

Robot Convergence via Center-of-Gravity Algorithms

Reuven Cohen and David Peleg

Department of Computer Science and Applied Mathematics,
The Weizmann Institute of Science, Rehovot 76100, Israel
E-mail: {r.cohen,david.peleg}@weizmann.ac.il

Abstract. Consider a group of N robots aiming to converge towards a single point. The robots cannot communicate, and their only input is obtained by visual sensors. A natural algorithm for the problem is based on requiring each robot to move towards the robots' center of gravity. The paper proves the correctness of the center-of-gravity algorithm in the semi-synchronous model for any number of robots, and its correctness in the fully asynchronous model for two robots.

1 Introduction

1.1 Background and motivation

In hazardous or hostile environments, it may be desirable to employ large groups of low cost robots for cooperatively performing various tasks. This approach has the advantage of being more resilient to malfunction and more configurable than a single high cost robot. Consequently, autonomous mobile robot systems have been studied in different contexts, from engineering to artificial intelligence (e.g., [1–11]). A survey on the area is presented in [12, 13].

During the last decade, various issues related to the coordination of multiple robot systems have been studied from the point of view of distributed computing (cf. [14–20]). The focus is on trying to model an environment consisting of mobile autonomous robots, and studying the capabilities the robots must have in order to achieve their common goals. A number of computational models were proposed in the literature for such systems. In this paper we follow the models of [19–21]. The robots are identical and indistinguishable, cannot communicate between them, and do not operate continuously. They wake up at unspecified times, observe their environment using sensors which are capable of identifying the locations of the other robots, perform a local computation determining their next move and move accordingly.

To model the behavior of the robots, several activation scheduling models have been suggested. In the synchronous model, all robots are active at every cycle. In the semi-synchronous model some robots are active at each cycle. In the asynchronous model, no cycles exist and no limit is placed on the latency of each robot movement. The models are presented more formally in the next section.

Two basic coordination tasks in autonomous mobile robot systems that have received considerable attention are *gathering* and *convergence*. The gathering task requires the robots to occupy a single point within a finite number of steps, starting from any initial configuration. The closely related convergence task requires the robots to converge to a single point, rather than reach it (namely, for every $\epsilon > 0$ there must be a time t_ϵ by which all robots are within distance of at most ϵ of each other).

A straightforward approach to these problems is based on requiring the robots to calculate some median position of the group and move towards that position. The current paper focuses on what is arguably the most natural variant of this approach, namely, using the *center of gravity* (also known as the *center of mass*, the *barycenter* or the average position) of the robot group.

The center of gravity approach is easy to analyze in the fully synchronous model. In the semi-synchronous or asynchronous models, however, analyzing the process becomes more involved, since the robots may take their measurements at different times, including while other robots are in movement. This might result in oscillatory effects on the centers of gravity calculated by the various robots, and possibly cause them to pass each other by in certain configurations. Subsequently, the correctness of the center of gravity algorithm has not been proven formally so far (to the best of our knowledge), i.e., it has not been shown that this algorithm guarantees convergence.

Several different and more sophisticated algorithms have been proposed before, some of which also guarantee gathering within finite time. The gathering problem was first discussed in [19, 20] in the semi-synchronous model. It was proven that it is impossible to achieve gathering of *two* oblivious autonomous mobile robots that have no common sense of orientation under the semi-synchronous model. (Convergence is easy to achieve in this setting.) Also, an algorithm was presented in [20] for gathering $N \geq 3$ robots in the semi-synchronous model. In the asynchronous model, an algorithm for gathering $N = 3, 4$ robots is brought in [22, 16], and an algorithm for gathering $N \geq 5$ robots has recently been described in [15]. The gathering problem was also studied in a system where the robots have limited visibility [23, 24].

Nevertheless, the center of gravity approach has several important advantages, making it desirable in many cases.

