Convergence of Autonomous Mobile Robots With Inaccurate Sensors and Movements*

Reuven Cohen †   David Peleg ‡

October 5, 2007

Abstract

A number of recent studies concern algorithms for distributed control and coordination in systems of autonomous mobile robots. The common theoretical model adopted in these studies 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 theorems, limiting the inaccuracy allowing convergence, and prohibiting a general algorithm for gathering, namely, meeting at a point, in a finite number of steps. The main positive result is an algorithm for convergence under bounded measurement, movement and calculation errors.

*A preliminary version of this paper appeared in Proc. 23rd Int. Symp. on Theoretical Aspects of Computer Science, Feb. 2006, 549–560.

†Dept. of Physics, Massachusetts Institute of Technology, Cambridge, MA, USA. E-mail: `reuven@mit.edu`. Part of the work was done while this author was visiting the Weizmann Institute of Science. Supported in part by the Pacific Theaters Foundation.

‡Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, Rehovot 76100, Israel. E-mail: `david.peleg@weizmann.ac.il`. Supported by the Israel Science Foundation (grant No. 693/04).

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.

Subsequently, studies of autonomous mobile robot systems can be found in different disciplines, from engineering to artificial intelligence. (e.g., [17, 23, 24]). Notable engineering efforts in these directions include the Cellular Robotic System [18], swarm intelligence [5] and the self-assembly machine [20]. Relevant studies in the realm of artificial intelligence include social interaction and intelligent behavior [19], behavior based robot systems [4, 21, 22], multi robot learning [23, 25] and ant robotics [31]. (A survey on the area is presented in [6].)

A number of recent studies on autonomous mobile robot systems focus on algorithms for distributed control and coordination. Most of the studies mentioned above handled control and coordination issues following experimental, empirical or architectural approaches, which resulted in the design of heuristic protocols. Algorithmic aspects were handled implicitly, with little or no emphasis on formal analysis of the correctness, termination or complexity properties of the algorithms. During the last decade, various coordination related issues have been studied from a distributed computing point of view (cf. [2, 15, 27, 28, 29]). 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, (the model may assume perfect or limited visibility range),
(2) "Compute": execute the given algorithm, resulting in a goal point p_G, and
(3) "Move": travel towards the point p_G.

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. [8, 14, 15, 28]) 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, namely, storing information regarding history, become more complex when they need to be applied in a dynamic environment where the robots have different start times, i.e., they are activated in different cycles, or robots might be added to or removed from the system dynamically. In dynamic environments an algorithm relying on the outcome of previous rounds may enter an inconsistent state and fail to reach the desired result. In contrast, oblivious solutions are insensitive to changing conditions and thus require no modification in dynamic settings. 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., [26]. 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 to a high degree of 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. [7, 8, 15, 28, 29]). 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 $c > 0$ there must be a time t_c by which all robots are within distance of at most c of each other).

It is important to note that analyzing the effect of errors is not merely of theoretical value. In Section 3.1 we show that gathering cannot be guaranteed in environments with errors. In Section 3.2 we 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 6.10) 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 [28, 29] 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. Also, an algorithm was presented in [29] 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 [8, 15], and an algorithm for gathering $N \geq 5$ robots has recently been described in [7]. 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 [14]. In [2] a convergence algorithm was provided for any N, in limited visibility systems. The effect of sensor and control errors was also studied numerically in [2]. The natural gravitational algorithm based on going to the center of gravity, and its convergence properties, were studied in [10, 11] in the semi-synchronous and asynchronous models respectively.

Another class of problems studied, e.g., in [12, 13, 27, 28, 29] concerns the formation of geometric

Algorithm	Go_to_COG	RCG
$\mathcal{FSYNC}$	converges (Lemma 4.2)	converges (Thm. 4.8)
$\mathcal{SSYNC}$	?	converges (Thm. 4.10)
$\mathcal{ASYNC}$	diverges (Thm. 6.10)	converges in 1-dim (Thm. 6.8)

Table 1: The contribution of this paper. Rsults of convergence or divergence for the `Go_to_COG` and `RCG` algorithms under different timing models for robots with inaccurate sensors.

patterns. The robots are required to arrange themselves (approximately) in a simple geometric form (such as a circle, a simple polygon or a line segment) within a finite number of cycles. Algorithms were presented for enabling a group of robots to achieve such self-arrangement and even spread itself nearly evenly along the form shaped. Flocking (or "following the leader") is yet another task studied in the literature, where the robots are required to follow the movements of a predefined leader [15]. Distributed *search* by a group of robots after a (static or moving) target in a specified region is a potentially useful application for mobile robot systems. An important subtask in this context is achieving an *even distribution* of the robots, namely, requiring the robots to spread out uniformly over a specified region of a simple geometric shape. This problem has been studied in [27]. A related task of interest is *partitioning*, where the robots are required to organize themselves into a number of groups. An algorithm for this problem was also presented in [27]. A final example is the *wake-up* task, where a single initially awake robot must wake up all the others (with the help of those already waken). The *Freeze-Tag* problem, which is a paradigm for the distributed wakeup of a group of robots, was presented in [3, 30] and given a number of approximation algorithms.

