# Convergence of Autonomous Mobile Robots with Inaccurate Sensors and Movements
## (Extended Abstract)

Reuven Cohen* and David Peleg**

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

**Abstract.** The common theoretical model adopted in recent studies on algorithms for systems of autonomous mobile robots assumes that the positional input of the robots is obtained by perfectly accurate visual sensors, that robot movements are accurate, and that internal calculations performed by the robots on (real) coordinates are perfectly accurate as well. The current paper concentrates on the effect of weakening this rather strong set of assumptions, and replacing it with the more realistic assumption that the robot sensors, movement and internal calculations may have slight inaccuracies. Specifically, the paper concentrates on the ability of robot systems with inaccurate sensors, movements and calculations to carry out the task of convergence. The paper presents several impossibility results, limiting the inaccuracy allowing convergence. The main positive result is an algorithm for convergence under bounded measurement, movement and calculation errors.

## 1 Introduction

**Background.** Distributed systems consisting of autonomous mobile robots (a.k.a. *robot swarms*) are motivated by the idea that instead of using a single, highly sophisticated and expensive robot, it may be advantageous in certain situations to employ a group of small, simple and relatively cheap robots. This approach is of interest for a number of reasons. Multiple robot systems may be used to accomplish tasks that *cannot* be achieved by a single robot. Such systems usually have decreased cost due to the simpler individual robot structure. These systems can be used in a variety of environments where the acting (human or artificial) agents may be at risk, such as military operations, exploratory space missions, cleanups of toxic spills, fire fighting, search and rescue missions, and other hazardous tasks. In such situations, a multiple robot system has a better chance of successfully carrying out its mission (while possibly accepting the loss or destruction of some of its robots) than a single irreplaceable robot. Such systems may also be useful for carrying out simple repetitive tasks that humans may find extremely boring or tiresome.

* Supported by the Pacific Theatres Foundation.
** Supported in part by a grant from the Israel Science Foundation.

B. Durand and W. Thomas (Eds.): STACS 2006, LNCS 3884, pp. 549–560, 2006.

Subsequently, studies of autonomous mobile robot systems can be found in different disciplines, from engineering to artificial intelligence. (A survey on the area is presented in [4].)

A number of recent studies on autonomous mobile robot systems focus on algorithms for distributed control and coordination from a distributed computing point of view (cf. [10, 13, 12, 2]). The approach is to propose suitable computational models and analyze the minimal capabilities the robots must possess in order to achieve their common goals. The basic model studied in the these papers can be summarized as follows. The robots execute a given algorithm in order to achieve a prespecified task. Each robot in the system is assumed to operate individually in simple cycles consisting of three steps:

(1) "Look": determine the current configuration by identifying the locations of all visible robots and marking them on your private coordinate system,
(2) "Compute": execute the given algorithm, resulting in a goal point $p_G$, and
(3) "Move": travel towards the point $p_G$. The robot might stop before reaching its goal point $p_G$, but is guaranteed to traverse at least some minimal distance unit (unless reaching the goal first).

**Weak and Strong Model Assumptions.** Due to the focus on cheap robot design and the minimal capabilities allowing the robots to perform some tasks, most papers in this area (cf. [10, 13, 9, 5]) assume the robots to be rather limited. Specifically, the robots are assumed to be indistinguishable, so when looking at the current configuration, a robot cannot tell the identity of the robots at each of the points (apart from itself). Furthermore, the robots are assumed to have no means of direct communication. This gives rise to challenging "distributed coordination" problems since the only permissible communication is based on "positional" or "geometric" information exchange, yielding an interesting variant of the classical (direct-communication based) distributed model.

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.

On the other hand, the robot model studied in the literature includes the following *overly strong* assumptions:

- when a robot observes its surroundings, it obtains a perfect map of the locations of the other robots relative to itself,
- when a robot performs internal calculations on (real) coordinates, the outcome is exact (infinite precision) and suffers no numerical errors, and

– when a robot decides to move to a point $p$, it progresses on the straight line connecting its current location to $p$, stopping either precisely at $p$ or at some earlier point on the straight line segment leading to it.

All of these assumptions are unrealistic. In practice, the robot measurements suffer from nonnegligible inaccuracies in both distance and angle estimations. (The most common range sensors in mobile robots are sonar sensors. The accuracy in range estimation of the common models is about $\pm 1\%$ and the angular separation is about $3°$; see, e.g., [11]. Other possible range detectors are based on laser range detection, which is usually more accurate than the sonar, and on stereoscopic vision, which is usually less accurate.) The same applies to the precision of robot movements. Due to various mechanical factors such as unstable power supply, friction and force control, the exact distance a robot traverses in a single cycle is hard to control, or even predict with high accuracy. This makes most previous algorithms proposed in the literature inapplicable in most practical settings. Finally, the robots' internal calculations cannot be assumed precise, for a variety of well-understood reasons such as convergence rates of numerical procedures, truncated numeric representations, rounding errors and more.