- It requires a very simple and efficient calculation, which can be carried out on simple hardware, and requires very low computational effort.
- It can be easily applied to 1, 2 or 3 dimensions and to any number of robots.
- It has a bounded and simple to calculate error due to rounding.
- It is oblivious, i.e., it requires no memory of previous actions and positions, rendering it both memory-efficient and self-stabilizing (in the sense that a finite number of transient errors cannot prevent eventual convergence).
- It prevents deadlocks, that is, every robot can move at any given position (unless it is already at the center of gravity).

In the current paper, we address the convergence of the center of gravity algorithm. In Section 2 we prove the correctness of the algorithm in the semi-

synchronous model of [19]. The problem appears to be more difficult in the asynchronous model. In Section 3 we make a modest first step by providing a convergence proof in this model for $N = 2$ robots.

Recently, we have been able to extend the result presented in the current paper and show that the center of gravity algorithm converges in the fully asynchronous model for any number of robots [25].

1.2 The model

The basic model studied in [14–20] can be summarized as follows (with some minor changes). The robots are expected to execute a given algorithm in order to achieve a prespecified mission. Each of the N robots R_i in the system is assumed to operate individually, repeatedly going through simple cycles consisting of four steps:

- **Look:** Identify the locations of all robots in R_i 's private coordinate system; the result of this step is a multiset of points $P = \{p_1, \dots, p_N\}$ defining the current *configuration*. The robots are indistinguishable, so each robot R_i knows its own location p_i , but does not know the identity of the robots at each of the other points.
- **Compute:** Execute the given algorithm, resulting in a goal point p_G .
- **Move:** Move on a straight line towards the point p_G . The robot might stop before reaching its goal point p_G , but is promised to traverse a distance of at least S (unless it has reached the goal).
- **Wait:** The robot sleeps for an indefinite time and then awakens for the next Move.

The “look” and “move” operations are identical in every cycle, and the differences between various algorithms are in the “compute” step. The procedure carried out in the “compute” step is identical for all robots. Notice that for the sake of analysis, the Compute and Wait steps can be absorbed into the Move step since no restriction is placed on the relative rate of movement during this step.

In most papers in this area (cf. [26, 19, 23, 16]), the robots are assumed to be rather limited. To begin with, the robots are assumed to have no means of directly communicating with each other. Moreover, the robots are also assumed to be *oblivious* (or memoryless), namely, they cannot remember their previous states, their previous actions or the previous positions of the other robots. Hence the algorithm employed by the robots for the “compute” step cannot rely on information from previous cycles, and its only input is the current configuration. While this is admittedly an over-restrictive and unrealistic assumption, developing algorithms for the oblivious model still makes sense in various settings, for two reasons. First, solutions that rely on non-obliviousness do not necessarily work in a dynamic environment where the robots are activated in different cycles, or robots might be added to or removed from the system dynamically. Secondly, any algorithm that works correctly for oblivious robots is inherently

self-stabilizing, i.e., it withstands transient errors that alter the robots' local states.

We consider two timing models. The first is the well studied *semi-synchronous* model (cf. [19]). This model is partially synchronous, in the sense that all robots operate according to the same fixed length clock cycles. However, not all robots are necessarily active in all cycles (the model is based on the assumption that each cycle is instantaneous). Rather, at each cycle t , any non-predetermined subgroup of the robots may commence the Look–Compute–Move cycle. The activation of the different robots can be thought of as managed by a hypothetical scheduler, whose only fairness obligation is that each robot must be activated and given a chance to operate infinitely often in any infinite execution.

The second model is the *fully asynchronous* model (cf. [15, 16]). In this model, robots operate on their own (time-varying) rates, and no assumptions are made regarding the relative speeds of different robots. In particular, robots may remain inactive for arbitrarily long periods between consecutive operation cycles.