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 in a model allowing measurement errors, 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 measurement error, and prove its correctness, first in the fully synchronous model, and then in the semi-synchronous model. In Section 5 we describe how movement and calculation errors can be treated. In Section 6 we consider the fully asynchronous model, where we analyze our algorithm in the one-dimensional case and compare it with the ordinary center of gravity algorithm. The main contributions are summarized in Table 1. We remark that the paper does not explicitly concern the question of establishing the convergence rate of our algorithm. Note, however, that our convergence proofs do imply certain lower bounds on the convergence rate (that is, upper bounds on the time to halve some measure of the dispersion). Some additional comments on this issue are deferred to the conclusions section.

2 The model

Robot operation cycle. 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".

- **Look**: Identify the locations of all robots in 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 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 towards the point p_G.

A common assumption made in a number of papers dealing with this model, known as *premature stopping*, is that the robot might stop before reaching its goal point p_G, but is guaranteed to traverse at least some minimal distance s (unless it has reached the goal first). For ease of presentation, we assume throughout most of this paper (particularly, in Sections 3 and 4, which deal with measurement errors), a slightly simpler model where the move step of a robot is ensured to bring it to its goal point p_G. We handle premature stopping in Section 5, when we discuss movement errors.

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.

The synchronization model. 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 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, where each robot moves at each cycle. In Section 6.2 we also discuss its performance in the fully asynchronous ($\mathcal{ASYNC}$) model, in which there is no global clock, and the robots operate independently of each other, at arbitrary and possibly nonuniform rates.

Modeling measurement imprecisions. Our model assumes that the robot's location estimation is imprecise, where in general, this imprecision can affect both distance and angle estimations. However, we make the following restrictive assumption.

Bounded impresicion assumption: The distance imprecision is bounded by some accuracy parameter $\varepsilon_{\rm d}$ known at the robot's design. Similarly, the imprecision in angle measurements is bounded by an accuracy parameter ε_θ (where it can always be assumed that $\varepsilon_\theta \leq \pi$).

Formally, the bound on 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 $(1-\varepsilon_{\rm d})V < v < (1+\varepsilon_{\rm d})V$. (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). Similarly, the bound on angle imprecision means that the angle θ between the actual distance vector $\overline{V}$ and the measured distance vector $\bar{v}$ satisfies $\theta \leq \varepsilon_\theta$, or alternatively, $\cos\theta = \frac{\overline{V}\bar{v}}{Vv} \geq \cos\varepsilon_\theta$.

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 and rotations, 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 $\bar{v}^i_j = (1+\sigma^i_j)\overline{V}^i_j$, where $-\varepsilon_{\rm d} < \sigma^i_j < \varepsilon_{\rm d}$ is the actual 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 $v^i_j = (1+\sigma^i_j)V^i_j$, where $-\varepsilon_{\rm d} < \sigma^i_j < \varepsilon_{\rm d}$ and the angle θ between $\overline{V}^i_j$ and $\bar{v}^i_j$ will satisfy $|\theta| \leq \varepsilon_\theta$.

Throughout, 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 $\sigma^i_j[t]$. The parameter t is omitted whenever clear from the context.

Modeling movement and calculation imprecisions. Other than inaccuracies in measurements, inaccuracies in movement and calculations should also be taken into account. For movement, we may assume that if the robot wants to move from its current location $\bar{R}_i$ to some goal point p_G then it will move on a vector in angle of at most $\varepsilon_\theta^{\mathrm{mv}}$ from the vector $\overline{r_i p_G}$ and to any distance d in the interval $d \in [1 - \varepsilon_{\mathrm{d}}^{\mathrm{mv}}, 1 + \varepsilon_{\mathrm{d}}^{\mathrm{mv}}]|\overline{r_i p_G}|$. When it calculates a goal point $p_G = (x, y)$, it will have a multiplicative error of up to ε_{c}. Since in the center of gravity algorithms presented below, the calculation error is bounded linearly in the calculated terms, these inaccuracies can be treated as global inaccuracies in the measurement instead with the same effect. 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 ε_{d}. Absolute errors in movement or calculation can not be treated, since even if the robots have almost converged, they may lead to wider spreading. Therefore, throughout most of the ensuing technical development, we will assume only measurement inaccuracies, where the treatment of movement and calculations inaccuracies can be conducted by assimilating them into the measurement errors.