In this paper we address the issue of imperfections in robot measurements, calculations and movements. Specifically, we replace the unrealistic assumptions described above with more appropriate ones, allowing for measurement, calculation and movement inaccuracies, and show that efficient algorithmic solutions can still be obtained in the resulting model.

We focus on the *gathering* and *convergence* problems, which have been extensively studied in the common (fully accurate) model (cf. [13, 10, 5]). The *gathering* problem is defined as follows. Starting from any initial configuration, the robots should occupy a single point within a finite number of steps. The closely related *convergence* problem requires the robots to converge to a single point, rather than reach it (namely, for every $\epsilon > 0$ there must be a time $t_\epsilon$ by which all robots are within distance of at most $\epsilon$ of each other).

It is important to note that analyzing the effect of errors is not merely of theoretical value. In Section 3 we show that gathering cannot be guaranteed in environments with errors, and illustrate how certain existing geometric algorithms, including ones designed for fault tolerance, fail to guarantee even convergence in the presence of small errors. We also show (in Theorem 9) that the standard center of gravity algorithm may also fail to converge when errors occur.

**Related Work.** A number of problems concerning coordination in autonomous mobile robot systems have been considered so far in the literature. The gathering problem was first discussed in [13] in the semi-synchronous model. It was proven that it is impossible to gather *two* oblivious autonomous mobile robots that have no common sense of orientation under the semi-synchronous model. Also, an algorithm was presented in [13] for gathering $N \geq 3$ robots in the semi-synchronous model. In the asynchronous model, a gathering algorithm has recently been described in [5]. Fault tolerant gathering algorithms (in the crash and Byzantine fault models) were studied in [1]. The gathering problem was also studied in a system where the robots have limited visibility. The visibility conditions are modeled by means of a *visibility graph*, representing the (symmetric) visibility relation of the robots with respect to one another, i.e., an edge exists between robots $i$ and $j$ if and only if $i$ and $j$ are visible to each other. (Note that in this model visibility is a boolean predicate and does not involve imprecisions, namely, if robot $j$ is visible to robot $i$ then its precise coordinates are measured accurately.) It was shown that the problem is unsolvable in case the visibility graph is not connected [9]. In [2] a convergence algorithm was provided for any $N$, in limited visibility systems. The natural gravitational algorithm based on going to the center of gravity, and its convergence properties, were studied in [6].

Other problems studied, e.g., in [12, 13, 7, 8, 10, 3], concern formation of various geometric patterns, flocking (or "following the leader"), distributed search after (static or moving) targets, achieving even distribution, partitioning and wake-up via the freeze-tag paradigm.

**Our Results.** In this paper we study the convergence problem in the common semi-synchronous model where the robots' only inputs are obtained by inaccurate visual sensors, and their movements and internal calculations may be inaccurate as well. In Section 3 we present several impossibility theorems, limiting the inaccuracy allowing convergence, and prohibiting a general algorithm for gathering in a finite number of steps. In Section 4 we present an algorithm for convergence under bounded error, and prove its correctness, first in the fully synchronous model, and then in the semi-synchronous model. Finally, we compare the proposed algorithm with the ordinary center of gravity algorithm.

## 2 The Model

Each of the $N$ robots $i$ in the system is assumed to operate individually in simple cycles. Every cycle consists of three steps, "look", "compute" and "move". The result of the "look" step taken by $i$ is a multiset of points $P = \{p_1, \ldots, p_N\}$ (with $p_i = 0$ in $i$'s local coordinate system) defining the current *configuration* and used by the robot in calculating its next goal point $p_G$. Note that the "look" and "move" steps are carried out identically in every cycle, independently of the algorithm used. The differences between different algorithms occur in the "compute" step. Moreover, the procedure carried out in the "compute" step is identical for all robots. If the robots are oblivious, then the algorithm cannot rely on information from previous cycles, thus the procedure can be fully specified by describing a single "compute" step, and its only input is the current configuration $P$, giving the locations of the robots.