To describe the center of gravity algorithm, hereafter named Algorithm `Go_to_COG`, we use the following notation. Denote by $\bar{r}_i[t]$ the location of robot i at time t . Denote the true center of gravity at time t by $\bar{c}[t] = \frac{1}{N} \sum_{i=1}^N \bar{r}_i[t]$. Denote by $\bar{c}_i[t]$ the center of gravity as last calculated by the robot i before time t , i.e., if the last calculation was done at time $t' < t$ then $\bar{c}_i[t] = \bar{c}[t']$. Note that, as mentioned before, robot i calculates this location in its own private coordinate system; however, for the purpose of describing the algorithm and its analysis, it is convenient to represent these locations in a unified global coordinate system (which of course is unknown to the robots themselves). By convention $\bar{c}_i[0] = \bar{r}_i[0]$ for all i .

Algorithm `Go_to_COG` is very simple. After measuring the current configuration at some time t , the robot i computes the average location of all robot positions (including its own), $\bar{c}_i[t] = \sum_i \bar{r}_i[t]/N$, and then proceeds to move towards the calculated point $\bar{c}_i[t]$. (As mentioned earlier, the move may terminate before the robot actually reaches the point $\bar{c}_i[t]$, but in case the robot has not reached $\bar{c}_i[t]$, it must have traversed a distance of at least S . Also, in the semi-synchronous model, the move operation terminates by the end of the cycle.)

2 Convergence in the semi-synchronous model

In this section we prove that in the semi-synchronous model, Algorithm `Go_to_COG` converges for any number of robots $N \geq 2$.

We start with a technical Lemma. The robots' *moment of inertia* at time t is defined as

$$I[t] = \frac{1}{N} \sum_{i=1}^N (\bar{r}_i[t] - \bar{c}[t])^2 .$$

Lemma 1. *Suppose that at some given time, all robots reside in a radius R circle centered at the origin, and let $0 < x_0 \leq R$. If the x coordinate of each robot satisfies $x \geq x_0$, then their moment of inertia satisfies $I \leq 4(R^2 - x_0^2)$.*

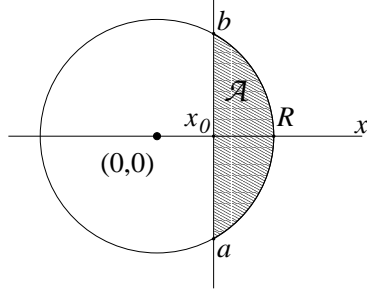


Fig. 1. Illustration of Lemma 1.

Proof: Since the region \mathcal{A} obtained from the intersection of the circle and the half-plane $x \geq x_0$ is convex, the center of gravity \bar{c} must also lie inside \mathcal{A} . Since \mathcal{A} contains less than half the circle, the longest chord in \mathcal{A} is the segment between the points $a = (x_0, -\sqrt{R^2 - x_0^2})$ and $b = (x_0, \sqrt{R^2 - x_0^2})$ (see Figure 1). The length of this segment is $L = 2\sqrt{R^2 - x_0^2}$ and therefore the distance between each robot and the center of gravity cannot exceed L . Thus,

$$I \equiv \frac{1}{N} \sum_{i=1}^N (\bar{r}_i - \bar{c})^2 \leq \frac{1}{N} \cdot N \cdot L^2 \leq L^2 . \quad \blacksquare$$

Defining the function $I_{\bar{x}}[t] \equiv \frac{1}{N} \sum_j (\bar{r}_j[t] - \bar{x})^2$, we notice the following (cf. [27]).

Lemma 2. $I_{\bar{x}} \geq I_{\bar{c}}$ for every \bar{x} .

Lemma 3. For the semi-synchronous model, in any execution of Algorithm *Go_to_COG*, $I[t]$ is a non-increasing function of time.