Another often ignored aspect of robot motion is the existence of a consistent sense of direction. As mentioned earlier, following the commonly used model of [28, 29], it is assumed in this paper that the robots do not share a common sense of orientation. Let us remark, however, that this is an expected feature of real robots, since the robot's body is positioned in an angle dependent on the previous movement direction. Hence in practice, this feature may allow the usage of the the robot's direction as a finite state memory between robot movements. The algorithms presented herein do not rely on sense of direction. However, one of the impossibility results presented below (Theorem 3.5) relies on the inability of using sense of direction.

Technical lemmas. We make use of the following technical lemmas. The first lemma says that if a point $\bar{b}$ outside the unit circle moves towards the center of the circle and stops at its boundary, then it gets closer to any point $\bar{a}$ inside the circle. (See Fig. 1.)

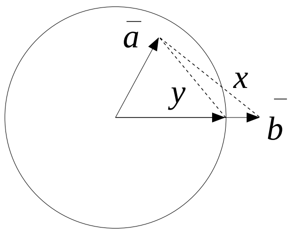

Figure 1: Illustration for Lemma 2.1.

Lemma 2.1 *For two vectors $\bar{a}$ and $\bar{b}$ with $a \leq 1 \leq b$, let $x = |\bar{a} - \bar{b}|$ and $y = |\bar{a} - \bar{b}/b|$. Then*

(1) $x^2 - y^2 \geq (b-1)^2 + 2(1-a)(b-1) \geq (b-1)^2$,

(2) $y \leq x$.

Proof: Note that $|\bar{a}-\bar{b}|^2 = a^2+b^2-2\bar{a}\bar{b}$ and $|\bar{a}-\bar{b}/b|^2 = a^2+1-2\bar{a}\bar{b}/b$. Thus

$$\begin{aligned} |\bar{a}-\bar{b}|^2 - |\bar{a}-\bar{b}/b|^2 &= b^2-1-2\bar{a}\bar{b}(1-1/b) \\ &\geq b^2-1-2a(b-1) \;=\; (b-1)^2+2(1-a)(b-1) \\ &\geq b^2-1-2(b-1) \;=\; (b-1)^2 \,, \end{aligned}$$

and part (1) of the lemma follows. Part (2) follows from part (1), as $(b-1)^2 \geq 0$. ∎

Lemma 2.2 *For vectors $\overline{V}$ and $\bar{v}$, if $(1-\varepsilon_d)V < v < (1+\varepsilon_d)V$ and $\frac{\overline{V}\bar{v}}{Vv} \geq \cos(\varepsilon_\theta)$, then $|\overline{V}-\bar{v}| < V\sqrt{2(1+\varepsilon_d)(1-\cos\varepsilon_\theta)+{\varepsilon_d}^2}$.*

Proof: Let $A = v/V$. Then $\bar{v}\overline{V} = AV^2\cos\theta$ where θ is the angle between the vectors, hence $(\bar{v}-\overline{V})^2 = \bar{v}^2+\overline{V}^2-2\bar{v}\overline{V} = (1+A^2)V^2-2AV^2\cos\theta$. In the range $A \in [1-\varepsilon_{\rm d}, 1+\varepsilon_{\rm d}]$ and $\theta \leq \varepsilon_\theta$, this function attains its maximum for $\theta = \varepsilon_\theta$ and $A = 1+\varepsilon_{\rm d}$, whereby $(\bar{v}-\overline{V})^2 = \big(1+(1+\varepsilon_{\rm d})^2-2(1+\varepsilon_{\rm d})\cos\varepsilon_\theta\big)\,V^2$, implying the lemma. ∎

The following Lemma is used in conjunction with Lemma 2.1 to show that moving some fraction of the distance to the circle perimeter will reduce the squared distance by at least the same fraction of the total improvement.

Lemma 2.3 *For any two vectors $\bar{A}$ and $\bar{B}$ staisfying $A^2-B^2 \geq c$, for some constant c, and for any $0 \leq \mu \leq 1$, the parameterized difference $X(\mu) = A^2-|\mu\bar{B}+(1-\mu)\bar{A}|^2$ satisfies $X(\mu) \geq \mu c$.*

Proof:

$$\begin{aligned} X(\mu) &= A^2-|\mu\bar{B}+(1-\mu)\bar{A}|^2 \;\geq\; A^2-(1-\mu)^2A^2-\mu^2B^2-2\mu(1-\mu)AB \\ &= \mu(A^2-B^2)+(\mu-\mu^2)(A-B)^2 \;\geq\; \mu(A^2-B^2) \;=\; \mu c \end{aligned}$$ ∎

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 [32], may be used to achieve gathering. However, this point is not computable, due to its infinite sensitivity to location errors [9].

