difficult to cope with (Ojeda and Borenstein 2000). These drawbacks, together with the need of providing every robot with a compass, is a reason for designing and implementing the algorithm proposed in this article. On the other hand Turgut et al. (2008) and Çelikkanat et al. (2009) make a group of robots to flock using on-board magnetic compasses. They assume that in the presence of external interferences the sensed north remains approximately the same within neighboring robots. According to them, the measures are quite robust, even in indoor environments with metal objects.

In a previous work by Navarro et al. (2008), robots move in a vertical plane, which allows robots to use gravity as a global orientation reference. They use their built-in 3-axes accelerometers to compute their heading with respect to gravity.

A method for agreement on a common orientation without the use of any global landmark or compasses is described by Lee and Chong (2006). It starts identifying two points in the plane that indicate the reference heading. The first point is the center of mass of the positions of all the robots in the group, while the second point is the position of the most far away robot. This implies that in principle every robot should be aware of the relative positions of every robot in the group, which makes the algorithm not scalable with increasing number of robots. In addition, it assumes perfect readings in the location of nearby robots, and this is never the case in real robotic systems. Two robots could identify different robots as the reference robot, resulting in no orientation agreement. So this method is not useful in real environments, and specially with large number of robots.

A method for agreement on the coordinate axes direction (Flocchini et al. 2008) builds he reference axis while moving the robots to create a desired pattern, which is the principal aim of the algorithm. They demonstrate that under certain conditions (asynchronous robots with memory) robots need to agree on a orientation in order be able to create an arbitrary pattern formation (a pattern is a set of points given by their coordinates). The pattern created by their algorithm is a circle formed by evenly spaced robots.

The main importance on the functionality of the method described in this article is that it provides a set of robots with a common *virtual compass* shared among all the robots and that can be very useful for more complex behaviors. This shared orientation can be used not only for the collective movement applications cited before, but also in collective mapping, collective search, and other collective robotic applications where robots need use a reference orientation. The principal advantage of the method is that robots do not need to include a compass and deal with the inherent problems of these sensors. The only restriction is to be able to estimate orientations to nearby robots and communicate with them.

The rest of this article is organized as follows. Section 2 explains the consensus problems and its relationship with the proposed algorithm. The proposed algorithm is described, analyzed and commented in Sect. 3, while its test experiments and results in simulation are presented in Sect. 4, and for real robots in Sect. 5. Section 6 describes and summarizes an application case of the distributed orientation agreement mechanism: a framework for the collective movement of mobile robots developed by the authors of this article. Conclusions are discussed in Sect. 7, and future lines of work are stated in Sect. 8.

2 Consensus problems

Consensus problems are commonly referred to in the literature as those in which a set of inter-communicated agents must agree on the value of a variable. In our case we want robots to agree on a reference direction. The resulting value in these problems is sometimes the average of all the initial values in each agent, but depending on the application it can be the result of some other function.

The averaging problem serves as inspiration for our agreement on the angle of reference, with the major difference that in our case we need to average angles, and angle values are cyclic since 2π is equivalent to 0. The averaging problem is explained by Denantes (2007) as follows. Each agent i (or robot in our case) must update its estimate $v_i[n]$ at each time step. The update operation is a linear combination of the current estimation and the estimation of its neighbors:

$$v_i[n+1] = w_{ii}[n]v_i[n] + \sum_{j\in\mathcal{N}_i} w_{ij}[n]v_j[n] \qquad i = 1,\ldots,N \tag{1}$$

where w_{ij} are the weighting factors that are defined depending on the algorithm, and $\mathcal{N}_i$ represents the set of neighbors of agent i. The aim of these algorithms is to compute the average:

$$\overline{v} = \frac{1}{N}\sum_{i=1}^{N} v_i[0] \tag{2}$$

The convergence of these averaging algorithms and its conditions are explained and proved both by Denantes (2007) and Xiao and Boyd (2003).

The problem of the averaging agreement in asynchronous systems and a solution based on an exchange message protocol is explained by Mehyar et al. (2005). It is a quite conservative solution but serves to maintain $\sum_{i=1}^{N} v_i[n]$ the same for every time step.

Olfati-Saber and Murray (2003) study the convergence of different consensus protocols (linear and non-linear) depending on the connectivity of the network. A model of the network that uses functions different to 1 in the connectivity links is used to model time delays and distortion and filtering. They use several methods to analyze the convergence, finding out that there is an upper bound on the maximum time-delay that can be tolerated in the network. In addition, it is demonstrated the connectivity of the network is also an important factor. Simulations are presented for different connectivity graphs, where a low number of agents are connected in a triangular lattice. The experimental results are consistent with demonstrated theory.

Some of the applications of the consensus problem include: sensor and information fusion (Talebi et al. 2006; Xiao et al. 2005), coordination of groups of robots (Olfati-Saber 2006; Tanner et al. 2003a, 2003b; Jadbabaie et al. 2003) and coordinated decision making (Bauso et al. 2003; Alanyali et al. 2004). A review of consensus problem on cooperation of robots is presented by Ren et al. (2007).

3 Method

The aim of the proposed method is to make a group of robots agree on a reference orientation without the use of any external reference, compass or global positioning system. This reference orientation is internally calculated by the robots, and does not imply that robots align physically with it. As a result of applying the method, robots share a distributed *virtual compass*, i.e., robots agree on a reference orientation. The agreement is maintained while the algorithm is working serving as a common reference. It can be the result of the initial values of all the robots or just the initial value of one of them. It might be useful for more complex tasks.

The sensor and communication systems used by the robots in this method are:

- Encoders or any other odometry system to keep track of the robot's heading changes;
- A wireless communication system to send and receive messages with neighboring robots;
- A relative positioning system, similar to (Pugh et al. 2009), which allows each robot to determine the bearing of neighboring robots within a certain range. In addition, they must be able to identify the ID of the robot.

The method works as follows. Every robot has an initial estimate of the reference orientation which represents the angular value the group of robots must agree on. Robots exchange this estimate among them and apply a consensus algorithm in order to agree. This reference orientation is expressed on each robot relative to its own coordinate system, so each pair of neighboring robots that exchanges their estimates must know their relative headings. This is performed by using a relative positioning system and exchanging wireless messages. In addition, the rotation of the robot must be taken into account by reading the robot encoders and modifying its reference orientation estimate. The basic steps of the algorithm, executed cyclically, are the following:

Step 0 Each robot r_i has an internal estimate of the reference orientation, expressed in its own coordinate system:

$$\{\theta_i\} \tag{3}$$

Step 1 Each robot r_i reads the bearings to neighboring robots r_j, i.e., the relative angle to robot r_j expressed in the coordinate system of robot r_i (β_{ij}). It is obtained using the relative positioning system and saved together with a time stamp, and the neighbor's identification number (ID_j):

$$\{\beta_{ij}(t_i), t_i, ID_j\} \tag{4}$$

Step 2 Each robot r_i receives wireless messages from every neighboring robot r_j, which contain information about the estimate reference orientation of each neighboring robot expressed in its own coordinate system (θ_j), the bearing where robot r_i was seen from r_j in the coordinate system of r_j (β_{ji}), the time stamp when bearing was measured (t_j), and the identification number of the neighboring robot (ID_j), so later robot r_i can couple information. All this information is saved by robot r_i for each one of its neighboring robots:

$$\{\theta_j, \beta_{ji}(t_j), t_j, ID_j\} \tag{5}$$

Step 3 The reference orientations received from neighboring robots r_j (θ_j) are expressed in the coordinate system of r_j and must be translated to the coordinate system of r_i (θ^i_j). It can be calculated by robot r_i without considering the timing, as follows:

$$\theta^i_j = \beta_{ij} - \beta_{ji} + \pi + \theta_j \tag{6}$$

In Fig. 1 a schematic with the angles involved in this operation is outlined.

But in practice robots move and messages suffer from delays, so for robot r_i to calculate θ^i_j applying Eq. (6) needs to use β_{ij} and β_{ji}, which correspond to exactly the same moment. We consider that robots' clocks are synchronized. When robot r_i receives $\beta_{ji}(t_j)$ from robot r_j it has suffered a delay, so it has to retrieve the value of $\beta_{ij}(t_i)$ corresponding to the same instant $t_j = t_i$. If Eq. (6) is applied with these values the resulting θ^i_j will be for the time t_i, which, in principle, due to the delays, is not the current time. So it is important to correct it using the change of heading

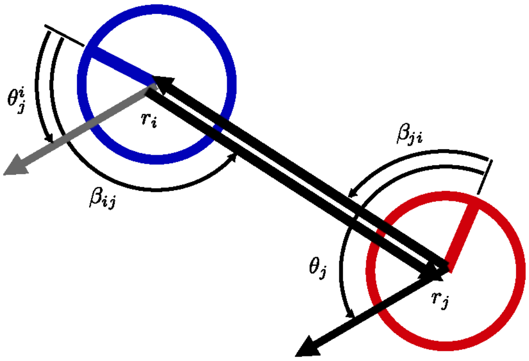


Fig. 1 Angles involved in the calculation of θ^i_j

$\Delta_{odometry}(t_i', t_i)$ of robot r_i, from t_i to the moment t_i', when the θ_i was estimated last. Then these values will be the inputs to the consensus algorithm. The following equation takes these details into account:

$$\theta_j^i\left(t_i'\right) = \beta_{ij}(t_i) - \beta_{ji}(t_j) + \pi + \theta_j - \Delta_{odometry}\left(t_i', t_i\right) \quad (7)$$

Robot r_i needs to get $\beta_{ij}(t_i)$ from its stored values. Since probably the value will not be stored for that exact moment, the robot will interpolate from the two samples with timing values around t_i. The same applies for the odometry $\Delta_{odometry}(t_i', t_i)$.

Each robot r_i will calculate $\theta_j^i(t_i')$ for every neighbor $r_j \in \mathcal{N}_i$, from which receives a wireless message containing $\beta_{ji}(t_j)$ and t_j.

Step 4 Each robot r_i applies a consensus algorithm using the neighbors' reference orientation $(\theta_j^i(t_i'))$ and its own estimate can be applied to compute the new estimate. Two different algorithms for this purpose are described in Sects. 3.1 and 3.2. They can be summarized in the next equation:

$$\theta_i\left(t_i'\right) = f\left(\theta_i\left(t_i'\right), \ldots, \theta_j^i\left(t_i'\right), \ldots\right) \quad \forall j \in \mathcal{N}_i \quad (8)$$

This algorithms do not need to compute necessarily the average of the initial values of the different reference orientation, but to make all the robots to agree on these values.

Step 5 The penultimate step in the cycle of the algorithm is for r_i to update θ_i with the odometry for the current time t_i'':

$$\theta_i\left(t_i''\right) = \theta_i\left(t_i'\right) - \Delta_{odometry}\left(t_i'', t_i'\right) \quad (9)$$

Step 6 Lastly, each robot r_i creates and sends wireless messages to neighboring robots. These messages include the reference orientation estimate $(\theta_i(t_i''))$, the bearing to the neighboring robot $(\beta_{ij}(t_i))$, the time-stamp specifying the moment when the robot was observed, (t_i) and the identification number of the robot (ID_i). This information is summarized as follows:

$$\left\{\theta_i, \beta_{ij}(t_i), t_i, ID_i\right\} \quad (10)$$

This information corresponds to the received in step 2 from neighboring robots.

3.1 Consensus Algorithm A

In this section the Consensus Algorithm A (CAA) is described. This algorithm is executed by every robot r_i in order to agree on the common reference orientation using as inputs the reference orientations (θ_j^i) of neighboring robots $r_j \in \mathcal{N}_i$ transposed to the coordinates of r_i, and its own reference orientation estimate (θ_i).

It is implemented in a similar way as the consensus averaging solution explained in Sect. 2 and described by Eq. (1). The main difference is that here an average over angles must be done, and due to the circular property of angles (2π equivalent to 0) they can not be directly weighted and added. Instead, angles are converted to unit vectors with the direction of the angles. If we denote $\mathbf{v}_i$ as the unit vector of angle θ_i, and $\mathbf{v}_j^i$ as the unit vector of angle θ_j^i, then Eq. (1) can be rewritten as follows:

$$\mathbf{v}_i[n+1] = w_{ii}[n]\mathbf{v}_i[n] + \sum_{j \in \mathcal{N}_i} w_{ij}[n]\mathbf{v}_j^i[n] \quad (11)$$

The new reference estimate is calculated by obtaining the angle of $\mathbf{v}_i[n+1]$:

$$\theta_i[n+1] = \arg\left(\mathbf{v}_i[n+1]\right) \quad (12)$$

The weighting parameters w_{ij} will all be set to a configuring parameter α. Then α indicates how much is taken into account information from neighboring robots. w_{ii} is set so the sum of every w_{ij} and w_{ii} its equal to 1, i.e., $w_{ii} = 1 - \sum_{j \in \mathcal{N}_i} w_{ij} = 1 - N_i * \alpha$, where N_i is the number of neighbors of r_i. The only restriction to this two definitions is that $w_{ii} \leq w_{ij}$, because when filtering we do not want to give more relevance to the information of a nearby robot than the own one. So if a certain combination of α and N_i makes $w_{ii} < w_{ij}$, then they will be redefined as: $w_{ii} = w_{ij} = \frac{1}{N_i+1}$. This will depend on the number of neighbors that a robot r_i will have at a certain moment. The larger $w_{ij} = \alpha$ more information will integrate each robot its neighbors. If a robot has 4 neighbors and $\alpha = 0.1$ the estimates of the reference orientation neighboring robots will weight 40 % in total for the next estimation, while the own estimate a 60 %. If we have $\alpha = 0.01$ and still 4 neighbors, then nearby robots will contribute with a 4 % to the calculation of the new value and the previous estimation with a 96 %.