Proof: In each cycle t , each robot that is activated, and is not already at the current center of gravity, can get closer to the observed center of gravity $\bar{c}[t]$, hence $I_{\bar{c}[t]}[t] \geq I_{\bar{c}[t]}[t+1]$. By Fact 2, $I_{\bar{c}[t]}[t+1] \geq I_{\bar{c}[t+1]}[t+1]$. Combined, we have

$$I[t] = I_{\bar{c}[t]}[t] \geq I_{\bar{c}[t+1]}[t+1] = I[t+1] . \quad \blacksquare$$

Recall that S is the minimum movement distance of the robots.

Lemma 4. If at some time t_0 the robots' moment of inertia is $I_0 \equiv I[t_0]$, then there exists some $\hat{t} > t_0$, such that $I_1 \equiv I[\hat{t}] \leq \max(I - \frac{S^2}{N}, (1 - \frac{1}{100N^5}) I)$.

Proof: Assume without loss of generality that $\bar{c}[t_0]$ is at the origin. Take i to be the robot most distant from the origin at time t_0 , and assume again that it lies on the positive x axis, at point $(R, 0)$. Since it was the most distant robot, we have $R^2/N \leq I_0 \leq R^2$. Now take t_1 to be the time of this robot's first activation after t_0 . By Lemma 3, $I[t_1] \leq I_0$. We separate the analysis into two cases. First, suppose that the distance of robot i from $\bar{c}[t_1]$, the center of gravity

at time t_1 , is at least $\frac{R}{10N^2}$. Then, in its turn, it can move a distance at least $m = \min\{\frac{R}{10N^2}, S\}$. This decreases I by at least $\frac{m^2}{N}$ as required. Thus taking $\hat{t} = t_1 + 1$, the lemma follows.

If the robot cannot move a distance m in its turn, this implies that the distance of this robot from the center of gravity at time t is less than m . Since the circle is convex, no robot can reside at a distance greater than R from the origin. The difference between the x coordinate of robot i and $\bar{c}[t_1]$, the center of gravity at time t_1 , cannot exceed $m \leq \frac{R}{10N^2}$. Since the center of gravity is the average of the robots' locations, the x coordinate of each robot j must satisfy $x_j \geq R - Nm$. By Lemma 1, $I[t_1] \leq 4(R^2 - (R - Nm)^2) \leq 8RNm \leq \frac{8R^2}{10N} \leq (1 - \frac{1}{100N^2}) I[t_0]$. Thus taking $\hat{t} = t_1$, the lemma follows. ■

Theorem 1. *In the semi-synchronous model, for $N \geq 2$ robots in 2-dimensional space, Algorithm `Go_to_COG` converges.*

3 Asynchronous convergence of two robots

Let us now turn to the fully asynchronous model. We prove that in this model, Algorithm `Go_to_COG` converges for two robots. Our first observation applies to any $N \geq 2$.

Lemma 5. *If for some time t_0 , $\bar{r}_i[t_0]$ and $\bar{c}_i[t_0]$ for all i reside in a closed convex curve, \mathcal{P} , then for every time $t > t_0$, $\bar{r}_i[t]$ and $\bar{c}_i[t]$ also reside in \mathcal{P} for every $1 \leq i \leq N$.*

Proof: For the Move operation, it is clear that if for some i , $\bar{r}_i[t_0]$ and $\bar{c}_i[t_0]$ both reside in a convex hull then for the rest of the move operation $\bar{c}_i[t] = \bar{c}_i[t_0]$ does not change and $\bar{r}_i[t]$ is on the segment $[\bar{r}_i[t_0], \bar{c}_i[t_0]]$, which is inside \mathcal{P} .

For the Look step, if $N = 2$ then the calculated center of gravity is on the line segment connecting both robots, and therefore respects convexity. For $N > 2$ robots the center of gravity is on the line connecting the center of gravity of $N - 1$ robots and the N th robot, and the Lemma follows by induction. ■

For the following we assume the robots reside on the x -axis. For every t , let $H[t]$ denote the convex hull of the points $\bar{r}_1[t]$, $\bar{r}_2[t]$, $\bar{c}_1[t]$ and $\bar{c}_2[t]$, namely, the smallest closed interval containing all four points.