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, no longer applicable when measurement or calculation errors occur.

3.1 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 instructions to the robots on how far to move in each round) is the same. Note that our impossibility proofs for gathering (Theorems 3.1 and 3.3) exploit the requirement of meeting at a point, and do not preclude the possibility of convergence. In contrast, Theorem 3.5 shows that sufficiently large inaccuracies in angle measurement, and in the absence of consistent sense of direction, even convergence is impossible, and the adversary can cause the robots to diverge under any algorithm.

Although this paper does not concern the issue of memory complexity, and all the algorithms presented hereafter are deterministic, it may be of interest to note that the following two impossibility theorems hold 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. The main idea of the proof is to divide the line into a set of segments, where each segment is small enough such that each robot may see each of its points as the location of the other robot due to measurement errors. Since the algorithm must perform the same movement (or a random movement chosen from the same distribution) it will fail to meet the other robot, or reach the next landmark for the algorithm, starting from almost any point in the segment.

Theorem 3.1 *Gathering is impossible for two robots on the line with inexact distance measurements even in the strong setting outlined above.*

Proof: Consider two robots 1 and 2 on the line in the $\mathcal{FSYNC}$ model and a potential gathering algorithm `ALG`. At each round t, the algorithm `ALG` at robot j only has one input $v^j_{j'}[t]$, namely, the result of the distance measurement taken by j to the other robot j'. Assuming the adversary ensures that in each round t the two robots obtain the same measurement result, i.e., $v^{j'}_j[t] = v^j_{j'}[t] = m[t]$, the inter-robot distance change $o(t)$ (namely, the distance reduction between the two robots) on round t becomes some function of $m[t]$, i.e., $o[t] = \varphi(m[t])$, and hence the total inetr-robot distance change $L(n)$ after n cycles is the sum of n outputs of the algorithm, $L(n) = \sum_{t \le n} o[t]$.

The adversary policy exploits this observation as follows. Partition the positive reals into infinitely many disjoint segments $(a_i, b_i]$, for $i \ge 1$, such that $b_i/a_i < (1+\varepsilon_\mathrm{d})/(1-\varepsilon_\mathrm{d})$ for each i. The adversary initially selects for every i a single possible measurement result m_i in the range $(1-\varepsilon_\mathrm{d})b_i < m_i < (1+\varepsilon_\mathrm{d})a_i$. During the execution of the algorithm, whenever a robot j takes a measurement where the true inter-robot distance $V^j_{j'}$ falls in the ith segment, $V^j_{j'} \in (a_i, b_i]$, the (inaccurate) measurement result will be $v^j_{j'} = m_i$.

This policy allows the introduction of only countably many distinct measurement results. Hence the total inter-robot distance change $L[t]$ after n cycles is the sum of a finite number of reals from a countable basis. The set of possible inter-robot distance changes is therefore still countable. As the set of possible initial inter-distances between the robots in the starting configuration is uncountable, it follows that gathering is never achieved from almost every starting configuration.

Observe that adding memory (allowing robots to store their previous measurements), or allowing the use of randomness, will not help, since now the output of the algorithm on round t is a function of a finite number of variables, $o[t] = \varphi(\langle m(t'), \rho(1,t'), \rho(2,t')\rangle_{t' \leq t})$, representing the entire history of the execution till this round, where $\rho(j,t')$ is the random number drawn by robot j on round t'. The results of each measurement are again taken from a countable set and the values of ρ for each run form a countable set. It follows that, fixing the sequence of random number pairs $\langle \rho(1,t'), \rho(2,t')\rangle$ drawn in the execution and considering all uncountably many possible starting configurations, the robots will achieve gathering only from a countable number of starting configurations, which are of zero measure. Therefore, the robots cannot gather even with constant probability. ∎

In the case of $N > 2$ robots on the line it is presumable that a similar construction can apply. However, in the plane, assuming exact angle measurements there is a continuum of possible measurements, and therefore it may be possible, using some method to estimate the distance using the angles (and possibly memory of several angles). We make the conjecture that (possibly with some limitations on the model) this does not help to achieve gathering.

Conjecture 3.2 *$N > 2$ robots with inaccurate distance measurements and accurate angle measurements in the $\mathcal{FSYNC}$ model cannot achieve convergence in the plane.*

In the case of angle and distance errors in the plane, gathering is impossible for any number of robots.