As mentioned earlier, our computational model for studying and analyzing problems of coordinating and controlling a set of autonomous mobile robots follows the well studied *semi-synchronous* ($\mathcal{SSYNC}$) model. This model is partially synchronous, in the sense that all robots operate according to the same clock cycles, but not all robots are necessarily active in all cycles. Those robots which are awake at a given cycle make take a measurement of the positions of all other robots. Then they may make a computation and move instantaneously accordingly. 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. On the way to establishing the result on the $\mathcal{SSYNC}$ model, we prove it first in the fully synchronous ($\mathcal{FSYNC}$) model. Finally, we also discuss its performance in the fully asynchronous ($\mathcal{ASYNC}$) model.

Our model assumes that the robot's location estimation is imprecise, with imprecision bounded by some accuracy parameter $\epsilon$ known at the robot's design. In general, this imprecision can affect both distance and angle estimations. In particular, distance imprecision means that if the true location of an observed point in $i$'s coordinate system is $\overline{V}$ and the measurement taken by $i$ is $\bar{v}$, then this measurement will satisfy $(1-\epsilon)V < v < (1+\epsilon)V$. (Throughout, for a vector $\bar{v}$, we denote by $v$ its scalar length, $v = |\bar{v}|$. Also, capital letters are used for exact quantities, whereas lowercase ones denote the robots' views).

The accuracy in angle measurements is $\theta_0$ (where it can always be assumed that $\theta_0 \leq \pi$). I.e., the angle $\theta$ between the actual distance vector $\overline{V}$ and the measured distance vector $\bar{v}$ satisfies $\theta \leq \theta_0$, or alternatively, $\cos\theta = \frac{\overline{V}\bar{v}}{Vv} \geq \cos\theta_0$. In what follows, we consider the model $\mathcal{ERR}$ in which both types of imprecision are possible, and the model $\mathcal{ERR}^-$ where only distance estimates are inaccurate. This gives rise to six composite timing/error models, denoted $\langle \mathcal{T}, \mathcal{E} \rangle$, where $\mathcal{T}$ is the timing model under consideration ($\mathcal{FSYNC}$, $\mathcal{SSYNC}$ or $\mathcal{ASYNC}$) and $\mathcal{E}$ is the error model ($\mathcal{ERR}$ or $\mathcal{ERR}^-$).

While in reality each robot uses its own private coordinate system, for simplicity of presentation it is convenient to assume the existence of a global coordinate system (which is unknown to the robots) and use it for our notation. Throughout, we denote by $\bar{R}_j$ the location of robot $j$ in the global coordinate system. In addition, for every two robots $i$ and $j$, denote by $\overline{V}^i_j = \bar{R}_j - \bar{R}_i$ the true location of robot $j$ from the position of robot $i$ (i.e., the true vector from $i$ to $j$), and by $\bar{v}^i_j$ the location of robot $j$ as measured by $i$, *translated* to the global coordinate system. Likewise, our algorithm and its analysis will be described in the global coordinate system, although each of the robots will apply it in its own local coordinate system. As the functions computed by the algorithm are all invariant under translations, this representation does not violate the correctness of our analysis.

If the robots may have inaccuracies in distance estimation but not in directions, then $i$ will measure $\overline{V}^i_j$ as $\bar{v}^i_j = (1+\epsilon^i_j)\overline{V}^i_j$, where $-\epsilon < \epsilon^i_j < \epsilon$ is the local error factor in distance estimation at robot $i$. For robots with inaccuracy in angle measurement as well, if the true distance is $V^i_j$, then $i$ will measure it as $v^i_j = (1+\epsilon^i_j)V^i_j$, where $-\epsilon < \epsilon^i_j < \epsilon$ and the angle $\theta$ between $\overline{V}^i_j$ and $\bar{v}^i_j$ will satisfy $|\theta| \leq \theta_0$. Values computed at time-slot $t$ are denoted by a parameter $[t]$. Also, the actual error factor is time dependent and its value at time $t$ is denoted by $\epsilon^i_j[t]$. The parameter $t$ is omitted whenever clear from the context.

Inaccuracies in movement and calculations should also be taken into account. For movement, we may assume that if a robot wants to move from its current location $\bar{R}_i$ to some goal point $p_G$, then it will move on a vector at an angle of at most $\phi_0$ from the vector $\overline{r_i p_G}$ and to any distance $d$ in the interval $d \in [1-\epsilon, 1+\epsilon] \cdot |\overline{r_i p_G}|$. Also, when it calculates a goal point $p_G = (x, y)$, it will have a multiplicative error of up to $\epsilon$. In the center of gravity algorithms presented below, the calculation error is bounded linearly in the calculated terms. Hence it can be seen that relative movement and calculation errors can be replaced with errors in measurement causing the same effect, so these errors can be treated using the same algorithm by recalibrating $\epsilon$. (Note that *absolute* errors in movement or calculation can not be treated, since even when the robots have already almost converged, such errors may cause them to spread again.) Therefore, throughout most of the ensuing technical development, we will assume only measurement inaccuracies.