If we were working with real numbers and wanted to agree on using the average, the convergence of these algorithms can be demonstrated under certain conditions as outlined in Sect. 2. But in this case we want to agree on an angle, and angles have the property of being cyclical in their values, since 2π is equivalent to 0. In the work by Jadbabaie et al. (2003), the convergence is demonstrated working without taking into account the cyclical property. The resulting average of $\theta_1 = 0$ and $\theta_2 = 2\pi$ would be π and not 0. In their case it can be done because the agents in this algorithm share a common global coordinate system, but this is not the case of the present method.

Although it computes the average from the initial values of the reference orientation of each robot, it is not a requirement, and the aim is just to make all the robots to agree on a reference orientation.

Here we present an example in which our algorithm diverges. As will be seen in Sect. 4, this is a very similar situation to what has been seen in some of the obtained results.

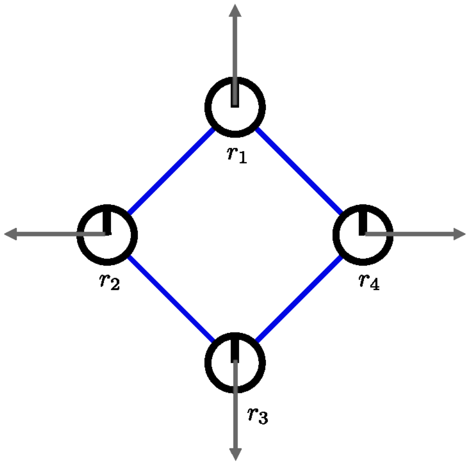


Fig. 2 An initial condition in which CAA diverges

In Fig. 2, four robots are shown, each of them having two neighboring robots, and with evenly distributed reference orientations indicated by the vectors. If we consider no communication losses, perfect synchronization in the robots' clocks, and that robots update their estimates synchronously, then given the initial reference orientations, CAA will diverge. If applying CAA in robot r_1 we have:

$$\begin{aligned}\mathbf{v}_1[n+1] &= w_{11} * \mathbf{v}_1[n] + w_{14} * \mathbf{v}_4^1[n] + w_{12} * \mathbf{v}_2^1[n] \\ &= w_{11} * \mathbf{v}_1[n] \end{aligned} \tag{13}$$

As it can be seen in Fig. 2, $\mathbf{v}_2^1$ and $\mathbf{v}_4^1$ are opposite vectors and by definition $w_{12} = w_{14}$, so the term $w_{14} * \mathbf{v}_4^1[n] + w_{12} * \mathbf{v}_2^1[n]$ is equal to zero, since vectors are canceled with each other. Then, the resulting θ_1 will be the same as the initial one, so in the next step $\mathbf{v}_1$ will remain unchanged. The same happens for the four robots, so they will never agree on a common reference orientation.

It looks that this problem would be just overcome by noise and communication delays that would break the equilibrium. But as it will be seen in most of the experiments of Set 2 in simulation, the problem takes place. Although robots do not start in this state of equilibrium, in some of the experiments they reach it as it can be seen in Fig. 6(b).

Next section describes a modification of CAA, that intuitively overcomes the problem described in this previous example.

3.2 Consensus Algorithm B

Consensus Algorithm B (CAB) is a modification of CAA, to overcome the divergence problem experienced under certain topology configurations. The basic idea behind CAB is that initially there exists a robot which imposes its orientation reference to its neighbors, in the next step its neighbors do the same, and so on, resulting in a propagation by flooding of the initial reference orientation of one of the robots.

In CAB, robots maintain and exchange via wireless messages a variable about how confident they are on their reference orientation estimate. Initially every robot has this variable set to zero, except one of the robots that will have it set to one. We will name this robot *seed robot*. Confidence is used to estimate the new orientation reference, creating the propagation of the initial reference orientation of the *seed robot*. When a robot updates its orientation estimate it also updates its confidence setting it to the maximum of its own confidence and neighbors' confidences. It can be summarized in the following equations:

$$\mathbf{v}_i[n+1] = c_i[n] w_{ii}[n] \mathbf{v}_i[n] + \sum_{j \in \mathcal{N}_i} c_j[n] w_{ij}[n] \mathbf{v}_j^i[n] \tag{14}$$

$$c_i[n+1] = \max\big(c_i[n], \ldots, c_j[n], \ldots\big) \quad \forall j \in \mathcal{N}_i \tag{15}$$

where c_i is the confidence variable of robot r_i and c_j the confidence of neighbor r_j.

After the propagation of the reference orientation of the *seed robot*, every robot will have a similar reference orientation, that will be slightly different due to the errors introduced by noise and lack of synchronicity. After that CAB will act basically as a distributed filter on the common reference orientation. We will not prove the convergence of CAB here, but empirically we will show how robots are able to agree. After the propagation every robot will agree on the reference orientation.

CAB does not compute the average from the initial values of the reference orientation of each robot, but makes all the robots to agree on a value similar to the reference orientation of the *seed robot*. The goal is that all the robots agree on the reference orientation.

For a better understanding, a graphical example on the workings of CAB is shown in Fig. 3.

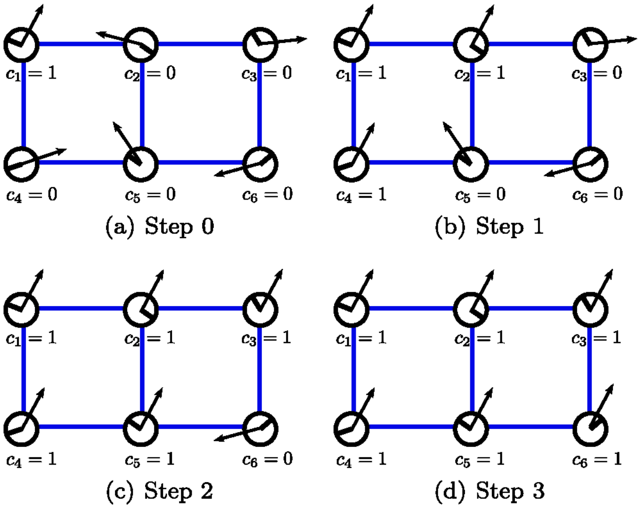
The example explained in CAA (Fig. 2) will in this case allow the robots to agree on the reference orientation given initially by the robot which will be labeled as *seed robot*.

4 Experiments in simulation

In order to test the method described in this article, systematic experiments were performed both with real robots and in simulation. Simulated experiments and their results are the scope of this section, while real robot experiments are described in Sect. 5. Simulated experiments were run using Webots simulator, with a realistic model of the Khepera III robot (Pugh et al. 2009).

We ran nine different sets of simulated experiments, each one composed of 30 different experiments, with the same parameters but different random initial headings of the robots. The sets are differentiated by:

- the topology of the robots' communication graph,



(a) Step 0

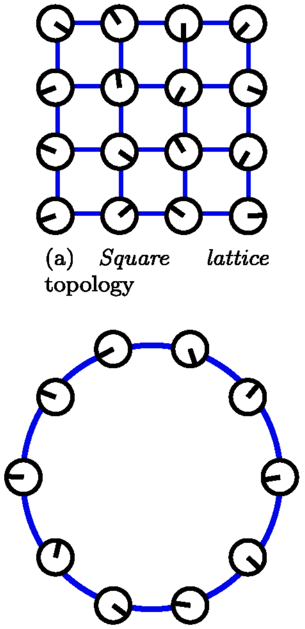
(b) Step 1

(c) Step 2

(d) Step 3

Fig. 3 Example showing graphically the workings of CAB algorithm. (**a**) Initial step 0. Robot r_1 has a $c_1 = 1$ and θ_1 indicated by the *vector* in the sketch. The rest of the robots have different θ_i and $c_i = 0$. (**b**) Step 1. Robots r_2 and r_4 update θ_2 and θ_4 respectively. Since $c_1 = 1$ (received from r_1), then applying Eq. (14) results in $\theta_2 = \theta_1^2$ and $\theta_4 = \theta_4^1$, as it can be seen in the sketch. Afterwards, using Eq. (15) they update $c_2 = 1$ and $c_4 = 1$, given that r_1 is a neighboring robot and $c_1 = 1$. (**c**) Step 2. As previously done by robot r_2 and r_4, this time r_3 and r_5 update θ_3 and θ_5 with the same orientation as θ_1, θ_2 and θ_4, imposed by c_2 and c_4. Later, using Eq. (15), c_3 and c_5 are set to 1. (**d**) Step 3. The same happens for robot r_6, which updates it θ_6 in the same direction as the rest of the robots using Eq. (14). It updates $c_6 = 1$ applying Eq. (15). In addition, during the next steps robots continue updating and agreeing on θ_i, but now every $c_i = 1$

Fig. 4 Communication topologies used in the experiments



(a) *Square lattice* topology

(b) *Ring* topology

- the behavior of the robots,
- the α parameter of the consensus algorithm,
- the presence or not of error in the synchronicity of the robots' clocks,
- and the consensus algorithm chosen (CAA or CAB).

All the experiments were run using 49 robots, placed in one of the two communication topologies chosen: *square lattice* and *ring*. In the *square lattice* topology the 49 robots are placed in seven rows and seven columns, inner robots having four neighbors and bordering robots two or three neighbors. A schematic of this topology with less robots is shown in Fig. 4(a). In the *ring* topology, robots form a ring, each one of them having two neighbors, as can be seen in Fig. 4(b). The *ring* topology was chosen because of its simplicity, it is the one that ensures that every robot has the minimum number of neighbors and also the same number of them. The *square lattice* was chosen because it represents a simple mesh that covers a 2-dimensional plane, but also a triangular lattice could have been chosen. Since in *square lattice* topology robots are more connected, faster agreement can be expected than in *ring* topology. Only static topologies were study in order keep the analysis and study simpler.

Two different movement behaviors have been used to test the proposed method. In the *random turn* behavior robots spin on the spot with a random evenly distributed angular velocity ω between -2 rad/s and 2 rad/s. The aim of this behavior is to present a difficult case since robots change their heading continuously and this can make the method sensitive to synchronization problems and odometry errors. It can only be tested in simulation, since in order to measure its quality it is necessary to know every step the heading of each robot and their own reference orientation estimate (θ_i). By coupling these two values, the reference orientation of each robot (θ_i) is calculated relative to an external reference system (θ_i^{EXT}). In the *alignment* behavior robots align their headings with their reference orientation θ_i. In this way, the evolution of the method can be observed externally by looking at the robot headings. This behavior is necessary to test the method with real robots. The headings of the robots expressed relative to an external reference system (ϕ_i) are used to determine the performance of the method. When robots are in *alignment* behavior and correctly aligned ϕ_i is equivalent to θ_i^{EXT}. ϕ_i is calculated only in the *alignment* behavior as an indirect way of reading θ_i^{EXT} which is not possible in the case of real robots.

Random turn behavior is tested in seven of the simulation sets, while *alignment* is tested in the other two. For a better understanding of both behaviors 3 videos can be found at http://www.robolabo.etsit.upm.es/~inaki/article_DOAGR/videos/. Two videos show simulations of robots behaving as *random turn* and *alignment*, while in the third one real robots perform *alignment*.

In the simulations performed, robots read the relative positions of their neighbors every 128 ms. They also exchange wireless messages every 128 ms; they are received by their neighboring robots with a delay of 128 ms. The bearing accuracy of the relative positioning system is modelled with additive Gaussian noise of standard deviation $\sigma = 0.1$ rad. The odometry noise is modeled as follows. A uniform noise of ±10 % on the movement due to slip of the wheels, which implies that the wheels (and encoders) move but robot's movement is modified by this value. It is calculated for each wheel and each simulation step. On the other hand, no noise is considered measuring the encoder steps which means also that wheels have an ideal radius and encoders are read perfectly. The duration of each experiment is 60 s. This period was chosen since it was thought that after 30–40 s the orientation agreement would be in steady state. After analyzing the results from the systematic experiments we realized that these was not true for all the sets. More specifically in Set 6 and Set 7 steady state is not reached for $t = 60$ s and longer experiments would have been desirable in order to give better statistical results.

Analyzing both CAA vs CAB it will allow to test the fast initial agreement of CAB compared to CAA, and see in cases of divergence of CAA how CAB solves it. In the experiments using CAB, the *seed robot* is placed in one of the external corners of the lattice. This position is the one that ensures the longest path propagation of the reference orientation of this robot towards the robot in the opposite corner. In addition, corner robots only have two direct neighboring robots. In this manner, we test the most difficult case. If the *seed robot* were placed in the center of the lattice then propagation would be easier and faster.