Corollary 1. *For $N = 2$ robots, for any times t, t_0 , if $t > t_0$ then $H[t] \subseteq H[t_0]$, namely, the points $\bar{r}_1[t]$, $\bar{r}_2[t]$, $\bar{c}_1[t]$ and $\bar{c}_2[t]$ reside in the line segment $H[t_0]$.*

Lemma 6. *If for some time t_0 , $\bar{r}_2[t_0] \geq \bar{r}_1[t_0]$, $\bar{c}_2[t_0] \geq \bar{r}_1[t_0]$ and $\bar{r}_1[t]$ is a monotonic non-increasing function for $t \in [t_0, \hat{t}]$, then*

1. *at all times $t \in [t_0, \hat{t}]$, $\bar{r}_2[t] \geq \bar{r}_1[t]$ and $\bar{c}_2[t] \geq \bar{r}_1[t]$, and*
2. *if at time $t^* \in [t_0, \hat{t}]$ robot 2 performed a Look step, then $\bar{c}_2[t] \in [\bar{r}_1[t], \bar{r}_2[t]]$ at all times $t \in [t^*, \hat{t}]$.*

Proof: Suppose that during the time interval $[t_0, \hat{t}]$, robot 2 performed $k \geq 0$ Look steps. Set $t_{k+1} = \hat{t}$, and if $k \geq 1$ then denote the times of these Look steps by $t_1, t_2, \dots, t_k \in [t_0, t_{k+1}]$. We now prove by induction that claims 1 and 2 hold also for all times $t \in [t_i, t_{i+1}]$ for $0 \leq i \leq k$.

For $i = 0$, observe that since robot 2 did not perform a Look step throughout the entire time interval $[t_0, t_1]$, its location $\bar{r}_2[t]$ at any time $t \in [t_0, t_1]$ is in the interval $[\bar{r}_2[t_0], \bar{c}_2[t_0]]$, which is entirely to the right of $\bar{r}_1[t_0] \geq \bar{r}_1[t]$, hence claim 1 of the lemma follows. Claim 2 of the lemma holds vacuously, and we are done.

We now assume that claim 1 holds for the interval $[t_{i-1}, t_i]$ and prove that both claims hold at any time in the interval $[t_i, t_{i+1}]$. Indeed, notice that by assumption robot 2 is to the right of robot 1 at time t_i , and since a Look step is performed at that time, $\bar{c}_2[t_i]$ is at the average position of the robots, hence it is also to the right of robot 1. Thus, $\bar{c}_2[t] = \bar{c}_2[t_i] \geq \bar{r}_1[t_i] \geq \bar{r}_1[t]$ for all $t \in [t_i, t_{i+1}]$ and claim 2 follows for that interval. During the time interval $[t_i, t_{i+1}]$, robot 2 traverses the segment $[\bar{c}_2[t_i], \bar{r}_2[t_i]]$, which is entirely to the right of $\bar{r}_1[t_i] \geq \bar{r}_1[t]$, hence claim 1 follows as well. ■

Lemma 7. For $N = 2$ robots, for every time t_0 there exists a time $\hat{t} \geq t_0$ in which $\bar{c}_i[\hat{t}] = \bar{r}_i[\hat{t}]$ and $\bar{c}_j[\hat{t}]$ resides on the line segment $[\bar{r}_1, \bar{r}_2]$ for $i \neq j$ and $i, j \in \{1, 2\}$.

Proof: Let $[a, b] = H[t_0]$. Take $t_1 > t_0$ to be the first time at which robot 1 completes its next Move operation. Thus, $\bar{r}_1[t_1] = \bar{c}_1[t_1]$. Without loss of generality assume $\bar{r}_2[t_1] > \bar{r}_1[t_1]$, i.e. $\bar{r}_2[t_1]$ falls in the segment $[\bar{r}_1[t_1], b]$. There are now three possible configurations (see Fig. 2).