We use the following technical lemma. (Most proofs are deferred to full paper.)

**Lemma 1.** *For two vectors $\bar{a}$ and $\bar{b}$ with $a \le 1 \le b$, let $x = |\bar{a} - \bar{b}|$ and $y = |\bar{a} - \bar{b}/b|$. Then (1) $x^2 - y^2 \ge (b-1)^2 + 2(1-a)(b-1) \ge (b-1)^2$, and (2) $y \le x$.*

## 3 The Effect of Measurement Errors

To appreciate the importance of error analysis one must realize two facts. First, computers are limited in their computational power, and therefore cannot perform perfect precision calculations. This may seem insignificant, since floating point arithmetic can be made to very high accuracy with modern computers. However, this may prove to be a practical problem. For instance, the point that minimizes the sum of distances to the robots' locations (also known as the Weber point) may be used to achieve gathering. However, this point is not computable, due to its infinite sensitivity to location errors. Second, the correctness of algorithms that use geometric properties of the plane is usually proven using theorems from Euclidean geometry. However, these theorems are, in many cases, inappropriate when measurement or calculation errors occur.

**Impossibility Results.** We start with some impossibility results. The proofs of these results are based on the ability of the adversary to partition the space of possible initial configurations into countably many regions, each of uncountably many configurations (say, on the basis of the initial distance between the robots), such that within each region, the outcome of the algorithm (i.e., the movement instructions to the robots) is the same. The following theorem holds even in a rather strong setting where the timing model is fully synchronous, and the robots have unlimited memory and are allowed to use randomness.

**Theorem 1.** *Even in the strong setting outlined above, gathering is impossible*
*(1) for two robots on the line with inexact distance measurements,*
*(2) for any number of robots assuming inaccuracies in both the distance* and *angle measurements.*

It seems reasonable to conjecture that even convergence is impossible for robots with large measurement errors. The exact limits are not completely clear. The following theorem gives some rather weak limits on the possibility of convergence.

In the theorem we assume that the robot has no sense of direction in a strong way, *i.e.*, at every cycle the adversary can choose each robot's axes independent of previous cycles.

**Theorem 2.** *For a configuration of $N = 3$ robots having an error parameter $\theta_0 \geq \pi/3$ in angle measurement, there is no deterministic algorithm for convergence even assuming* exact *distance estimation, fully synchronous model and unlimited memory.*

**Problems with Existing Algorithms.** To illustrate the second point raised in the beginning of this section, consider the algorithm $\texttt{3} - \texttt{Gather}$ presented in [1]. This algorithm achieves gathering of three robots using several simple rules. One of these rules states that if the robots form an obtuse triangle, then they move towards the vertex with the obtuse angle. As shown above, no algorithm can guarantee gathering when measurement errors occur. Furthermore, although this algorithm is designed to robustness and achieves gathering even if one of the robots fails, one can verify that it might fail to achieve even convergence in the presence of angle measurement errors of at least 15°.

Likewise, for a group of $N > 3$ robots the algorithm $\texttt{N} - \texttt{Gather}$ is presented in [1]. In this algorithm the smallest enclosing circle of the robot group is calculated, and in case there is a single robot inside this circle, it does not move. In the presence of measurement inaccuracies, this rule can potentially cause deadlock, implying that the algorithm might fail to achieve even convergence in the presence of angle and distance measurement errors of $\epsilon > 0$.

## 4 The Convergence Algorithm

**Algorithm `Go_to_COG`.** A natural algorithm for autonomous robot convergence is the gravitational algorithm, where each robot computes the average position (center of gravity) of the group, $\bar{v}^i_{cog} = \frac{1}{N}\sum_j \bar{v}^i_j$, and moves towards it.

The properties of Algorithm `Go_to_COG` in a model with fully accurate measurements have been studied in [6]. In particular, it is proven that a group of $N$ robots executing Algorithm `Go_to_COG` will converge in the $\mathcal{ASYNC}$ model with no measurement errors. If measurements are not guaranteed to be accurate, Algorithm `Go_to_COG` may not guarantee convergence. Nevertheless, convergence *is* guaranteed in the fully synchronous model, i.e., we have the following.