Different values of α are used to compare the speed of agreement and final accuracy of the algorithms. Large α implies large w_{ij} parameters which measure how much information from neighboring robots is used to obtain the new estimate of θ_i. More exchange of information implies then faster agreement, unless there exist oscillations. The consensus algorithm can be seen as a proportional controller, that aims to approximate the reference orientation of nearby, where w_{ij} represents the proportional parameter. Then large w_{ij} (and so large α) will imply faster agreement but also less accuracy. As it will be seen in the results low values of α give after some time more accuracy on the agreement.

All the sets of experiments studied in simulation except one consider perfect synchronicity among the robots. This helps to isolate the consequences of asynchronicity and be able to compare other variables as α, topology, type of algorithm, and behavior used. In addition, no perfect synchronization exists in the real robot experiments but the synchronization bias in real robots is smaller than the tested in simulation with asynchronicity, so this simulation set represents a worst case test that will not occur in reality.

Two metrics: *Mean Orientation Error* (*MOE*) and *Polarization* (*Po*) (Navarro and Matía 2009; Gutiérrez et al. 2009), are used to measure the error in the orientation agreement, in order to assess the performance of the method applied to ϕ_i or θ_i^{EXT}. *MOE* is the error orientation of each robot ϕ_i or θ_i^{EXT} with respect a reference. This reference orientation (θ_{ref}) used in this case is the instant mean of ϕ_i or θ_i^{EXT} over the different robots. In the case of ϕ_i, the heading mean can be calculated as follows: $HM = \arg(\sum_{i=1}^{N} \mathbf{u}_{\phi_i})$, where $\mathbf{u}_{\phi_i}$ is a vector of magnitude 1 and direction ϕ_i. The *MOE* is defined as:

$$MOE = \frac{1}{N} \sum_{i=1}^{N} |\phi_i - \theta_{ref}| \tag{16}$$

If robot orientations are equally spaced then $\sum_{i=1}^{N} \mathbf{u}_{\phi_i} = \mathbf{0}$, so HM will be undetermined, but in this case $MOE = \frac{\pi}{2}$, independently of ϕ_i.

Po is similar to *MOE* and also aims to measure how similar are robots' angles under study (ϕ_i or θ_i^{EXT}). It is defined using the angular nearest neighbor. For a robot r_i, the corresponding angular nearest neighbor r_{ann} is defined so that $\theta^{r_i r_{ann}}$, the relative angle ϕ_i of robot r_{ann} with respect to ϕ_i of robot r_i is as small as possible: $\theta^{r_i r_{ann}} = \min(\theta^{r_i r_j}), \forall r_j \neq r_i$. We denote $\theta_{ann}(r_i)$ the relative angle of the angular nearest neighbor of the robot r_i. The formal definition of polarization is defined as follows:

$$Po = \sum_{i=1}^{N} \theta_{ann}(r_i) \tag{17}$$

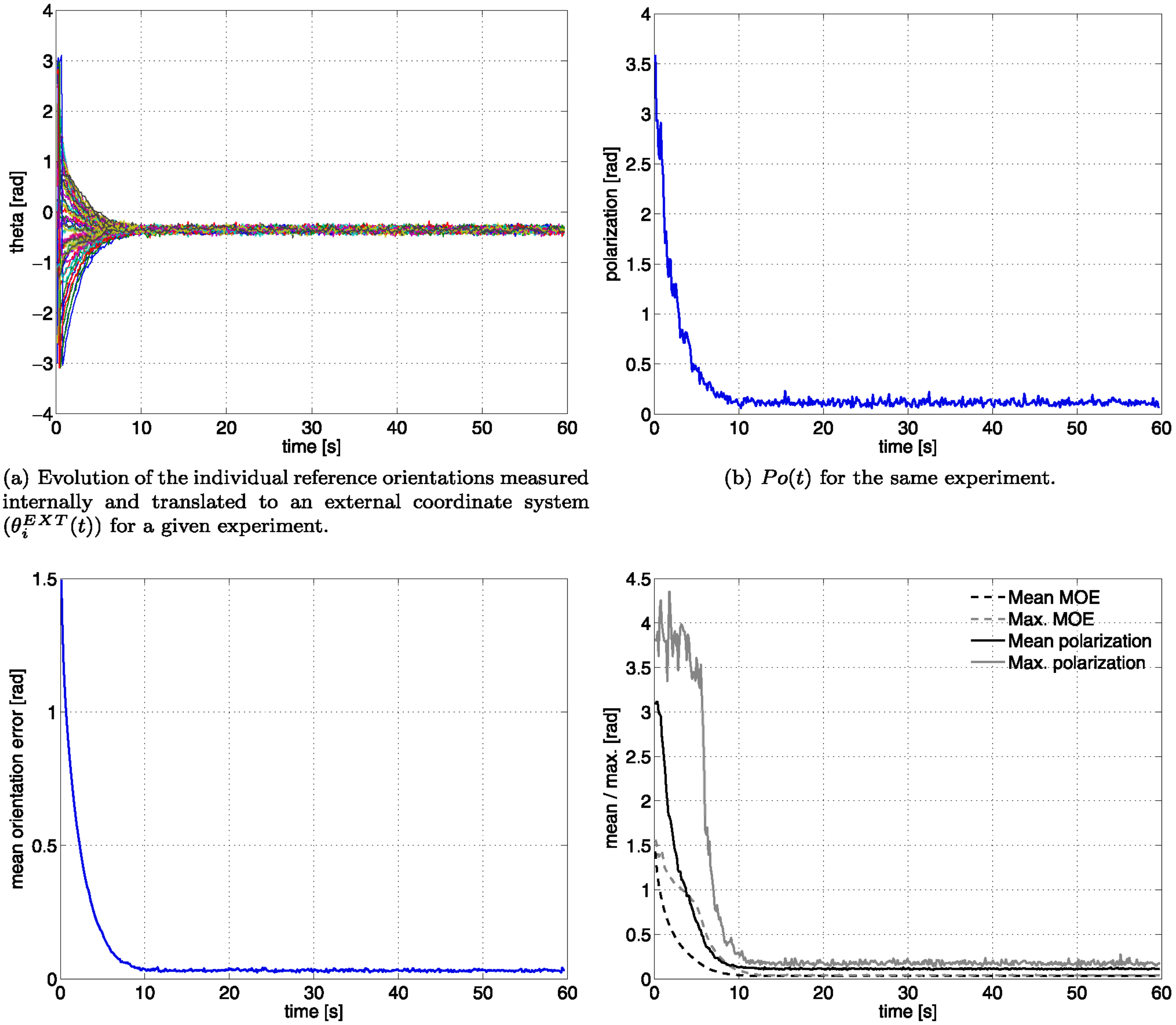
MOE and polarization can be obtained for the θ_i^{EXT} case in the same way. The lower the value of both metrics the more agreement in θ_i^{EXT} (or ϕ_i). $MOE = 0$ or $Po = 0$ indicate perfect agreement. Conversely, if headings are evenly distributed, $Po = 2\pi$.

In addition, the evolution of θ_i^{EXT} (or ϕ_i in the case of *alignment* behavior) with time is shown in most of the sets for one characteristic experiment.

4.1 Set 1: CAA, *square lattice* topology, $\alpha = 0.1$, *random turn*

In this set of experiments, robots apply the CAA as consensus algorithm, with $\alpha = 0.1$. They are placed in *square lattice* topology and perform the *random turn* behavior. In Fig. 5(a), the evolution of the reference orientation of each robot ($\theta_i^{EXT}(t)$) is plotted with respect to time for a given experiment of this set. It can be seen that after time $t = 10$ s the difference between them remains stable and the average value keeps constant. This can be better understood looking at Fig. 5(b) and Fig. 5(c) where *Po* and *MOE* are plotted with time for the same experiment. $Po(t)$, for $t > 10$ s, remains under 0.25 rad and $MOE(t)$ under 0.1 rad.

The instantaneous maximum and the instantaneous mean of *Po* and *MOE* of all the experiments of this set, i.e., the maximum values of *Po* and *MOE* of each experiment for



(a) Evolution of the individual reference orientations measured internally and translated to an external coordinate system ($\theta_i^{EXT}(t)$) for a given experiment.

(b) $Po(t)$ for the same experiment.

(c) $MOE(t)$ for the same experiment.

(d) Instantaneous maximum and instantaneous mean of Po and MOE of all the experiments.

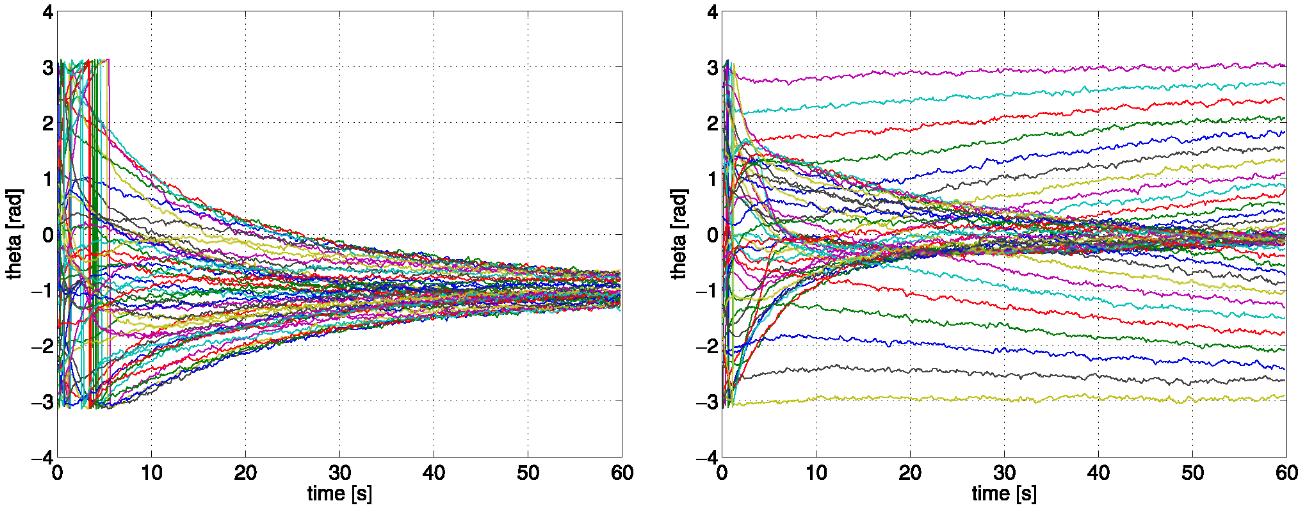
Fig. 5 Set 1: CAA, *square lattice* topology, $\alpha = 0.1$, *random turn*

each temporal step and the mean values of Po and MOE of each experiment for each temporal step, are shown in Fig. 5(d). As it can be seen, after $t = 12$ s the instantaneous maximum and the instantaneous mean of Po and MOE remain all of them approximately under 0.25 rad, which shows that in the 30 experiments robots agree on the reference orientation.

For $t > 15$ s, the mean value of Po for every experiment and for every temporal step is 0.1118 rad, while the maximum value is 0.2736 rad. In the case of MOE this mean value is 0.0311 rad, and the maximum value is 0.0459 rad. These MOE values are quite small in relation to the bearing noise of the relative positioning system of $\sigma = 0.1$ rad.

4.2 Set 2: CAA, *ring* topology, $\alpha = 0.1$, *random turn*

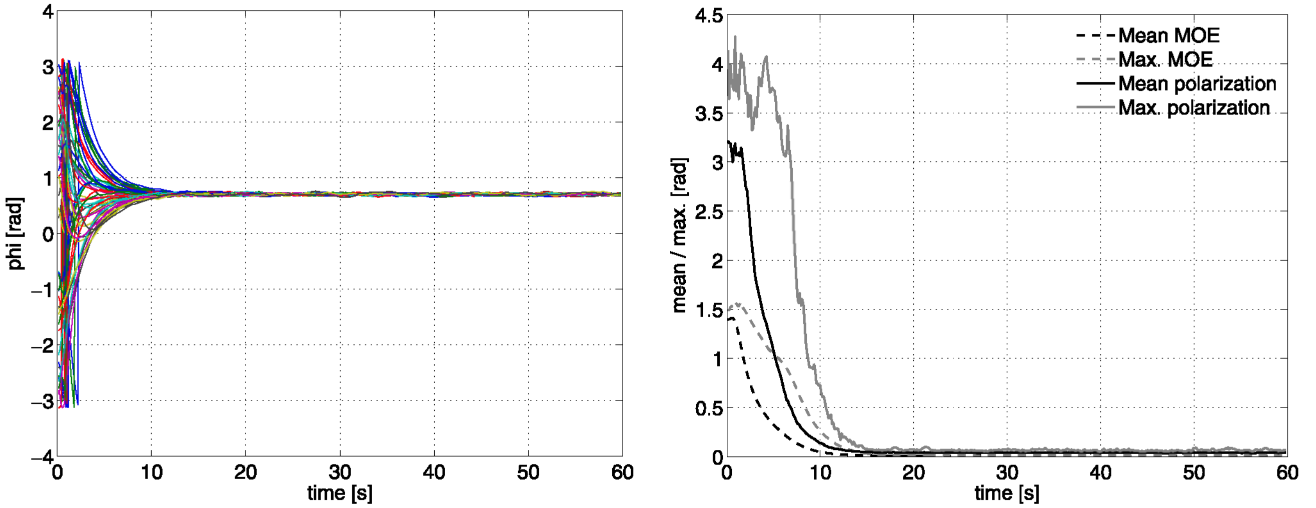
In this set of experiments robots apply CAA as consensus algorithm, with an $\alpha = 0.1$. The topology in comparison with the previous set is the *ring* topology while the *random turn* behavior is the same. In this set of experiments in only 6 of the 30 experiments robots agree and they do so at a slower rate than in Set 1, because this topology presents less connectivity. Since most of the experiments diverge no statistics are given in this case. In Fig. 6(a) the evolution of the reference orientation of each robot is plotted for an experiment with agreement, while in Fig. 6(b) it is plotted for a divergent experiment.