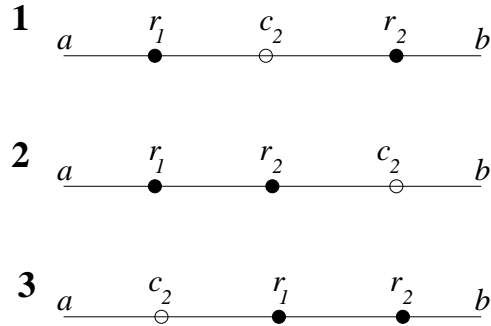


Fig. 2. The three cases in the proof of Lemma 7: ordering of the robots.

1. \bar{c}_2 resides on the segment $[\bar{r}_1, \bar{r}_2]$. Then take $\hat{t} = t_1$ and we are done.
2. \bar{c}_2 resides on the segment $[\bar{r}_2, b]$. Then let \hat{t} be the time when robot 2 completes its current Move operation. Since robot 2 moves away from robot 1,

\bar{c}_1 will always reside on the segment $[\bar{r}_1, \bar{r}_2]$ (by Lemma 6). At \hat{t} , $\bar{r}_2 = \bar{c}_2$ and c_1 resides on the segment $[\bar{r}_1, \bar{r}_2]$. Thus, we are done.

3. \bar{c}_2 resides on the line segment $[a, \bar{r}_1]$. In this case take t_2 to be the time at which robot 2 ends its Move operation. Whence, $\bar{c}_2[t_2] = \bar{r}_2[t_2]$. Since for any $t \in [t_1, t_2]$, $\bar{r}_2[t]$ is on the right of $\bar{c}_2[t]$ and $\bar{c}_1[t] = (\bar{r}_1[t_3] + \bar{r}_2[t_3])/2$ for some $t_3 \in [t_1, t]$, it follows that $\bar{c}_1[t]$ is to the right of $\bar{c}_2[t]$ and therefore also $\bar{r}_1[t]$ stays to the right of $\bar{c}_2[t] = \bar{c}_2[t_1]$. Hence, at time t_2 , \bar{r}_1 and \bar{c}_1 are on the left of $\bar{r}_2 = \bar{c}_2$. Therefore, we reach either case 1 or 2 with the roles of 1 and 2 reversed. This completes the proof. ■

Lemma 8. *For $N = 2$ robots, for every time t_0 there exists a time $\hat{t} > t_0$ such that $|H[\hat{t}]| \leq |H[t_0]|/2$.*

Proof: Let $L = |H[t_0]|$. Take $t_1 > t_0$ to be the time when both robots have completed at least one cycle after t_0 . By Corollary 1 they are still in $H[t_0]$. By Lemma 7 there exists $t_2 \geq t_1$ such that at time t_2 , $\bar{c}_1[t_2] = \bar{r}_1[t_2]$ and $\bar{c}_2[t_2] \in [\bar{r}_1, \bar{r}_2]$ (where robots 1 and 2 and the left/right directions are chosen to make this true). By Corollary 1, $[\bar{r}_1[t_2], \bar{r}_2[t_2]] \subseteq H[t_0]$. Take $D = d(\bar{r}_1[t_2], \bar{r}_2[t_2]) \leq L$.

Now take t_3 to be the time of the next Look step of robot 1. Clearly $L_1 \equiv d(\bar{c}_1[t_3], \bar{r}_1[t_3]) < L/2$ (since the center of gravity is the average of the robots' locations and $d(\bar{r}_1[t_3], \bar{r}_2[t_3]) \leq |H(t_3)| \leq |H(t_0)| = L$). Now take \hat{t} to be the time of the end of robot 1's Move operation, so $\bar{c}_1[\hat{t}] = \bar{r}_1[\hat{t}]$. Without loss of generality assume at t_3 robot 1 is the rightmost. Either of the following three cases is possible at \hat{t} (see Fig. 3).