**Lemma 2.** *In the $\langle \mathcal{FSYNC}, \mathcal{ERR}^- \rangle$ model with $\epsilon < \frac{1}{2}$, a group of $N$ robots performing Algorithm* `Go_to_COG` *converges.*

The convergence of Algorithm `Go_to_COG` in the $\mathcal{SSYNC}$ model is not clear at the moment. However, as shown below, in the $\mathcal{ASYNC}$ model there are scenarios where robots executing Algorithm `Go_to_COG` fail to converge. This leads us to propose the following slightly more involved algorithm.

**Algorithm `RCG`.** Our algorithm, named `RCG`, is based on calculating the center of gravity (CoG) of the group of robots, and also estimating the maximum possible error in the CoG calculation. The robot makes no movement if it is within the maximum possible error from the CoG. If it is outside the circle of error, it moves towards the CoG, but only up to the bounds of the circle of error. We fix a conservative error estimate parameter, $\epsilon_0 > \epsilon$.

Following is a more detailed explanation of the algorithm. In step 1, the measured center of gravity is estimated using the conducted measurements. In step 2 the distance to the furthest robot is found. Notice that this distance may not be accurate, and that this needs not be even the real furthest robot. The result of step 2 is used in step 3 to give an estimate of the possible error in the CoG calculation. In step 4 the robots decide to hold if it is within the circle of error, or calculates its destination point, which is on the boundary of the error circle centered at the calculated CoG. A formal description of the algorithm is given next. Note that Algorithm `Go_to_COG` is identical to Algorithm `RCG` with parameter $\epsilon_0 = 0$.

**Code for robot** $i$

1. Estimate the measured center of gravity, $\bar{v}^i_{cog} = \frac{1}{N}\sum_j \bar{v}^i_j$
2. Let $d^i_{max} = \max_j\{v^i_j\}$ /* max distance measured to another robot
3. Let $\rho^i = \dfrac{\epsilon_0}{1-\epsilon_0} \cdot d^i_{max}$ /* estimate for max error in calculated CoG
4. If $v^i_{cog} > \rho^i$ then move to the point $\bar{c}_i = (1 - \rho^i / v^i_{cog}) \cdot \bar{v}^i_{cog}$.
   Otherwise do not move.

**Analysis of `RCG` in the Semi-synchronous Model.** We first prove the convergence of Algorithm `RCG` in the $\langle\mathcal{FSYNC}, \mathcal{ERR}^-\rangle$ model. Denote the true center of gravity of the robots in the global coordinate system by $\bar{R}_{cog} = \frac{1}{N}\sum_j \bar{R}_j$, and the vector from robot $i$ to the center of gravity by $\overline{V}^i_{cog} = \bar{R}_{cog} - \bar{R}_i = \frac{1}{N}\sum_j \overline{V}^i_j$, where $\overline{V}^i_j = \bar{R}_j - \bar{R}_i$. Denote the distance from the true center of gravity of the robots to the robot farthest from it by $D_{cog} = \max_i\{V^i_{cog}\}$. Also, denote the true distance from $i$ to the robot farthest from it by $D^i_{max} = \max_j\{V^i_j\}$. We use the following two properties.

**Fact 3.** *For every i:* *(a)* $D^i_{max} \le 2D_{cog}$,
*(b)* $(1-\epsilon_0)D^i_{max} < (1-\epsilon)D^i_{max} \le d^i_{max} \le (1+\epsilon)D^i_{max} < (1+\epsilon_0)D^i_{max}$.

For the synchronous model, we define the $t$th round to begin at time $t$ and end at time $t+1$. The robots all perform their Look phase simultaneously. The robots' *moment of inertia* at time $t$ is defined as

$$I[t] = \frac{1}{N}\sum_j \left(\overline{V}^j_{cog}[t]\right)^2 = \frac{1}{N}\sum_j \left(\bar{R}_j[t] - \bar{R}_{cog}[t]\right)^2 .$$

Defining $I_{\bar{x}}[t] \equiv \frac{1}{N}\sum_j (\bar{R}_j[t] - \bar{x})^2$, we use the following fact.

**Fact 4.** $I_{\bar{x}}[t]$ *attains its minimum on* $\bar{x} = \bar{R}_{cog}[t]$.

For ease of presentation, we assume a slightly simpler model where the move step of a robot is ensured to bring it to its goal point $p_G$. A slightly more involved analysis, deferred to the full paper, applies to the usual setting where it is assumed that the robot might stop before reaching $p_G$, but is guaranteed to traverse at least some minimal distance unit (unless reaching the goal first).

Our main lemma is the following.