(a) Evolution of the individual reference orientations measured internally and translated to an external coordinate system ($\theta_i^{EXT}(t)$) for a experiment with agreement.

(b) Evolution of the individual reference orientations measured internally and translated to an external coordinate system ($\theta_i^{EXT}(t)$) for a divergent experiment.

Fig. 6 Set 2: CAA, *ring* topology, $\alpha = 0.1$, *random turn*



(a) Evolution of robots' headings aligning with the reference orientation seen from an external coordinate system ($\phi_i(t)$) for a given experiment.

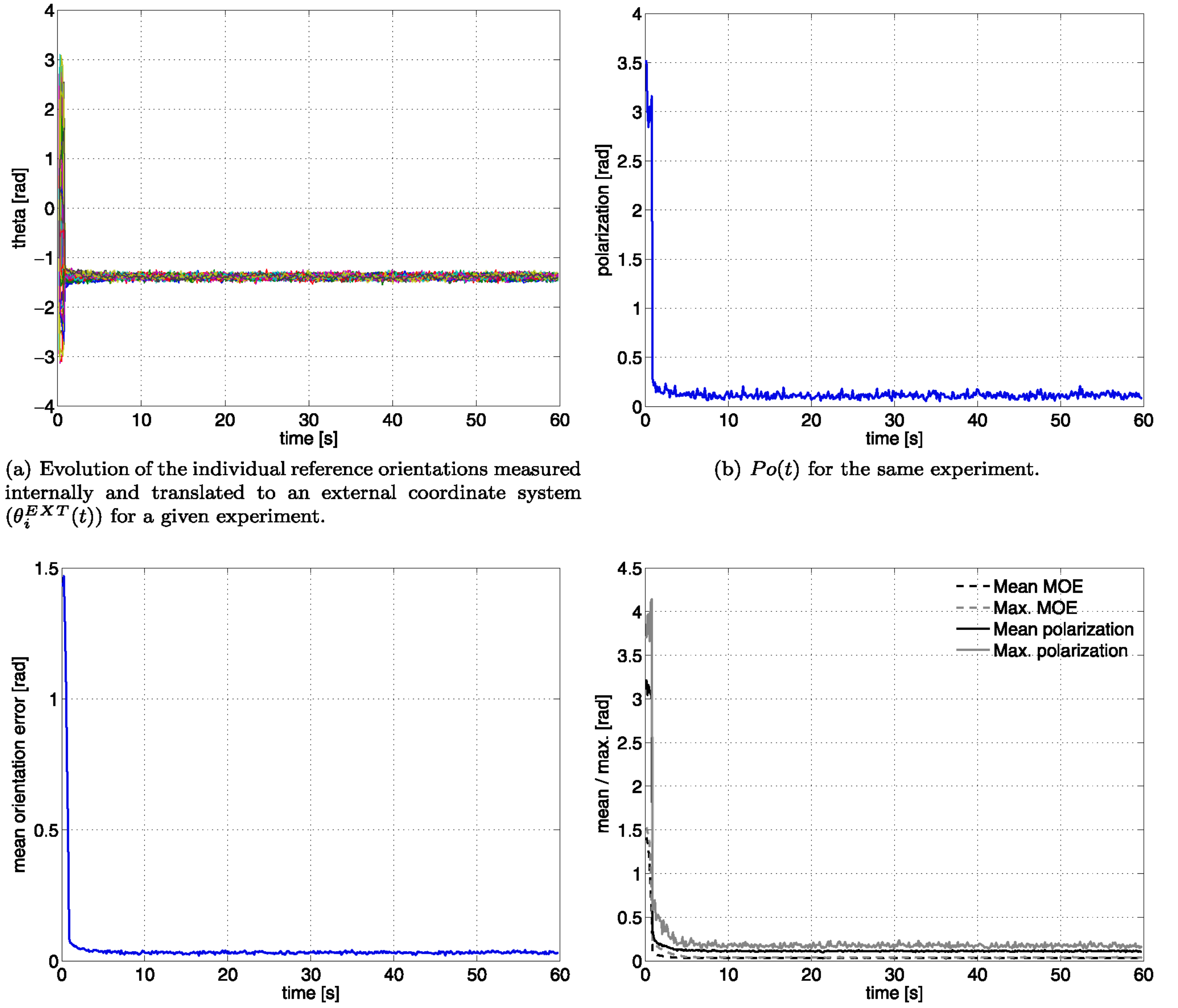
(b) Instantaneous maximum and instantaneous mean of *Po* and *MOE* of all the experiments.

Fig. 7 Set 3: CAA, *square lattice* topology, $\alpha = 0.1$, *alignment*

4.3 Set 3: CAA, *square lattice* topology, $\alpha = 0.1$, *alignment*

In this set of experiments robots apply CAA as consensus algorithm, with $\alpha = 0.1$ as in the two previous sets, but their behavior is *alignment* of its heading with the reference orientation estimate. The placement topology is *square lattice*. In Fig. 7(a) the evolution of the headings of the robots is plotted for a given experiment of this set. In Fig. 7(b) the instantaneous maximum and the instantaneous mean of *Po* and *MOE* of all the experiments of this set are shown. It can be seen that the heading agreement is better than in Set 1, this is because the alignment controller filters the orientation estimates θ_i.

After $t = 15$ s the difference in the headings is very small. In all the experiments robots agree. The mean of *Po* for every experiment and for every temporal step after $t = 15$ s is 0.0420 rad, while the maximum of *Po* is 0.0997 rad. For *MOE* the corresponding values are 0.0115 rad for the mean and 0.0313 rad for the maximum. These values are smaller than in Set 1.



(a) Evolution of the individual reference orientations measured internally and translated to an external coordinate system ($\theta_i^{EXT}(t)$) for a given experiment.

(b) $Po(t)$ for the same experiment.

(c) $MOE(t)$ for the same experiment.

(d) Instantaneous maximum and instantaneous mean of Po and MOE of all the experiments.

Fig. 8 Set 4: CAB, *square lattice* topology, $\alpha = 0.1$, *random turn*

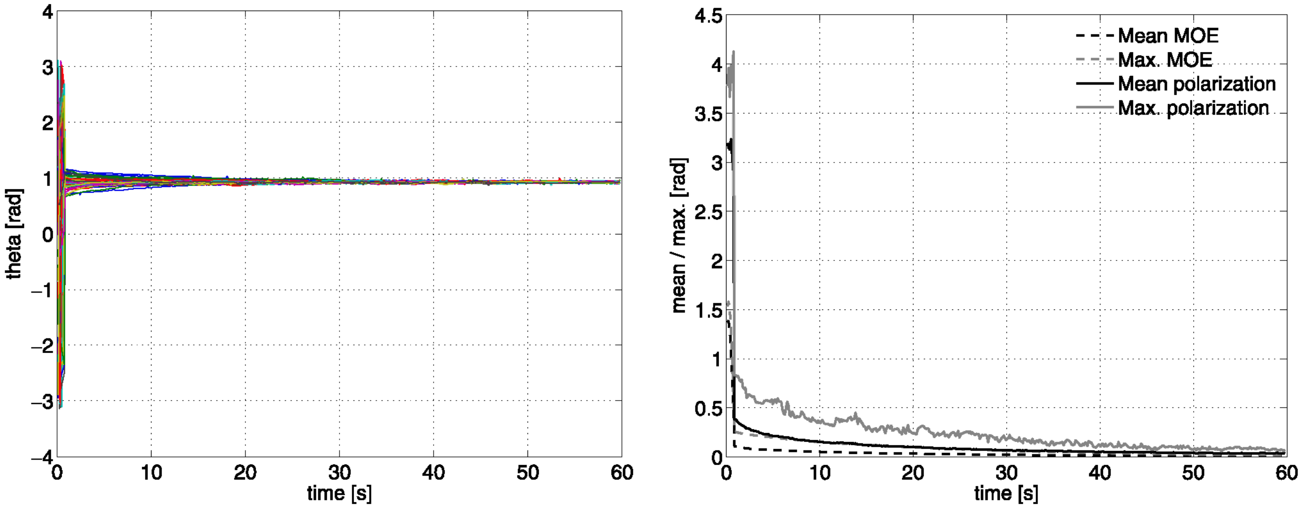
4.4 Set 4: CAB, *square lattice* topology, $\alpha = 0.1$, *random turn*

This set of experiments uses the CAB for consensus, with $\alpha = 0.1$. The communication topology is *square lattice* and the behavior *random turn*. The conditions are then the same as in Set 1, but applying CAB instead of CAA. Figure 8 shows the evolution of $\theta_i^{EXT}(t)$, $Po(t)$, $MOE(t)$, and the instantaneous maximum and the instantaneous mean of Po and MOE. It can be seen that in CAB agreement is faster than in CAA with the same parameters. In the worst case for $t > 2$ s MOE and Po are under 0.3 rad. This is due to the propagation of the θ_i of the *seed robot*.

Mean of Po for every experiment and for every temporal step after $t = 15$ s is 0.1119 rad, and the maximum is 0.2368 rad. For MOE the corresponding values are 0.0310 rad for the mean and 0.0461 rad for the maximum.

4.5 Set 5: CAB, *square lattice* topology, $\alpha = 0.01$, *random turn*

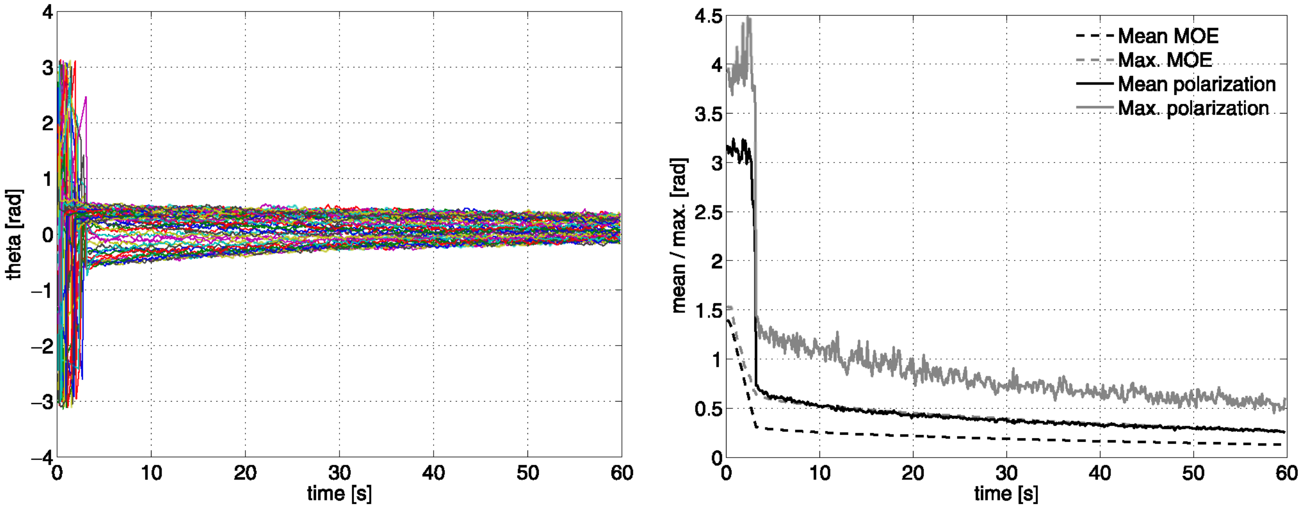
This set is the same as Set 4 but with $\alpha = 0.01$, so it uses CAB, the topology is *square lattice* and the behavior *random turn*. As it can be seen in Fig. 9(a), where the orientation reference estimates of every robot in a given experiment are plotted, the agreement is slower than in Set 4 ($\alpha = 0.1$), but



(a) Evolution of the individual reference orientations measured internally and translated to an external coordinate system ($\theta_i^{EXT}(t)$) for a given experiment.

(b) Instantaneous maximum and instantaneous mean of *Po* and *MOE* of all the experiments.

Fig. 9 Set 5: CAB, *square lattice* topology, $\alpha = 0.01$, *random turn*



(a) Evolution of the individual reference orientations measured internally and translated to an external coordinate system ($\theta_i^{EXT}(t)$) for a given experiment.

(b) Instantaneous maximum and instantaneous mean of *Po* and *MOE* of all the experiments.