Theorem 3.3 *Gathering in the plane is impossible for any number of robots assuming inaccuracies in both the distance* and *angle measurements even in the strong setting outlined above.*

Proof: Consider N robots in the plane in the $\mathcal{FSYNC}$ model and a potential gathering algorithm `ALG`. At each round t, the algorithm `ALG` at robot j only has $N-1$ input $\bar{v}^j_{j'}(t)$, namely, the result of the measurement taken by j to any the other robot $1 \geq j' \neq j \leq N$. Assuming the adversary ensures that the robot j obtains the distance measurement results $m^j_{j'}[t]$ and angle measurements $\alpha^j_{j'}$ for the distance and angle to the j' robot, respectively, in each round t, the robot configuration $\bar{R}_j$ in round t becomes $\bar{R}_j[t+1] = \bar{R}_j[t] + \Delta\bar{R}_j[t]$, where $\Delta\bar{R}_j[t]$ is some function $\varphi(\{m^j_{j'}, \alpha^j_{j'}\})$ of $m^j_{j'}[t]$ and $\alpha^j_{j'}[t]$, and hence the total change $\bar{L}^j(n)$ in each robot j after n cycles is the sum of n outputs of the algorithm, $\bar{L}^j(n) = \sum_{t \leq n} \Delta\bar{R}_j[t]$.

The adversary policy exploits this observation as follows. Partition the positive reals into infinitely many disjoint segments $(a_i, b_i]$, for $i \geq 1$, such that $b_i/a_i < (1+\varepsilon_\mathrm{d})/(1-\varepsilon_\mathrm{d})$ for each i. Partition the range $[0, 2\pi)$ into the $n > 2\pi/\varepsilon_\theta$ segments $[0, 2\pi/n), \ldots, [(k\pi/n, (k+1)\pi/n), \ldots, [(n-1)\pi/n, 2\pi)$. The adversary initially selects for every i a single possible measurement result m_i in the range $(1-\varepsilon_\mathrm{d})b_i < m_i < (1+\varepsilon_\mathrm{d})a_i$, and for every k the adversary selects an angle θ_k such that

$(k+1)\pi/n - \varepsilon_\theta < \theta_k < k\pi/n + \varepsilon_\theta$. During the execution of the algorithm, whenever a robot j takes a measurement where the true inter-robot distance $V^j_{j'}$ falls in the ith segment, $V^j_{j'} \in (a_i, b_i]$, the (inaccurate) measurement result will be $v^j_{j'} = m_i$. Whenever the angle between a robot i and a robot j falls in the range $[(k\pi/n, (k+1)\pi/n)$ the angle will be measured as θ_k.

This policy allows the introduction of only countably many distinct measurement results. Hence each of the changes $\bar{L}^j(t)$ after n cycles is the sum of a finite number of real vectors from a countable basis. The set of possible robot location changes is therefore still countable. As the set of possible initial inter-distances between the robots in the starting configuration is uncountable, it follows that gathering is never achieved from almost every starting configuration.

Observe that adding memory (allowing robots to store their previous measurements), or allowing the use of randomness, will not help, since now the output of the algorithm on round t is a function of a finite number of variables, $\bar{L}^j(t) = \varphi(\langle m^j_{j'}(t'), \theta^j_{j'}, \rho(j,t')\rangle_{t' \leq t})$, representing the entire history of the execution till this round, where $\rho(j,t')$ is the random number drawn by robot j on round t'. The results of each measurement are again taken from a countable set and the values of ρ for each run form a countable set. It follows that, fixing the sequence of random number lists $\langle(\rho(j,t'))\rangle$ drawn in the execution and considering all uncountably many possible starting configurations, the robots will achieve gathering only from a countable number of starting configurations, which are of zero measure. Therefore, the robots cannot gather even with constant probability. ▮

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. We start with a technical lemma.

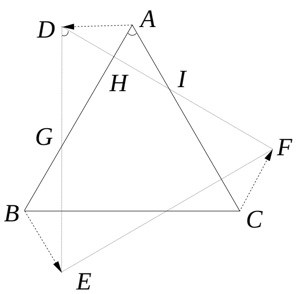

Figure 2: Illustration for Lemma 3.4.