**Lemma 3.** *For fixed $\epsilon_0 < 0.2$, in the $\langle \mathcal{FSYNC}, \mathcal{ERR}^- \rangle$ model, Algorithm* RCG *guarantees that at every round $t$:*

1. *at least one robot can move,*
2. *every robot $i$ decreases its distance from the true center of gravity at time $t$, i.e., $|\bar{R}_i[t+1] - \bar{R}_{cog}[t]| < |\bar{R}_i[t] - \bar{R}_{cog}[t]|$,*
3. *the robots' moment of inertia decreases, i.e., $I[t+1] < I[t]$.*

*Proof.* Consider some time $t$. Denote by $\overline{err}^i = \frac{1}{N}\sum_j \epsilon^i_j \overline{V}^i_j$ the error component in the center of gravity calculation by robot $i$. Then the center of gravity computed by robot $i$ can be expressed as

$$\bar{v}^i_{cog} \;=\; \frac{1}{N}\sum_j \bar{r}^i_j \;=\; \frac{1}{N}\sum_j (\bar{R}_j + \epsilon^i_j \overline{V}^i_j) \;=\; \overline{V}^i_{cog} + \overline{err}^i \;.$$

By the bounded error assumption and Fact 3(a),

$$err^i \;=\; \frac{1}{N}\sum_j \epsilon^i_j \cdot V^i_j \;\le\; \epsilon D^i_{max} \;\le\; 2\epsilon D_{cog} \;<\; 2\epsilon_0 D_{cog}. \qquad (1)$$

By the two parts of Fact 3, the calculated value $\rho^i$ is bounded by

$$\rho^i \;\le\; \frac{\epsilon_0(1+\epsilon_0)}{1-\epsilon_0} \cdot D^i_{max} \;\le\; \frac{\epsilon_0(1+\epsilon_0)}{1-\epsilon_0} \cdot 2D_{cog}. \qquad (2)$$

Combining (1) and (2), we have for each $i$,

$$err^i + \rho^i \;\le\; f(\epsilon_0) \cdot D_{cog} \;<\; D_{cog} \;, \qquad (3)$$

where $f(\epsilon_0) = 4\epsilon_0/(1-\epsilon_0)$, and the last inequality follows from the assumption that $\epsilon_0 < 0.2$.

For $k = \arg\max_j\{V^j_{cog}\}$, the robot farthest from the center of gravity, we have $V^k_{cog} = D_{cog}$ and $\bar{v}^k_{cog} = \overline{V}^k_{cog} + \overline{err}^k$, hence by (3) and the triangle inequality, $\rho^k \;<\; V^k_{cog} - err^k \;\le\; v^k_{cog}$ . This implies that at round $t$, robot $k$ is allowed to move in Step 4 of the algorithm, proving Part 1 of the Lemma.

To prove Part 2, consider a round $t$ and a robot $i$ which moved in round $t$. Fix $\bar{x} = \bar{R}_{cog}[t]$ and take $\bar{a} = \overline{err}^i[t]/\rho^i[t]$ and $\bar{b} = \bar{v}^i_{cog}[t]/\rho^i[t]$. Note that by (1) and Fact 3(b), $err^i[t] \;\le\; \frac{\epsilon}{1-\epsilon} \cdot d^i_{max}[t] \;<\; \frac{\epsilon_0}{1-\epsilon_0} \cdot d^i_{max}[t] \;=\; \rho^i[t]$, hence $a \le 1$. Also, at round $t+1$, robot $i$ moves if and only if $v^i_{cog}[t] > \rho^i[t]$, hence if $i$ moved then $b = v^i_{cog}[t]/\rho^i[t] > 1$. Hence Lemma 1(1) can be applied. Noting that $\bar{b}/b = (\bar{R}_i[t+1] - \bar{R}_{cog}[t])/\rho^i[t]$, we get $|\bar{R}_i[t+1] - \bar{x}| < |\bar{R}_i[t] - \bar{x}|$ , yielding Part 2 of the Lemma. It remains to prove Part 3. Note that for a robot that did not move, $|\bar{R}_i[t+1] - \bar{x}| = |\bar{R}_i[t] - \bar{x}|$. Using this fact that and Part 2, we have that $I_{\bar{x}}[t+1] < I_{\bar{x}}[t] = I[t]$ . Finally, by Fact 4, $I[t+1] \leq I_{\bar{x}}[t+1]$, yielding Part 3 of the Lemma. ∎

**Theorem 5.** *In every execution of Algorithm* RCG *in the* $\langle \mathcal{FSYNC}, \mathcal{ERR}^- \rangle$ *model, the robots converge.*