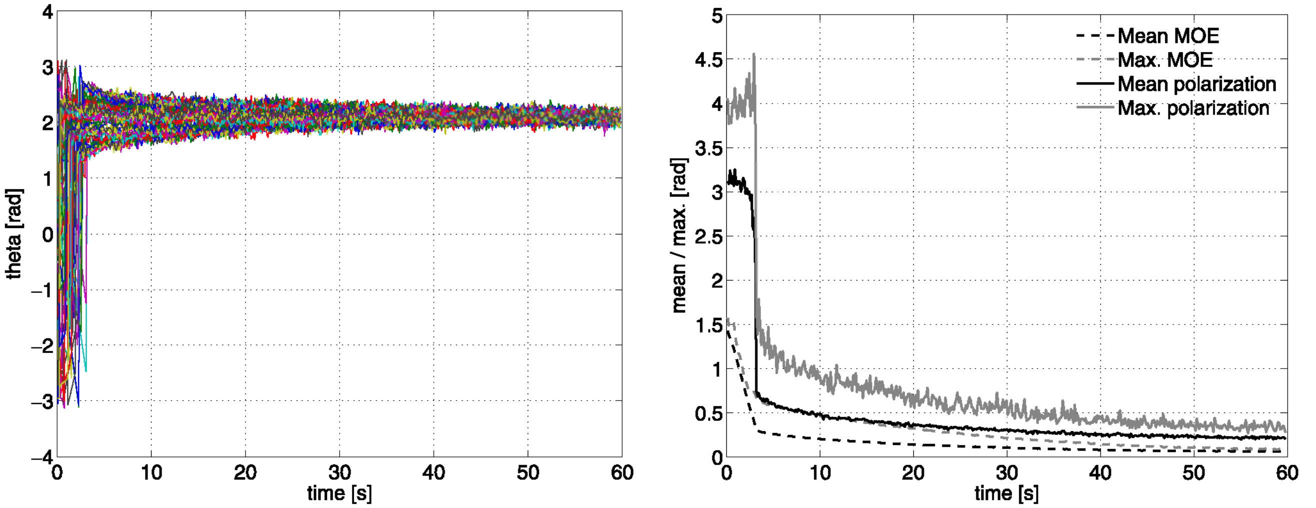
Fig. 10 Set 6: CAB, *ring* topology, $\alpha = 0.1$, *random turn*

it reaches smaller differences between the estimates. This can be seen in Fig. 9(b), where the instantaneous maximum and the instantaneous mean of *Po* and *MOE* are plotted.

The mean of *Po* for every experiment and temporal step after $t = 15$ s is 0.0639 rad, and the maximum 0.3260 rad. For *MOE* the corresponding values are 0.02075 rad for the mean and 0.1229 rad for the maximum. But as it can be seen in Fig. 9(b) the *MOE* and *Po* continue decreasing after $t = 15$ s, so a better characterization can be obtained calculating these mean and maximum values for $t > 50$ s. Then we have that the mean value of *Po* is 0.0395 rad and maximum 0.1062 rad. For *MOE* the mean is 0.0115 rad and the maximum 0.0269 rad. This values are much smaller than in Set 4.

4.6 Set 6: CAB, *ring* topology, $\alpha = 0.1$, *random turn*

This set of experiments has the same parameters as Set 4, *ring* topology, $\alpha = 0.1$, *random turn* behavior, but the topology is *ring*. It has also the same configuration as Set 2, but in this case robots use CAB instead of CAA. Figure 10(a) shows the evolution of the reference orientations translated



(a) Evolution of the individual reference orientations measured internally and translated to an external coordinate system ($\theta_i^{EXT}(t)$) for a given experiment.

(b) Instantaneous maximum and instantaneous mean of *Po* and *MOE* of all the experiments.

Fig. 11 Set 7: CAB, *ring* topology, $\alpha = 0.3$, *random turn*

to an external coordinate system ($\theta_i^{EXT}(t)$) for a particular experiment. As it can be seen robots agree on the orientation, but more slowly than in Set 4 (*square lattice* topology). In Fig. 10(b) the instantaneous maximum and the instantaneous mean of *Po* and *MOE* of all the experiments in the set are plotted. Comparing this figure to Fig. 8 of Set 4, we can see that the sudden steep change in the mean *Po* and the mean *MOE* occurs later in Set 6. This is due to the topology of Set 6, since the reference orientation propagates from the *seed robot*, jumping from two robots to other two different each step, so it takes longer to reach the last robot than with the *square lattice* topology where robots are more connected. In addition, after this abrupt drop in *MOE* and *Po* they continue decreasing and in $t = 60$ s these values are on average higher than in Set 4. It is again due to the less connected topology of Set 6.

The mean and maximum values of *Po* and *MOE* are not given for this set since at the end of the experiments ($t = 60$ s) *Po* and *MOE* are still decreasing. In addition, it is important to mention that the introduction of CAB allows the robots to agree on the reference orientation for the *ring* topology, which in the case of CAA was not possible.

4.7 Set 7: CAB, *ring* topology, $\alpha = 0.3$, *random turn*

This set experiments has the same configuration as Set 6, only α changes to $\alpha = 0.3$. This allows to take more into account the information about the reference orientation estimate of the neighboring robots. This might make the filter to agree faster. In Fig. 11 we can see that the steep drop in the means of *Po* and *MOE* occurs at the same time as in Set 6, but later on the values decrease much faster. As in Set 6, the mean and maximum values of *Po* and *MOE* are not given for this set since at the end of the experiments ($t = 60$ s) *Po* and *MOE* are still decreasing.

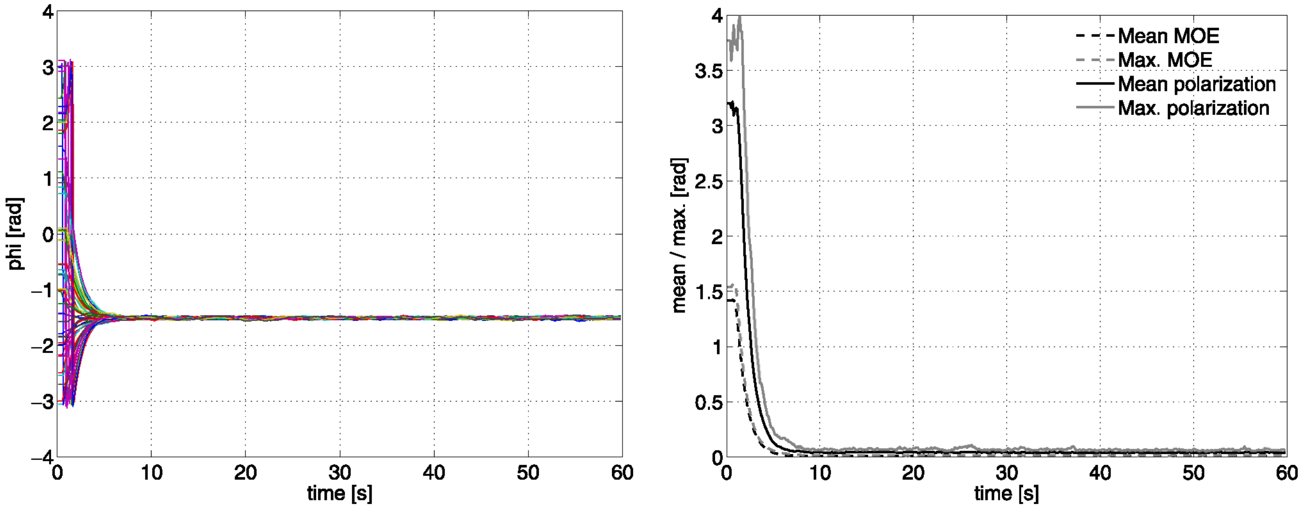
4.8 Set 8: CAB, *square lattice* topology, $\alpha = 0.1$, *alignment*

In this set robots perform the *alignment* behavior, placed in *square lattice* topology, using CAB with $\alpha = 0.1$. In Fig. 12 the evolution of the different robots' headings aligning with the reference orientation (ϕ_i) for a given experiment is shown, while the instantaneous maximum and the instantaneous mean of *Po* and *MOE* are plotted in Fig. 12(b). From them we can see that the deep drop in the mean and the maximum of *Po* and *MOE* occurs later than in Set 4 (same parameters except for the robots' behaviors), but the values after $t = 10$ s are smaller. As in Set 3, this is due to the alignment controller of the robots, that here not only filters the robots' headings (ϕ_i,), but also delays them compared to the reference orientation estimates.

The mean of *Po* for every experiment and for every temporal step after $t = 15$ s is 0.0419 rad, while the maximum is 0.1088 rad. For *MOE* these values are 0.0114 rad for the mean and 0.0180 rad for the maximum. These values are smaller than in the case of Set 4.

4.9 Set 9: CAB, *square lattice* topology, $\alpha = 0.1$, *random turn*, clock asynchronicity

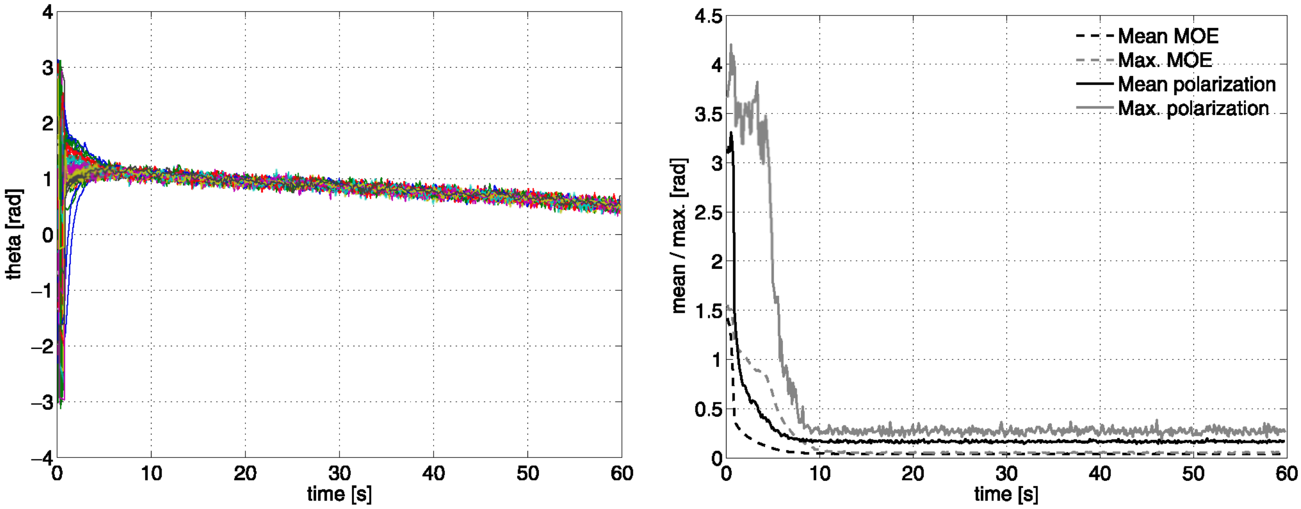
In this set asynchronicity in the robots' clocks is introduced to analyze its consequences. Besides this asynchronicity the characteristics of the experiments are the same as in Set 4:



(a) Evolution of robots' headings aligning with the reference orientation seen from an external coordinate system ($\phi_i(t)$) for a given experiment.

(b) Instantaneous maximum and instantaneous mean of Po and MOE of all the experiments.

Fig. 12 Set 8: CAB, *square lattice* topology, $\alpha = 0.1$, *alignment*



(a) Evolution of the individual reference orientations measured internally and translated to an external coordinate system ($\theta_i^{EXT}(t)$) for a given experiment.

(b) Instantaneous maximum and instantaneous mean of Po and MOE of all the experiments.

Fig. 13 Set 9: CAB, *square lattice* topology, $\alpha = 0.1$, *random turn*

CAB algorithm with $\alpha = 0.1$, *square lattice* topology and *random turn* behavior. The asynchronicity incorporates additive Gaussian noise of standard deviation $\sigma = 100$ ms to the clock of every robot. Results can be drawn from the graphs in Fig. 13, where the evolution of the individual reference orientations translated to an external coordinate system ($\theta_i^{EXT}(t)$) and the instantaneous maximum and the instantaneous mean of Po and MOE are plotted.

It can be seen in the shown experiment that robots agree on the reference orientation but that the value changes over time. For the shown experiment the variation is approximately of 0.01 rad/s. This drift on orientation agreement over time is not present in any of the synchronized experiments, so just from experimentation and without doing any mathematical analysis, we could think that it is caused by asynchronicity. But it should be proved mathematically and also better characterized with more sets of asynchronous experiments including different values of asynchronicity. For the application of collective movement such as the one described in Sect. 6, the value of the drift is small enough for the correct working of the framework. It will only make the group of robots not to move in a perfect straight line. On the other hand it could be a problem if orientation reference would be used for collective mapping applications, where the reference orientation would not be constant with time.

The mean of *Po* for every experiment and for every temporal step after $t = 15$ s is 0.1649 rad, while the maximum is 0.3818 rad. For *MOE* these values are 0.0400 rad for the mean and 0.0622 rad for the maximum. These values are slightly larger than in the case of Set 4. In addition, the drop in maximum *MOE* is delayed to $t = 8$, so this set takes more time to agree than Set 4.