Lemma 3.4 *If three robots are positioned in an equilateral triangular configuration and every robot moves the same distance counterclockwise parallel to the line formed by the locations of the other two robots, then the distances between the robots increase.*

Proof: A movement as in the lemma is illustrated in Fig. 2, where the robots move from the triangle $\triangle ABC$ to $\triangle DEF$, which by symmetry is also equilateral. Therefore $\angle BAC = 60°$ and $\angle EDF = 60°$. Now $\angle GHD = \angle FHA$. Thus, $\triangle HDG \sim \triangle HAI$. Since $DA \parallel BC$, also $\angle HAD = 60°$. Since $\angle HAI = 60°$, it follows that $\angle AHI < 120°$, and since it is external to $\triangle AHD$, $\angle HDA <$

$60° = \angle HAD$. This implies that $HD > HA$ and thus also $area(\triangle HDG) > area(\triangle HAI)$. By similar considerations on the other two corners, it follows that $area(\triangle DEF) > area(\triangle ABC)$. ∎

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

Proof: Start with a configuration with the robots positioned at the vertices of an equilateral triangle. The adversary can distort the measurements of each robot in such a way that it sees the locations of the other two robots as if its own location is exactly at the center of the line segment connecting them. This situation is symmetric under 180° rotation around the robot. Therefore, assuming a deterministic algorithm, the output vector $\bar{v}$ depends only on the robot's coordinate system. By rotating the coordinate system by 180° the adversary can always ensure that the vector of movement is in the half plane external to the triangle. By symmetry, the adversary can enforce a similar movement vector (rotated by 120°) for each of the three robots. In the resulting configuration, the robots will again be positioned at the vertices of an equilateral triangle (see Fig. 3). Moreover, each of these movement vectors can be split into two movements: a movement away from the center, which will trivially increase the distances, and movement perpendicular to the line connecting the robot to the center, which by Lemma 3.4 will also increase the distances. Hence, the adversary can cause the algorithm to diverge. ∎

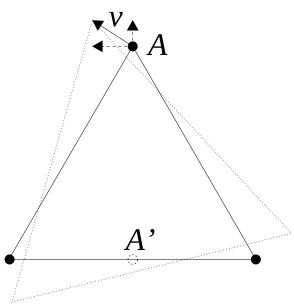

Figure 3: Illustration for Theorem 3.5. Robot A observes its environment and deduces it is in location A'. It applies the algorithm and moves by vector v (whose projections are dashed). By symmetry the final configuration is the (larger) dotted triangle.

3.2 Problems with existing algorithms

To illustrate the second point raised in the beginning of this section, consider the algorithm $3 - \texttt{Gather}$ presented in [1]. This algorithm achieves gathering of three robots using the following three simple rules:

(1) if two robots already reside in the same point, then the third should go to that point.
(2) If the robots form an obtuse triangle, they move towards the obtuse angled vertex.
(3) If the robots form an acute triangle, they move towards the intersection point of the angle bisectors.

As shown above, no algorithm can guarantee gathering when measurement errors occur. Furthermore, by Thm. 3.5, no algorithm can guarantee even convergence when angle measurement errors of $\varepsilon_\theta \geq 60°$ might occur. We now show that even though Algorithm $3-$`Gather` is designed to robustness and achieves gathering even if one of the robots fails, we have the following.

Observation 3.6 *Algorithm* $3-$`Gather` *might fail to achieve convergence in the presence of angle measurement errors of at least* $\varepsilon_\theta \geq 15°$.

Proof: Suppose the three robots form an equilateral triangle. Due to measurement errors of 15° to each of its neighbors, each robot may think its corner of the triangle forms a 90° angle, and subsequently conclude that it needs not move. Thus, a deadlock occurs. ∎

Likewise, for a group of $N > 3$ robots the algorithm $\mathtt{N}-$`Gather` is presented in [1]. In this algorithm (which is more complex and will not be described here), 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 the following.

Observation 3.7 *Algorithm* $\mathtt{N}-$`Gather` *might fail to achieve even convergence in the presence of angle and distance measurement errors of* $\varepsilon_d > 0$.

Proof: Suppose the $N > 3$ robots form a regular N-gon. All robots are on the smallest enclosing circle. Any $\varepsilon_{\rm d} > 0$ error in the measurements taken by each of the robots may cause it to believe it is located inside the circle while all others are on the circle. Therefore, a deadlock can occur. ∎

4 The convergence algorithm

4.1 Algorithm `Go_to_COG`

Arguably, the most natural algorithm for autonomous robot convergence is the gravitational algorithm, where each robot computes the average position (center of gravity) of the group as perceived by it,

$$\bar{v}^i_{cog} = \frac{1}{N}\sum_j \bar{v}^i_j ,$$

and moves towards it. A formal definition of this algorithm follows.

The properties of Algorithm `Go_to_COG` in a model with fully accurate measurements have been studied in [11]. In particular, the following theorem has been proven therein.

Algorithm `Go_to_COG` (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. Move to the point $\bar{v}^i_{cog}$.