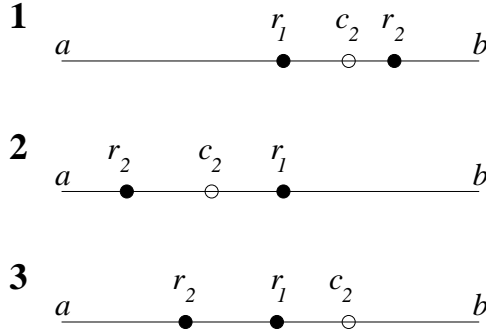


Fig. 3. The three cases in the ordering of the robots in Lemma 8.

1. $\bar{r}_2[\hat{t}]$ is to the right of $\bar{r}_1[\hat{t}]$. This means that for some time $t_4 \in [t_3, \hat{t}]$, $\bar{r}_1[t_4] = \bar{r}_2[t_4]$. At any $t \in [t_2, t_4]$, $\bar{r}_1[t]$ was to the right of $\bar{r}_2[t]$, implying that $\bar{c}_2[t]$ was also on the right of $\bar{r}_2[t]$, and specifically $\bar{c}_2[\hat{t}] \geq \bar{r}_2[\hat{t}] \geq \bar{r}_1[\hat{t}] = \bar{c}_1[\hat{t}]$. Therefore they are all on the segment $[\bar{r}_1[\hat{t}], \bar{r}_1[t_3]]$ of length $L_1 \leq L/2$.

2. $\bar{r}_2[\hat{t}]$ and $\bar{c}_2[\hat{t}]$ are to the left of $\bar{r}_1[\hat{t}]$. Since robot 2 must have been moving right at all $t_2 \leq t \leq \hat{t}$ it must hold that $\bar{c}_2[\hat{t}]$ resides on the segment $[\bar{r}_2[\hat{t}], \bar{r}_1[\hat{t}]$ and that $d(\bar{r}_1[\hat{t}], \bar{r}_2[\hat{t}]) \leq d(\bar{r}_1[t_2], \bar{r}_2[t_2]) = D$. Now, since $d(\bar{r}_1[t_2], \bar{c}_1[t_2]) = d(\bar{r}_1[t_2], \bar{r}_2[t_2])/2 = D/2$, and $\bar{r}_2[\hat{t}]$ and $\bar{c}_2[\hat{t}]$ are to the left of $\bar{r}_1[\hat{t}]$ they are restricted to the segment left of $\bar{r}_1[\hat{t}]$ and to the right of $\bar{r}_2[t_2]$ of length $D/2 \leq L/2$.
3. $\bar{r}_1[\hat{t}] \in [\bar{r}_2[\hat{t}], \bar{c}_2[\hat{t}]$. Since $d(\bar{r}_2[\hat{t}], \bar{c}_2[\hat{t}]) \leq L/2$ we are done. ■

Theorem 2. *In the fully asynchronous model, for two robots in d dimensional space, Algorithm *Go-to-COG* converges.*