4.10 Summary and discussion

A summary of the characteristics of sets carried out in simulation is presented in Table 1, as a reference for later discussion. In Fig. 14 boxplots of polarization and *MOE* results are presented for each set of experiments (except Set 2, Set 6 and Set 7), for every experiment for $t > 15$ s, in order to compare them. The numerical values of the mean and maximum values of polarization and *MOE* for every experiment and for every temporal step are shown for each of the sets of experiments in simulation (except for Set 2, Set 6 and Set 7) in Table 2. They are calculated for $t > 15$ s. Set 2 is not presented since not all the experiments agree on a reference orientation. Set 6 and Set 7 the agreement is slow and for $t = 15$ s robot have not yet agreed on the reference orientation and so the results can not be compared. Experiments with longer duration would be necessary for a better comparison.

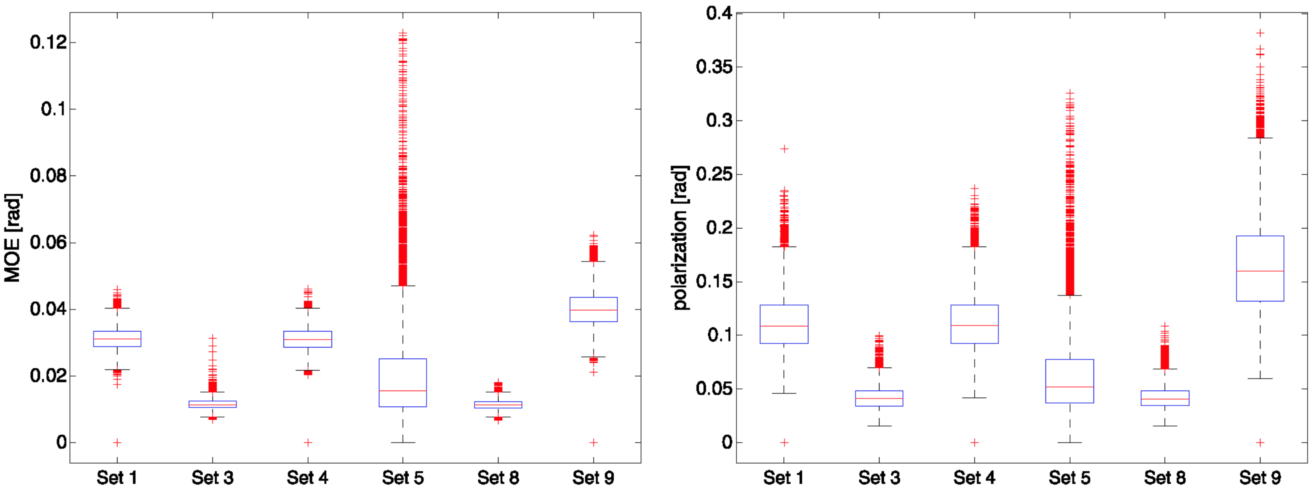
In order to see the effects of the α parameter we need to look at Set 5 ($\alpha = 0.01$) and Set 7 ($\alpha = 0.3$) and compare them with the equivalent cases in the rest of parameters but different α. Set 6 ($\alpha = 0.1$) and Set 7 are identical in their parameters except α, but statistical results can not be used because for $t > 15$ s robots have not yet agreed. But looking to Fig. 10(b) and Fig. 11(b) it can be seen that the agreement occurs faster in the case of $\alpha = 0.3$, as it was previously predicted. Set 4 ($\alpha = 0.1$) and Set 5 have the same characteristics except α and can be compared looking at Fig. 8(d) and Fig. 9(b). As seen before, in the case of larger α the

Table 1 Summary of the characteristics of the sets performed in simulation

	Alg.	Topology	α	Behavior	Asynch.
Set 1	CAA	*square l.*	0.1	*rand. turn*	No
Set 2	CAA	*ring*	0.1	*rand. turn*	No
Set 3	CAA	*square l.*	0.1	*alignment*	No
Set 4	CAB	*square l.*	0.1	*rand. turn*	No
Set 5	CAB	*square l.*	0.01	*rand. turn*	No
Set 6	CAB	*ring*	0.1	*rand. turn*	No
Set 7	CAB	*ring*	0.3	*rand. turn*	No
Set 8	CAB	*square l.*	0.1	*alignment*	No
Set 9	CAB	*square l.*	0.1	*rand. turn*	100 ms

Table 2 Mean and maximum values of *Po* and *MOE* for every experiment and time steps, for the different sets of experiments in simulation

	Mean *MOE* (rad)	Max *MOE* (rad)	Mean *Po* (rad)	Max *Po* (rad)
Set 1	0.0311	0.0459	0.1118	0.2736
Set 3	0.0115	0.0313	0.0420	0.0997
Set 4	0.0310	0.0461	0.1119	0.2368
Set 5	0.0207	0.1229	0.0639	0.3260
Set 8	0.0114	0.0180	0.0419	0.1088
Set 9	0.0400	0.0622	0.1649	0.3818



(a) Boxplot representing the different the results of MOE for the different sets.

(b) Boxplot representing the different the results of polarization for the different sets.

Fig. 14 Boxplots for polarization and MOE in simulation sets. Each boxplot was created using 10500 measures

agreement is faster. In addition, looking at these graphs for $t > 50$ s and to the statistics of Set 5 also for $t > 50$ s, it can be observed that case of $\alpha = 0.01$ (Set 5) has more accuracy, as it was expected in theory. Results can not be directly compared for $t > 15$ s since Set 5 is still agreeing on the reference orientation while Set 6 is already stable. Longer experiments would be desirable in order to do a better comparison.

CAA and CAB can be compared looking at the polarization and *MOE* results. Set 1 and Set 4 are the same except on the algorithm they use, and the same happens for the pair Set 3 and Set 8. Looking at the boxplots (Fig. 14) and to the summary of results (Table 2) it can be seen that both *MOE* and polarization are identical when comparing Set 1 with Set 4 and Set 3 with Set 8. It could be expected since after the value of the *seed* robot is propagated the algorithms work in the same manner. The main difference can be observed in the first seconds of workings of both algorithms, where in CAB exists a fast initial agreement. It can be seen comparing Fig. 5(d) with Fig. 8(d), and Fig. 7(b) with Fig. 12(b).

When looking at the influence of the topology used we can look at Set 4 (*square lattice* topology) and Set 6 (*ring* topology) since they are identical except in the connectivity of the robots. Looking at the evolution of *MOE* and polarization in Fig. 8(d) and Fig. 10(b) two relevant observations can be done. First, in the *ring* topology the initial agreement done thanks to the propagation of the *seed* robot orientation reference is slower than in the *square lattice* topology, due to the its less connectivity. Second, the same happens for the later agreement due to the operation of CAB with already every $c_i = 1$.

If we want to determine the differences between the behaviors of the robots (*alignment* vs *random turn*), Set 1 can be compared to Set 3 and Set 4 to Set 8. If we look at the boxplots (Fig. 14) and to the summary of results (Table 2) it can be observed that both *MOE* and polarization are slightly smaller in the case of *alignment* behavior. This smaller value might be due to a more static behavior of the robots once they are aligned that avoids the effects of odometry plus communication delays.

One of the effects of asynchronicity is the variation with time of the reference estimation, as it was previously mention. Furthermore, if we compare Set 1 with Set 9, which have the same parameters and are only different in the synchronicity, it can be observed that *MOE* and polarization are larger in the asynchronous case. It can be seen both looking at the boxplots (Fig. 14) and to the statistical summary (Table 2).

5 Experiments with real robots

The execution of experiments just in simulation is not enough for the validation of robotic experiments, therefore systematic experiments with eight real robots were carried out. Khepera III robots were used in the experiments. They are provided with a relative positioning system, to allow for the estimation of the relative position of nearby robots together with an identification number for each robot (Pugh et al. 2009).

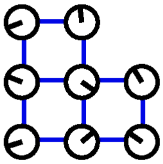


Fig. 15 Communication topology of type *square lattice* used in the experiments with real robots

Two different sets of experiments with real robots were run, each one composed of 20 different experiments, with the same parameters but different random initial headings of the robots. Both sets are only differentiated by the type of consensus algorithm used: CAA in one case and CAB in the other. The behavior of the robots is *alignment*, and so the evolution of the experiments can be determined externally by looking at the robot headings. Other behaviors were not tested in the case of real robots since there is no possibility of inner inspection of the robots' states in a synchronized manner. Robots are placed to form a *square lattice* topology, consisting of two rows of three robots and a third one of two robots, depicted in Fig. 15 indicating the communication graph by the blue connection lines.

In simulation, robots perform a complete cycle of the algorithm in a fixed period of 128 ms but in reality this period varies from cycle to cycle and robot to robot. This period was measured in several experiments and it was in most cases around 180 ms. The difference between both periods is given by the simulator, imposing a simulation step multiple of 64 ms. The period in simulation was chosen by rounding down the 180 ms of real robots. The duration of each experiment is 60 s, the same as in the simulation experiments. In the experiments using CAB, the *seed robot* was placed at the bottom left corner of the lattice according to Fig. 15.

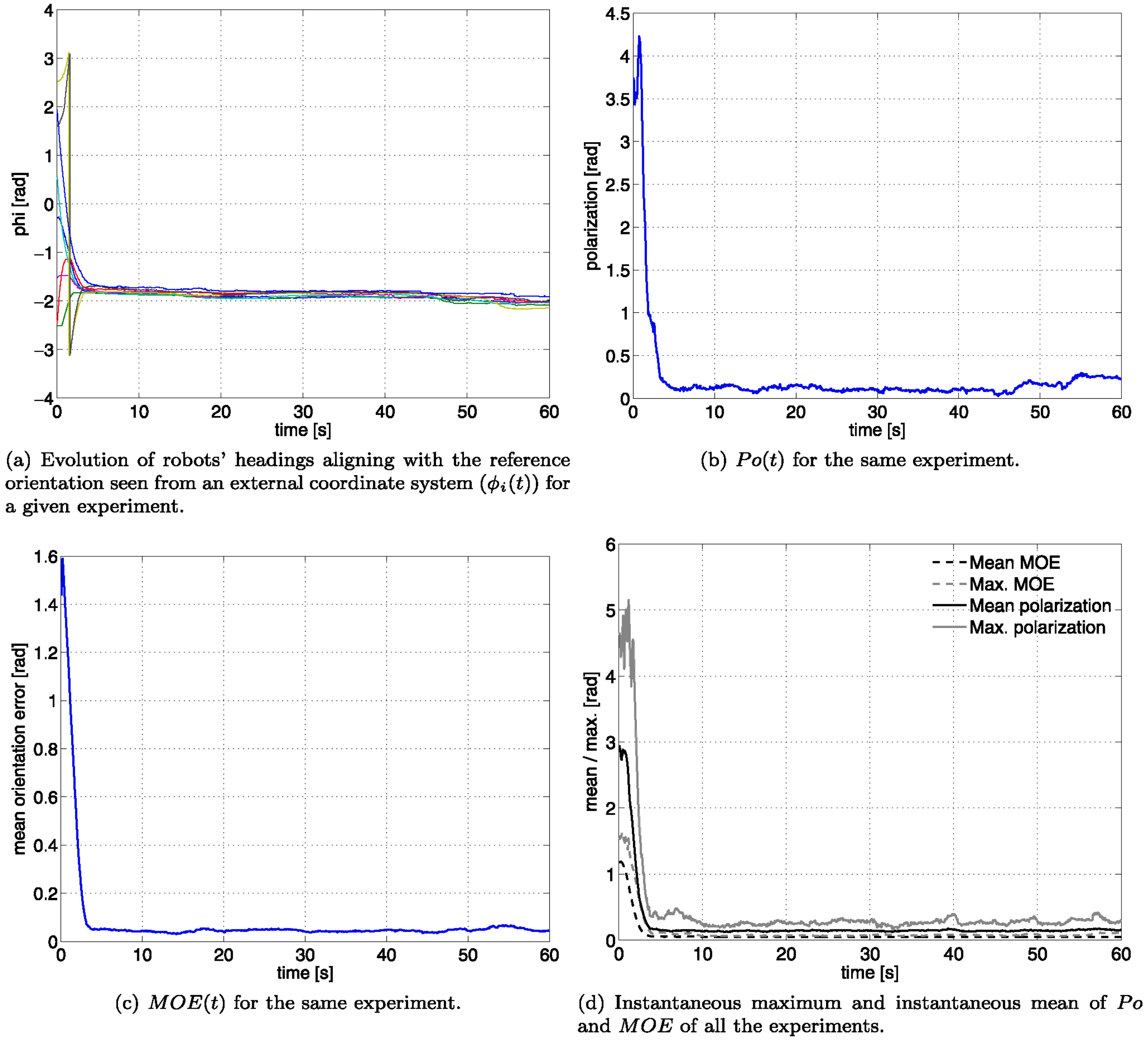
The same metrics used in the simulation experiments to measure the error in the orientation agreement are calculated in the real robots experiments: *Mean Orientation Error* (*MOE*) and *Polarization* (*Po*). Moreover, the evolution with time of robots' headings aligning with the reference orientation seen from an external coordinate system ($\phi_i(t)$) is shown in the two sets for an experiment.

Every experiment was recorded using an overhead camera for a later extraction of the robots' headings with a period of 40 ms. For this aim, a two colored marker was placed on top of each robot. The videos were processed using the tracking tool *SwisTrack* (Lochmatter et al. 2008). Image segmentation is not perfectly made and there is also discretization of pixels. Both introduce an error in the measures which can be observed in the graphs representing the evolution of $\phi_i(t)$ with time.