Theorem 4.1 **[11]** *A group of N robots executing Algorithm* `Go_to_COG` *will converge in the $\mathcal{ASYNC}$ model with no measurement errors.*

If distance measurements are not guaranteed to be accurate (but angle measurements are accurate, i.e., $\varepsilon_\theta = 0$), Algorithm `Go_to_COG` may not guarantee convergence. Nevertheless, as shown next, convergence *is* guaranteed in the fully synchronous 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

$$\overline{V}^i_{cog} \;=\; \bar{R}_{cog} - \bar{R}_i \;=\; \frac{1}{N}\sum_j \overline{V}^i_j \ ,$$

recalling that $\overline{V}^i_j = \bar{R}_j - \bar{R}_i$. 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 have the following.

Lemma 4.2 *In the $\langle \mathcal{FSYNC}, \mathcal{ERR}^- \rangle$ model with $\varepsilon_d < \frac{1}{2}$, a group of N robots performing Algorithm* `Go_to_COG` *converges.*

Proof: At every time step, each robot moves to its perceived center of gravity, $\bar{v}^i_{cog}$. For every robot, the perceived center of gravity satisfies $|\bar{v}^i_{cog} - \overline{V}^i_{cog}| \le \varepsilon_{\rm d} D_{cog}$. Therefore, after step t, all robots are concentrated in a circle of radius $\varepsilon_{\rm d} D_{cog}[t]$ around $\bar{R}_{cog}[t]$. By convexity, the new center of gravity $\bar{R}_{cog}[t+1]$ also falls in this circle, hence the maximum distance between a robot and the new center of gravity is at most $D_{cog}[t+1] \le 2\varepsilon_{\rm d} D_{cog}[t]$. Hence as long as $\varepsilon_{\rm d} < \frac{1}{2}$, $D_{cog}[t+1] \le (1-\eta)D_{cog}[t]$ for some constant $\eta > 0$. Thus, the robots converge. ▮

The convergence of Algorithm `Go_to_COG` in the $\mathcal{SSYNC}$ model is not clear at the moment. However, as shown below in Section 6.2, 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.

4.2 Algorithm RCG

Our algorithm, named `Restricted_Go_to_COG`, or `RCG` for short, 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, $\epsilon_0 > \varepsilon_{\mathrm{d}}$. We also fix a parameter $0 < \beta \leq 1$, controlling the rate of convergence. In Subsection 4.4, dealing only with measurement inaccuracies, we assume $\beta = 1$. Later on, in Section 5, this parameter needs to be adjusted.

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 farthest robot is found. Notice that this distance may not be accurate, and that this might not be even the real farthest 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 robot decides to stay in place 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, as illustrated in Fig. 4.

A formal description of the algorithm is given next. Note that Algorithm `Go_to_COG` is identical to Algorithm `RCG` with parameter $\epsilon_0 = 0$.

Algorithm `RCG` (Code for robot i):

1. Estimate the measured center of gravity, $\bar{v}^i_{cog} = \dfrac{1}{N}\sum_j \bar{v}^i_j$
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 between calculated and actual CoG
   and $\mathcal{F} = 1 - \rho^i / v^i_{cog}$ /* safe movement fraction
4. If $\mathcal{F} > 0$ then move to the point $\bar{c}_i = \beta \cdot \mathcal{F} \cdot \bar{v}^i_{cog}$.
   Otherwise do not move.

4.3 Analysis of RCG for measurement errors in the $\mathcal{FSYNC}$ model

We first prove the convergence of Algorithm `RCG` in the $\langle \mathcal{FSYNC}, \mathcal{ERR}^- \rangle$ model. We use the following two properties of D^i_{max}.

Fact 4.3 *For every i, the true and perceived maximum distances satisfy*

(a) $(1-\epsilon_0)D^i_{max} < (1-\varepsilon_d)D^i_{max} \leq d^i_{max} \leq (1+\varepsilon_d)D^i_{max} < (1+\epsilon_0)D^i_{max}$,

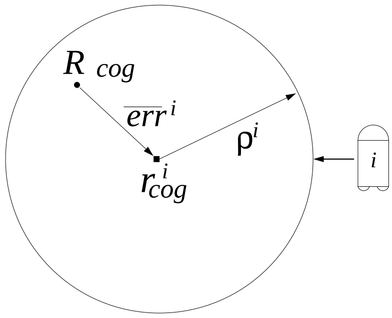

Figure 4: Illustration of the algorithm.

(b) $D_{cog} \le D^i_{max} \le 2D_{cog}$.