Proof: Apply Lemma 8 to each dimension separately. ■

References

1. Parker, L., Touzet, C.: Multi-robot learning in a cooperative observation task. In: Distributed Autonomous Robotic Systems 4. (2000) 391–401
2. Jung, D., Cheng, G., Zelinsky, A.: Experiments in realising cooperation between autonomous mobile robots. In: Proc. Int. Symp. on Experimental Robotics. (1997)
3. Parker, L., Touzet, C., Fernandez, F.: Techniques for learning in multi-robot teams. In Balch, T., Parker, L., eds.: Robot Teams: From Diversity to Polymorphism. A. K. Peters (2001)
4. Kawachi, Y., Inaba, M., Fukuda, T.: A principle of decision making of cellular robotic system (CEBOT). In: Proc. IEEE Conf. on Robotics and Automation. (1993) 833–838
5. Beni, G., Hackwood, S.: Coherent swarm motion under distributed control. In: Proc. DARS'92. (1992) 39–52
6. Murata, S., Kurokawa, H., Kokaji, S.: Self-assembling machine. In: Proc. IEEE Conf. on Robotics and Automation. (1994) 441–448
7. Mataric, M.: Interaction and Intelligent Behavior. PhD thesis, MIT (1994)
8. Parker, L.: Designing control laws for cooperative agent teams. In: Proc. IEEE Conf. on Robotics and Automation. (1993) 582–587
9. Parker, L.: On the design of behavior-based multi-robot teams. J. of Advanced Robotics **10** (1996)
10. Balch, T., Arkin, R.: Behavior-based formation control for multi-robot teams. IEEE Trans. on Robotics and Automation **14** (1998)
11. Wagner, I., Bruckstein, A.: From ants to a(ge)nts. Annals of Mathematics and Artificial Intelligence **31**, **special issue on ant-robotics** (1996) 1–5
12. Cao, Y., Fukunaga, A., Kahng, A., Meng, F.: Cooperative mobile robots: Antecedents and directions. In: Proc. Int. Conf. of Intel. Robots and Sys. (1995) 226–234
13. Cao, Y., Fukunaga, A., Kahng, A.: Cooperative mobile robotics: Antecedents and directions. Autonomous Robots **4** (1997) 7–23
14. Ando, H., Suzuki, I., Yamashita, M.: Formation and agreement problems for synchronous mobile robots with limited visibility. In: Proc. IEEE Symp. of Intelligent Control. (1995) 453–460
15. Cieliebak, M., Flocchini, P., Prencipe, G., Santoro, N.: Solving the robots gathering problem. In: Proc. 30th Int. Colloq. on Automata, Languages and Programming. (2003) 1181–1196

16. Cieliebak, M., Prencipe, G.: Gathering autonomous mobile robots. In: Proc. 9th Int. Colloq. on Structural Information and Communication Complexity. (2002) 57–72
17. Prencipe, G.: CORDA: Distributed coordination of a set of autonomous mobile robots. In: Proc. 4th European Research Seminar on Advances in Distributed Systems. (2001) 185–190
18. Sugihara, K., Suzuki, I.: Distributed algorithms for formation of geometric patterns with many mobile robots. *Journal of Robotic Systems* **13** (1996) 127–139
19. Suzuki, I., Yamashita, M.: Distributed anonymous mobile robots - formation and agreement problems. In: Proc. 3rd Colloq. on Structural Information and Communication Complexity. (1996) 313–330
20. Suzuki, I., Yamashita, M.: Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM J. on Computing* **28** (1999) 1347–1363
21. Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Hard tasks for weak robots: The role of common knowledge in pattern formation by autonomous mobile robots. In: Proc. 10th Int. Symp. on Algorithms and Computation. (1999) 93–102
22. Prencipe, G.: Distributed Coordination of a Set of Autonomous Mobile Robots. PhD thesis, Università Degli Studi Di Pisa (2002)
23. Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Gathering of autonomous mobile robots with limited visibility. In: Proc. 18th Symp. on Theoretical Aspects of Computer Science. (2001) 247–258
24. Ando, H., Oasa, Y., Suzuki, I., Yamashita, M.: A distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Trans. Robotics and Automation* **15** (1999) 818–828
25. Cohen, R., Peleg, D.: Robot convergence via center-of-gravity algorithms. Technical Report MSC 04-2, Weizmann Institute of Science, Rehovot, Israel (2004)
26. Suzuki, I., Yamashita, M.: Agreement on a common x-y coordinate system by a group of mobile robots. In: Proc. Dagstuhl Seminar on Modeling and Planning for Sensor-Based Intelligent Robots. (1996)
27. Goldstein, H.: *Classical Mechanics*. 2nd edn. Addison-Wesley, Reading, MA, USA (1980) 672 pages.