As has been explained, the proposed algorithms need synchronization between the robots. Synchronization is implicit in simulation but a mechanism allowing it must be performed in real robots. Several synchronization mechanisms exist, some really distributed and oriented towards wireless networks of the type that the set of robots form. Finally, the Network Time Protocol (NTP) (Mills 1992), the Internet standard for time synchronization, was used because it is already implemented in the Linux distribution present in the robots. An external computer was used as NTP server, and robots synchronized with it at the beginning of the experiments. Its accuracy is in the millisecond scale, drifts are usually under 20 ms. Due to the distributed nature of the problem it is not easy to measure quantitative how synchronous are the robots, but given the NTP specifications we will consider that asynchronicity does not exceed 20 ms.

5.1 Set 1: CAA, *square lattice* topology, $\alpha = 0.1$, *alignment*

In this set of experiments robots apply CAA as consensus algorithm, with $\alpha = 0.1$. The behavior is alignment of its heading with the reference orientation estimate, this is the same case as Set 3 of the simulation experiments but with eight robots. The placement topology is *square lattice*. In Fig. 16(a), the evolution of the headings of the robots is plotted for a given experiment of this set. It can be seen that after time $t = 5$ s the difference between them remains



(a) Evolution of robots' headings aligning with the reference orientation seen from an external coordinate system ($\phi_i(t)$) for a given experiment.

(b) $Po(t)$ for the same experiment.

(c) $MOE(t)$ for the same experiment.

(d) Instantaneous maximum and instantaneous mean of Po and MOE of all the experiments.

Fig. 16 Set 1: CAA, *square lattice* topology, $\alpha = 0.1$, *alignment*, real robots

bounded. This can be better understood looking at Fig. 16(b) and Fig. 16(c) where *Po* and *MOE* are plotted with time for the same experiment. $Po(t)$ for $t > 5$ s remains under 0.3 rad and $MOE(t)$ under 0.1 rad.

In Fig. 16(d) the instantaneous maximum and the instantaneous mean of *Po* and *MOE* of all the experiments of this set are shown. As it is shown in the figure, after $t = 5$ s, the instantaneous maximum of *Po* remains under 0.5 rad, and the instantaneous maximum of *MOE* remains under 0.15 rad, which shows that the in the 20 experiments robots agree on the reference orientation. For $t > 15$ s, the mean value of *Po* for every experiment and for every temporal step is 0.1458 rad and the maximum 0.4195 rad. In the case of *MOE* this mean value is 0.0480 rad and the maximum 0.1137 rad.

If this set is compared to its equivalent in simulation (Set 3, Fig. 7), it can be seen that the real robots agree much faster, since after 5 s *Po* and *MOE* are under 0.5 rad and 0.15 rad respectively, while simulated robots take more than 10 s to agree. This might be due to the difference in the number of robots, since here there are eight robots instead of 49. On the other hand, the maximum values of *MOE* and *Po* and the mean value of *MOE* and *Po* for every experiment and for every temporal step is higher in the case of real robots. This can be due to a non perfect model of the relative positioning system in simulation, but also to noise introduced in the tracking system. This idea is corroborated by the fact that robots in the video do not oscillate but in the plotting we see that robots do.

The variation of the heading with time due to the lack of perfect synchronicity, as was shown in Set 9 of simulation experiments, is almost negligible in the case of the experiment shown. The experiment of the set that has the largest variation of heading is approximately of 0.01 rad/s.

5.2 Set 2: CAB, *square lattice* topology, $\alpha = 0.1$, *alignment*

In this set robots perform the *alignment* behavior, placed in *square lattice* topology, using the CAB algorithm with $\alpha = 0.1$. The set is the same as Set 1 with real robots but using CAB algorithm, and the same case as Set 8 in simulation but using eight real robots instead of 49 simulated robots. In Fig. 17(a), the evolution of robots' headings aligning with the reference orientation ($\phi_i(t)$) for a given experiment is shown. *Po* and *MOE* are plotted with time for the same experiment in Fig. 17(b) and Fig. 17(c).

The instantaneous maximum and the instantaneous mean of *Po* and *MOE* are plotted in Fig. 17(d). From it we can see that after only 3 s these values are small, showing the fast consensus in the 20 experiments of the set. The mean of *Po* for every experiment and for every temporal step after $t = 15$ s is 0.1470 rad, while the maximum is 0.4580 rad. For *MOE* these values are 0.0489 rad for the mean and 0.0972 rad for the maximum. These mean values are very similar to Set 1 with real robots, which is normal since CAA and CAB behave very similar after agreement.

If this set is compared to Set 1 of experiments with real robots it can be seen that its consensus is reached faster due to the nature of CAB in comparison to CAA.

If comparison is made between this set and Set 8 in simulation it is observed that agreement in this set is faster but the error in heading agreement is higher. The same thing happened when comparing Set 1 of real robots with Set 3 in simulation, being the reasons for it the same.

The variation of heading with time because of non synchronicity between the robots is for the shown experiment of Fig. 17(a) of about 0.007 rad/s on average. The experiment of the set that has the largest variation of heading is approximately of 0.01 rad/s, as in the case of Set 1.

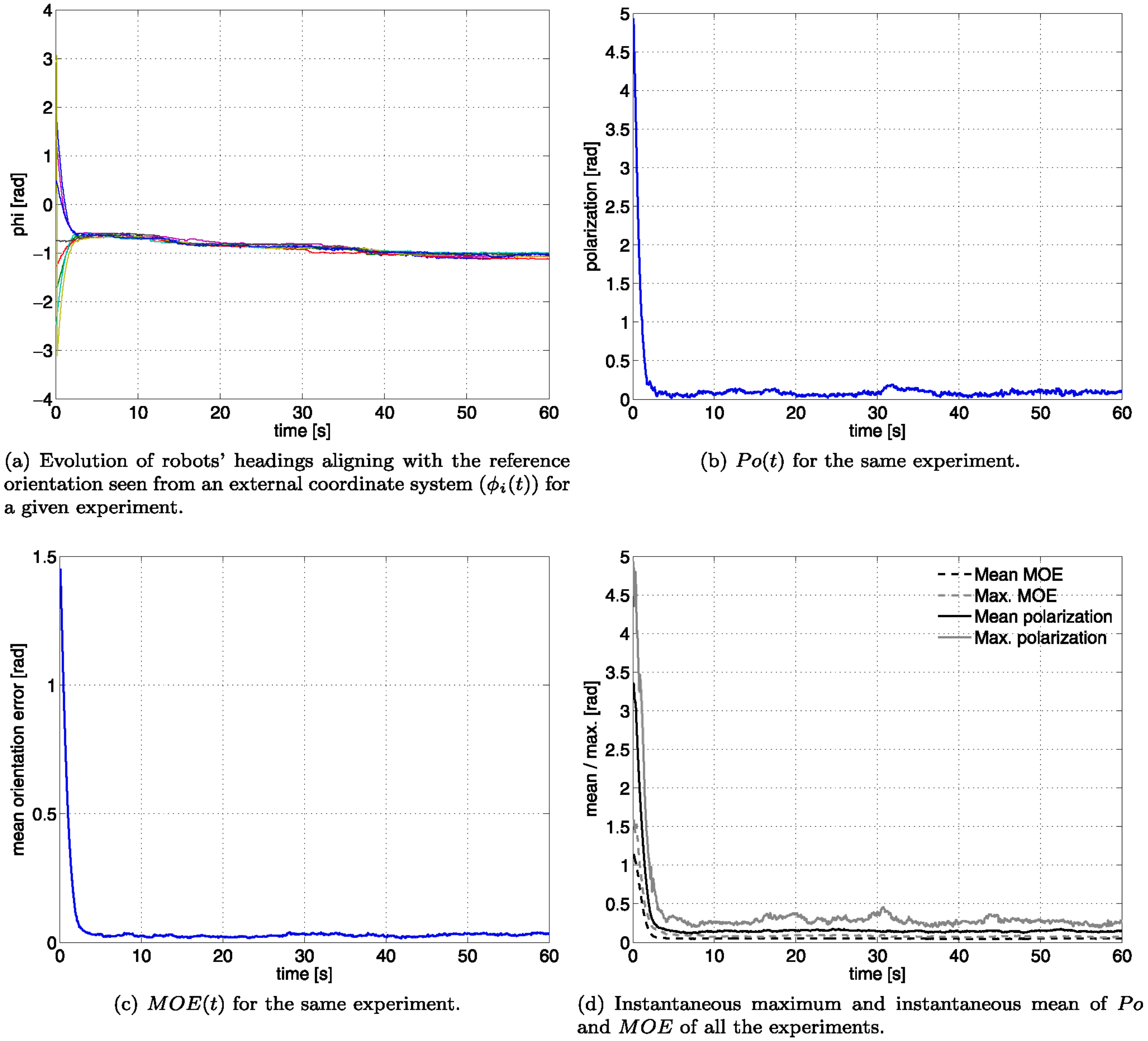
5.3 Summary and discussion

The results of the two sets in simulations show a very similar performance as in the real robots cases. The most important difference is that in the real robot case the agreement is faster, but it is due to the reduced number of robots used. In addition, and like in Set 9 of simulation with asynchronicity there is a variation over time in the estimation of the reference orientation, due to a non perfect real synchronization. As in the simulation case, the characteristics of sets carried out with real robots are summarized in Table 3. Also, Fig. 18 shows boxplots of polarization and *MOE* results each set of experiments for every experiment for $t > 15$ s in order to compare them. In Table 4, the mean and maximum values of *Po* and *MOE* for every experiment and for every temporal step are shown for each of the sets of experiments in with real robots. They are calculated for $t > 15$ s.

The only difference between Set 1 and Set 2 with real robots is the consensus algorithm used (CAA and CAB). As it was said when discussing CAA versus CAB in simulation, it can be seen looking at the evolution of *MOE* and polarization with time in Fig. 16(d) and Fig. 17(d), that the agreement occurs faster in the CAB case thanks to the propagation of the estimate orientation of the *seed* robot. Furthermore, looking at the statistical results of *MOE* and polarization summarized in Table 4 and the boxplots of Fig. 18, it can be observed that the accuracy of both algorithms is the same.

6 Application case: collective movement of mobile robots

The distributed orientation agreement method described in this article has been used as a key part of a framework for



(a) Evolution of robots' headings aligning with the reference orientation seen from an external coordinate system ($\phi_i(t)$) for a given experiment.

(b) $Po(t)$ for the same experiment.

(c) $MOE(t)$ for the same experiment.

(d) Instantaneous maximum and instantaneous mean of Po and MOE of all the experiments.

Fig. 17 Set 2: CAB, *square lattice* topology, $\alpha = 0.1$, *alignment*, real robots

the collective movement of mobile robots by the same authors (Navarro and Matía 2011). This framework allows a group of robots to move as a unique entity, being able to avoid obstacles, forming a non-perfect triangular lattice with a desired inter-robot distance, and deciding collectively the direction of movement and speed. It is based on three key elements:

Distributed orientation agreement mechanism. The method under study in this article.

Controllers. They control the movements of the robots. The *Flocking Controller* is distributed among the robots, making them to move as a group in a determined direction while maintaining a desired inter-robot distance. It requires the robots to share a common reference orientation given by the *Distributed Orientation Agreement Mechanism*. The *Safety Controller* is in charge of avoiding collisions with very near obstacles.

Modules. They modify one or more of the parameters of the *Flocking Controller* in order to allow obstacle avoidance, speed control or changing inter-robot distance. They are based on the *Distributed Consensus Mechanism*, that allows information and decisions to be spread among the robots agreeing on a parameter of the *Flocking Controller*.

Robots agree, using wireless messages, on the direction of movement expressed in relation to the reference orientation θ_i. So they do not need to estimate other robots'

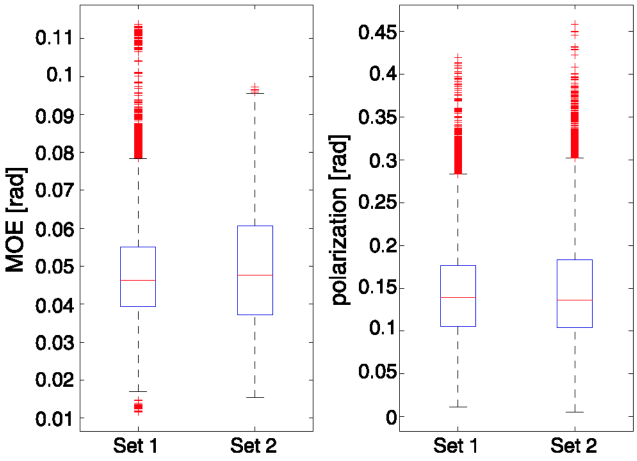


Fig. 18 Boxplots representing the different the results of *MOE* and polarization for the different sets in real robot experiments. Each boxplot was created using 22500 measures

Table 3 Summary of the characteristics of the sets performed with real robots

	Alg.	Topology	α	Behavior	Asynch.
Set 1	CAA	*square l.*	0.1	*alignment*	No
Set 2	CAB	*square l.*	0.1	*alignment*	No

Table 4 Mean and maximum values of *Po* and *MOE* for every experiment and time steps, for the different sets of experiments with real robots

	Mean *MOE* (rad)	Max *MOE* (rad)	Mean *Po* (rad)	Max *Po* (rad)
Set 1	0.0480	0.1137	0.1458	0.4195
Set 2	0.0489	0.0972	0.1470	0.4580

velocities by observation, which is a difficult task in real robots. This represents a great advantage, and allow the robots to make fast changes in the direction of movement of the group, thanks to the use of θ_i as virtual compass.