*Proof.* By Part 1 of Lemma 3, the robot $k$ most distant from the center of gravity can always move if Algorithm `Go_to_COG` is applied. By Part 2 of Lemma 3, in round $t$ every robot decreases its distance from the old center of gravity, $\bar{x} = \bar{R}_{cog}[t]$. Therefore, to bound from below the decrease in $I$, we are only required to examine the behavior of the most distant robot. By Lemma 1(1) with $a = \bar{R}_k[t+1] - \bar{R}_{cog}[t]$ and $b = \bar{R}_k[t] - \bar{R}_{cog}[t]$, for $\epsilon_0 < 0.2$ we have $v^k_{cog} > \rho^k$ and $(\bar{R}_k[t] - \bar{R}_{cog}[t])^2 - (\bar{R}_k[t+1] - \bar{R}_{cog}[t])^2 \geq (v^k_{cog} - \rho^k)^2$ . Since $\bar{v}^k_{cog} = \overline{V}^k_{cog} + \overline{err}^k$, and using the triangle inequality,

$$(\bar{R}_k[t] - \bar{R}_{cog}[t])^2 - (\bar{R}_k[t+1] - \bar{R}_{cog}[t])^2 \geq \left(V^k_{cog} - (\rho^k + err^k)\right)^2 \ . \quad (4)$$

Recall that since $k$ is the most distant robot, $V^k_{cog} = D_{cog}$. Denoting $\gamma = 1 - f(\epsilon_0)$, we have by (3) that

$$V^k_{cog} - (\rho^k + err^k) \ \geq \ \gamma \cdot D_{cog} \ . \quad (5)$$

As mentioned above, if $\epsilon_0 < 0.2$, then $\gamma > 0$. We also use the fact that $I[t] = I_{\bar{x}}[t] \leq D^2_{cog}$. Together with Fact 4 and inequalities (4) and (5), we have that

$$\begin{aligned}
I[t+1] \leq I_{\bar{x}}[t+1] \ &= \ \frac{1}{N}\left( (\bar{R}_k[t+1] - \bar{R}_{cog}[t])^2 + \sum_{j \neq k} (\bar{R}_j[t+1] - \bar{R}_{cog}[t])^2 \right) \\
&\leq \frac{1}{N}(\bar{R}_k[t+1] - \bar{R}_{cog}[t])^2 + \frac{1}{N}\sum_{j \neq k}(\bar{R}_j[t] - \bar{R}_{cog}[t])^2 \\
&\leq \frac{1}{N}(\bar{R}_k[t+1] - \bar{R}_{cog}[t])^2 - \frac{1}{N}(\bar{R}_k[t] - \bar{R}_{cog}[t])^2 + I[t] \\
&\leq I[t] - \frac{1}{N}\left(V^k_{cog} - (\rho^k + err^k)\right)^2 \ \leq \ I[t] - \frac{\gamma^2}{N} \cdot D^2_{cog} \ \leq \ I[t]\left(1 - \frac{\gamma^2}{N}\right)
\end{aligned}$$

and therefore the system converges, proving the theorem. ∎

We now turn to the $\mathcal{ERR}$ model, allowing also inaccuracies in angle measurements, and observe that Theorem 5 can be extended to hold true in this model as well, with a suitable choice of $\epsilon_0$.

**Theorem 6.** *Taking* $\epsilon_0 > \sqrt{2(1-\epsilon)(1-\cos\theta_0) + \epsilon^2}$, *in every execution of Algorithm* RCG *in the* $\langle \mathcal{FSYNC}, \mathcal{ERR} \rangle$ *model, the robots converge.*

Turning to the semi-synchronous model, we observe that the results of Theorem 6 hold true also for the $\langle \mathcal{SSYNC}, \mathcal{ERR} \rangle$ model, yielding the following.

**Theorem 7.** *In every execution of Algorithm* RCG *(with $\epsilon_0$ as in Theorem 6) in the $\langle \mathcal{SSYNC}, \mathcal{ERR} \rangle$ model, the robots converge.*

Finally, let us turn to robots with movement and calculation inaccuracies. In case of robots with inaccurate movements, we assume it is always possible to tune the algorithm such that the distance traveled is always less than or equal to the distance aimed i.e., instead of moving by a vector $\bar{v}$, move by a vector $\nu\bar{v}$. Suppose now that $D$, the distance traveled by the robot is bounded by $(1-\alpha)R \leq D \leq (1+\alpha)R$, where $R$ is the norm of the output of the algorithm, and $\alpha$ is a constant denoting the accuracy of the robot's movement. $\nu$ can be chosen such that $(1+\alpha)\nu \leq 1$. The result can be obtained following the same line of proof. The details are deferred to the full paper. As for calculation inaccuracies, since we assume a multiplicative inaccuracy, and by the linearity of the calculation, it can be treated as a measurement inaccuracy with the proper addition to $\epsilon$.