In addition, this framework and its systematic experiments described in the article constitute an indirect test on the workings of the distributed orientation agreement method. Robots are connected forming a non-perfect dynamic triangular lattice, where sometimes some links are broken and also new connections appear, at the same time that they move in the space.

The framework can be implemented on any mobile robot capable of estimating the relative positions of nearby robots and communicating with them. It was successfully tested using 8 real robots and in simulation with up to 40 robots.

7 Conclusions

In this article, a method for the agreement of a set of robots on a common reference orientation based on a distributed algorithm has been described. Robots make only use of a relative positioning system, capable of detecting the positions of nearby robots, and local communications. Two different consensus algorithms based on the exchange of information were proposed, tested and analyzed.

In order to test the method, systematic experiments were carried out in simulation and with real robots, varying the consensus algorithm, the α parameter of the consensus algorithm, and the communication topology. The CAA consensus algorithm diverged in certain experiments when robots were placed in *ring* topology. The success depended on the initial conditions. On the other hand, in CAB there was agreement in every experiment for all the sets of experiments run. Its convergence has not been proved mathematically, but experimental results show that the robots are able to agree on the reference orientation under certain conditions. A total of 310 experiments of 60 s were performed: 270 of them in simulation and the remaining 40 using real robots.

Scalability with increasing number of robots, allowed by local communications and sensing, was tested successfully in simulation with up to 49 robots. All experiments with real robots succeeded proving that the proposed method works with real robots. Experiments with a large group of real robots were not carried out, but the mentioned scalability in simulation and the nature (distributed, local communications and sensing) of the algorithms indicates that it would work properly.

It has been seen that the use of large values of α makes agreement faster but the error in the orientation agreement is larger. This is not surprising since it implies a *faster* filter. Consequently, small values of α imply slow agreement but smaller error.

The method assumes synchronicity of the robots' clocks. Experiments in simulation purposely introduced delays among the robots' clock, and experiments with real robots involve imperfect synchronicity. This revealed that the algorithms worked properly, and robots were able to agree on the reference orientation, but that these values did not always keep stable varying slowly with time. The maximum variation measured was 0.01 rad/s.

One of the advantages of these methods is that if robots already posses a system for estimating the relative position of nearby robots, specific hardware is not required. In principle, the algorithm could work on any robot capable of detecting the position of other robots and communicating with them.

The presence of a common orientation reference among all the robots of a group can be quite useful for some applications. It can be used as basis for flocking or other collective movement algorithms. The variation of the reference orientation with time because robots' clocks might not be synchronized can be compatible with the use of the reference orientation in collective movement algorithms, if that variation is in the order of magnitude of the values measured in the experiments.

8 Future work

There are some works that in the future can complete and improve the development described in this article. The algorithm could be modified to make it adaptive, being able to change the α parameter, having a higher value in the beginning for a fast agreement, and a smaller one later on for more accuracy.

It has been seen that the topology has a special relevance in the results of the algorithm, specially in the case of CAA algorithm were some of the experiments diverged for the case of *ring* topology. In addition, the agreement is much slower in the *ring* topology than in the *square lattice* topology. It would be interesting to study triangular lattice topologies, line topology and topologies very connected. Furthermore, in most of the applications, as the collective movement seen in Sect. 6, in which robots shift the positions and the topology of the communication network is dynamic. Thus, the study of these static networks is not enough to completely characterize the method and dynamic topologies should be studied.

It would also be important to study what would happen if two different groups of robots would gather at a certain moment, since the confidences c_i of all of them would be already set to 1. One possible solution would be that one of the groups have a higher confidence value than the other imposing its reference orientation to the other. The main disadvantage of this solution is that robots must have an identification group number. A better solution is a modification of the CAB algorithms in which robots do not belong explicitly to any group but act randomly as *seed robot*. Each robot once every T s would decide randomly with a low probability if it behaves as a *seed robot*. During a short period of time, shorter than T s, the temporary *seed robot* will increase its confidence c_i. Then the reference orientation of the temporary *seed robot* will be imposed to the rest of the robots. If robots are already on agreement the estimation of this robot will not be very different from the others, and the change in the reference orientation will not be large. On the other hand, if there is no agreement (because to group of robots just found each other) it will help to impose the reference orientation of the temporary *seed robot* to all the robots. Later on every robot should decrease automatically its confidence in order to be able to react to new temporary *seed robots*.

The method could also be complemented to add an origin point $(0, 0)$, which would imply having a complete coordinate system. Robots would need to keep track of all the odometry values (not only θ), and also to exchange messages with this information and the relative distances among them. Probably Kalman filters or a similar algorithm would help in its implementation.

References

Alanyali, M., Venkatesh, S., Savas, O., & Aeron, S. (2004). Distributed Bayesian hypothesis testing in sensor networks. In *Proceedings of the 2004 American control conference* (Vol. 6, pp. 5369–5374).

Bauso, D., Giarre, L., & Pesenti, R. (2003). Distributed consensus protocols for coordinating buyers. In *Proceedings of the 42nd IEEE conference on decision and control*, Maui, Hawaii (Vol. 1, pp. 588–592).

Borenstein, J., Everett, R., & Feng, L. (1996). *Where am I? Systems and methods for mobile robot positioning*. Ann Arbor: University of Michigan Press.

Carling, C., Svenson, P., Mårtenson, C., & Carlsen, H. (2003). A flock-based model for ad hoc communication networks. In *Eight international command and control research and technology symposium*, Washington, DC, USA.

Çelikkanat, H., Turgut, A. E., & Şahin, E. (2009). Heidelberg, chap guiding a robot flock via informed robots. In *Distributed autonomous robotic systems 8* (pp. 215–225). Berlin: Springer.

Chuang, Y. L., Huang, Y., D'Orsogna, M., & Bertozzi, A. (2007). Multi-vehicle flocking: scalability of cooperative control algorithms using pairwise potentials. In *2007 IEEE international conference on robotics and automation*.

Denantes, P. (2007). *Performance of averaging algorithms in time-varying networks*. Master's thesis, Ecole Polytechnique Fédérale de Lausanne.

Flocchini, P., Prencipe, G., Santoro, N., & Widmayer, P. (2008). Arbitrary pattern formation by asynchronous, anonymous, oblivious robots. *Theoretical Computer Science*, *407*, 412–447.

Gutiérrez, A., Campo, A., Dorigo, M., Donate, J., Monasterio-Huelin, F., & Magdalena, L. (2009). Open e-puck range and bearing miniaturized board for local communication in swarm robotics. In *Proceedings of the IEEE international conference on robotics and automation*, Piscataway, NJ (pp. 3111–3116). New York: IEEE Press.

Jadbabaie, A., Lin, J., & Morse, A. S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, *48*(6), 988–1001.

Lee, G., & Chong, N. Y. (2006). Decentralized formation control for a team of anonymous mobile robots. In *The 6th Asian control*, Bali, Indonesia.

Lochmatter, T., Roduit, P., Cianci, C., Correll, N., Jacot, J., & Martinoli, A. (2008). SwisTrack—a flexible open source tracking software for multi-agent systems. In *IEEE/RSJ 2008 international conference on intelligent robots and systems (IROS 2008)*, IEEE (pp. 4004–4010).

Martinson, E., & Payton, D. (2005). Lattice formation in mobile autonomous sensor arrays. In E. Şahin & W. Spears (Eds.), *Swarm robotics workshop: state-of-the-art survey* (pp. 98–111). Berlin: Springer.

Mehyar, M., Spanos, D., Pongsajapan, J., Low, S., & Murray, R. (2005). Distributed averaging on asynchronous communication networks. In *44th IEEE conference on decision and control, 2005 and 2005 European control conference*. CDC-ECC '05 (pp. 7446–7451).

Mills, D. L. (1992). *Network time protocol (version 3) specification, implementation and analysis (rfc 1305)* (Tech. rep.). University of Delaware.

Navarro, I., & Matía, F. (2009). A proposal of a set of metrics for collective movement of robots. In *Proceedings of workshop on good experimental methodology in robotics, robotics science and systems*.

Navarro, I., & Matía, F. (2011). A framework for collective movement of mobile robots based on distributed decisions. *Robotics and Autonomous Systems, 59*(10), 685–697.

Navarro, I., Gutiérrez, A., Matía, F., & Monasterio-Huelin, F. (2008). An approach to flocking of robots using minimal local sensing and common orientation. In E. Corchado, A. Abraham & W. Pedrycz (Eds.), *Lecture notes in artificial intelligence (LNAI): Vol. 5271. Hybrid artificial intelligent systems: 3th International Workshop, HAIS 2008* (pp. 616–624). Berlin: Springer.

Navarro, I., Pugh, J., Martinoli, A., & Matia, F. (2009). A distributed scalable approach to formation control in multi-robot systems. In *Distributed autonomous robotic systems 8* (pp. 203–214). Berlin: Springer.

Ojeda, L., & Borenstein, J. (2000). Experimental results with the kvh c-100 fluxgate compass in mobile robots. In *Proceedings of the IASTED international conference robotics and applications 2000*, Honolulu, Hawaii, USA.

Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Transactions on Automatic Control, 51*, 401–420.

Olfati-Saber, R., & Murray, R. M. (2003). Consensus protocols for networks of dynamic agents. In *American control conference* (Vol. 2, pp. 951–956).

Pugh, J., Raemy, X., Favre, C., Falconi, R., & Martinoli, A. (2009). A fast on-board relative positioning module for multi-robot systems. In *IEEE/ASME transactions on mechatronics, focused section on mechatronics multi robot systems*.

Ren, W., Beard, R. W., & Atkins, E. M. (2007). Information consensus in multivehicle cooperative control. *IEEE Control Systems Magazine, 27*, 71–82.

Reynolds, C. W. (1987). Flocks, herds, and schools: a distributed behavioral model. *Computer Graphics, 21*(4), 25–34.

Talebi, M. S., Kefayati, M., Khalaj, B. H., & Rabiee, H. R. (2006). Adaptive consensus averaging for information fusion over sensor network. In *The third IEEE international conference on mobile ad-hoc and sensor systems (MASS'06)*.

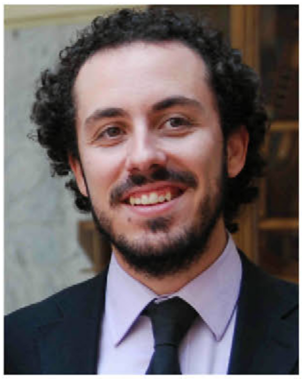
Tanner, H., Jadbabaie, A., & Pappas, G. (2003a). Stable flocking of mobile agents, part i: fixed topology. In *Proceedings. 42nd IEEE conference on decision and control* (Vol. 2, pp. 2010–2015).

Tanner, H., Jadbabaie, A., & Pappas, G. (2003b). Stable flocking of mobile agents, part ii: dynamic topology. In *Proceedings. 42nd IEEE conference on decision and control* (Vol. 2, pp. 2010–2015).

Turgut, A. E., Çelikkanat, H., Gökçe, F., & Şahin, E. (2008). Self-organized flocking in mobile robot swarms. *Swarm Intelligence, 2*(2–4), 97–120.

Xiao, L., & Boyd, S. (2003). Fast linear iterations for distributed averaging. In *42nd IEEE conference on in decision and control*, Hawaii (pp. 4997–5002).

Xiao, L., Boyd, S., & Lall, S. (2005). A scheme for robust distributed sensor fusion based on average consensus. In *International conference on information processing in sensor networks*, Los Angeles (pp. 63–70).



Iñaki Navarro received the M.S. degree in telecommunication engineering in 2004, and the Ph.D. degree in robotics and automation in 2010, both from Universidad Politécnica de Madrid (UPM), Madrid, Spain. He is currently Post-Doctoral Fellow at Laboratorio de Sistemas de Control, ETSI Telecomunicación, UPM. In 2006 he was a visiting Ph.D. student at the Distributed Intelligent Systems and Algorithms Laboratory (DISAL) at the École Polytechnique Fédérale de Lausanne (EPFL) Switzerland. His research interests include formation control of mobile robots, swarm robotics, distributed consensus of agents and mobile robotics.

Fernando Matía is Full Professor at Universidad Politécnica de Madrid since 2010, having being Associate Professor since 1997. He received the Ph.D. from that University in 1994, being engaged in control engineering since 1988, and in robotics since 1994. His main R&D activities are fuzzy control and mobile robotics. He has experience in eight European projects and 17 national research and industrial projects related with the previous topics. He is the author of three books, 16 book chapters, 26 articles in journals, and 75 papers in conferences, being reviewer of 12 international journals. He has also been Deputy Vice-Director of Studies, Vice-Director of Quality Management and Vice-Director of Studies of the High School of Industrial Engineering at Universidad Politécnica de Madrid.