**Analysis of** RCG **in the Fully Asynchronous Model.** So far, we have not been able to establish the convergence of Algorithm RCG in the fully asynchronous model. In this section we prove its convergence in the restricted one-dimensional case and with no angle inaccuracies, i.e., in the $\langle \mathcal{ASYNC}, \mathcal{ERR}^- \rangle$ model.

Denote by $\bar{c}_i[t]$ the calculated destination of robot $i$ at time $t$. If robot $i$ has not gone through a look yet, or has reached its previous destination then, by definition, $\bar{c}_i[t] = \bar{R}_i[t]$. Notice that we set $\bar{c}_i[t]$ to be the destination of the robot's motion after the look phase even if the robot has not yet completed its computation, and is still unaware of this destination.

**Theorem 8.** *In the $\langle \mathcal{ASYNC}, \mathcal{ERR}^- \rangle$ model, $N$ robots performing Algorithm* RCG *converge on the line.*

*Conjecture 1.* Algorithm RCG converges in the $\langle \mathcal{ASYNC}, \mathcal{ERR} \rangle$ model for sufficiently small error in the angle and distance measurements.

**Separating** Go_to_COG **from** RCG **in the $\mathcal{ASYNC}$ Model.** This section establishes the advantage of Algorithm RCG over the basic Algorithm Go_to_COG. In the fully synchronous case there is no justification for using the more involved Algorithm RCG, since the simpler Algorithm Go_to_COG also guarantees convergence as shown above in Lemma 2.

However, a gap between the two algorithms can be established in the fully asynchronous model. Specifically, we now show that the ordinary center of gravity algorithm Go_to_COG does not converge in the $\langle \mathcal{ASYNC}, \mathcal{ERR}^- \rangle$ model, even when the robots are positioned on a straight line. Contrasting this result with Theorem 8 yields the claimed separation between the two algorithms.

**Theorem 9.** *In the $\langle \mathcal{ASYNC}, \mathcal{ERR}^- \rangle$ model, for every $\epsilon$ and $N > 1/\epsilon$ there exists an activation schedule for which Algorithm* Go_to_COG *does not converge, even when the robots are restricted to a line.*

## References


1. N. Agmon and D. Peleg. Fault-tolerant gathering algorithms for autonomous mobile robots. In *Proc. 15th SODA*, 1063–1071, 2004.
2. H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita. A distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Trans. Robotics and Automation*, 15:818–828, 1999.
3. E. Arkin, M. Bender, S. Fekete, J. Mitchell, and M. Skutella. The freeze-tag problem: How to wake up a swarm of robots. In *Proc. 13th SODA*, 2002.
4. Y.U. Cao, A.S. Fukunaga, and A.B. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4(1):7–23, March 1997.
5. M. Cieliebak, P. Flocchini, G. Prencipe, and N. Santoro. Solving the robots gathering problem. In *Proc. 30th ICALP*, 1181–1196, 2003.
6. R. Cohen and D. Peleg. Convergence properties of the gravitational algorithm in asynchronous robot systems. *SIAM J. on Computing*, 34:1516–1528, 2005.
7. X. Defago and A. Konagaya. Circle formation for oblivious anonymous mobile robots with no common sense of orientation. In *Proc. 2nd ACM Workshop on Principles of Mobile Computing*, 97–104. ACM Press, 2002.
8. P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Hard tasks for weak robots: The role of common knowledge in pattern formation by autonomous mobile robots. In *Proc. 10th Int. Symp. on Algo. and Computation*, 93–102, 1999.
9. P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Gathering of autonomous mobile robots with limited visibility. In *Proc. 18th STACS*, 247–258, 2001.
10. B. V. Gervasi and G. Prencipe. Coordination without communication: The case of the flocking problem. *Discrete Applied Mathematics*, 143:203–223, 2004.
11. SensComp Inc. *Spec. of 6500 series ranging modules*. http://www.senscomp.com.
12. K. Sugihara and I. Suzuki. Distributed algorithms for formation of geometric patterns with many mobile robots. *J. of Robotic Systems*, 13(3):127–139, 1996.
13. I. Suzuki and M. Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM J. on Computing*, 28:1347–1363, 1999.