Proof: The first property is immediate from the definition of D^i_{max}, the choice of d^i_{max} and the assumption. For the lower bound in the second property notice that the center of gravity is in the convex hull of the robot group. For the upper bound in the second property, let $\ell = \arg\max_j\{V^i_j\}$. By the triangle inequality for each j, we have

$$D^i_{max} \;=\; V^i_\ell \;\le\; \frac{1}{N}\sum_j (V^i_j + V^\ell_j) \;=\; V^i_{cog} + V^\ell_{cog} \;\le\; 2D_{cog}. \qquad \blacksquare$$

For the synchronous model, we define the tth 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 the function $I_{\bar{x}}[t] \equiv \frac{1}{N}\sum_j(\bar{R}_j[t] - \bar{x})^2$, we notice the following fact (cf. [16]).

Fact 4.4 *$I_{\bar{x}}[t]$ attains its minimum on $\bar{x} = \bar{R}_{cog}[t]$.*

We now identify a key property required to ensure convergence of Algorithm RCG. For some time t, denote the *error component* in the center of gravity calculation by robot i by

$$\overline{err}^i \;=\; \frac{1}{N}\sum_j \sigma^i_j \overline{V}^i_j \;.$$

We refer to the ratio err^i/D_{cog} at time t as the *error ratio* of the algorithm at t. Algorithm RCG is said to have *bounded error ratio* if

$$\frac{err^i}{D_{cog}} \;<\; 2\epsilon_0 \quad \text{at any time } t. \tag{1}$$

Our main two lemmas, presented next, relate the bounded error ratio property to convergence.

Lemma 4.5 *In the $\langle \mathcal{FSYNC}, \mathcal{ERR}\rangle$ model, if $\epsilon_0 < 0.2$ and Algorithm* RCG *has bounded error ratio, then the algorithm guarantees that at every round t:*

1. *at least one robot can move,*

2. *every robot i that makes a move 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. By assumption, the algorithm satisfies Inequality (1) at time t, and by the two parts of Fact 4.3, the calculated value ρ^i is bounded by

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

By assumption (1), we have for each i,

$$err^i + \rho^i \leq f(\epsilon_0) \cdot D_{cog} < D_{cog} , \qquad (2)$$

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 Inequality (2) and the triangle inequality,

$$\rho^k < V^k_{cog} - err^k \leq v^k_{cog} .$$

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 4.3(a),

$$err^i[t] \leq \frac{\varepsilon_{\rm d}}{1-\varepsilon_{\rm d}} \cdot d^i_{max}[t] < \frac{\epsilon_0}{1-\epsilon_0} \cdot d^i_{max}[t] = \rho^i[t] ,$$

hence $a \leq 1$. 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 2.1(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 and Part 2, we have that

$$I_{\bar{x}}[t+1] < I_{\bar{x}}[t] = I[t] .$$

Finally, by Fact 4.4 we get $I[t+1] \leq I_{\bar{x}}[t+1]$, yielding Part 3 of the Lemma. ▮

Lemma 4.6 *In the $\langle \mathcal{FSYNC}, \mathcal{ERR} \rangle$ model, if $\epsilon_0 < 0.2$ and Algorithm* RCG *has bounded error ratio, then in every execution of the algorithm, the robots converge.*

Proof: By Part 1 of Lemma 4.5, 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 4.5, 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 2.1(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,

$$(\bar{R}_k[t] - \bar{R}_{cog}[t])^2 - (\bar{R}_k[t+1] - \bar{R}_{cog}[t])^2 \geq \left(V^k_{cog} - (\rho^k + err^k)\right)^2 \;. \tag{3}$$

Recall that since k is the most distant robot, $V^k_{cog} = D_{cog}$. Denoting $\gamma = 1 - f(\epsilon_0)$, we have by (2) that

$$V^k_{cog} - (\rho^k + err^k) \;\geq\; \gamma \cdot D_{cog} \;. \tag{4}$$

As mentioned above, if $\epsilon_0 < 0.2$, then $\gamma > 0$. We also use the fact that

$$I[t] = I_{\bar{x}}[t] \leq D^2_{cog} \tag{5}$$

Using Fact 4.4 and inequalities (3), (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. ▮

Next, our two main theorems, stating the convergence of Algorithm `RCG` in the $\mathcal{ERR}^-$ and $\mathcal{ERR}$ models respectively, are established by showing that a suitable selection of ϵ_0 ensures that the algorithm has bounded error ratio.

Theorem 4.7 *In the $\langle \mathcal{FSYNC}, \mathcal{ERR}^- \rangle$ model, if there exists a fixed ϵ_0 such that $\varepsilon_d < \epsilon_0 < 0.2$, then in every execution of Algorithm* `RCG`*, the robots converge.*

Proof: The center of gravity computed by robot i can be expressed as

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

By the bounded imprecision assumption and Fact 4.3(b),

$$err^i \;=\; \frac{1}{N}\left|\sum_j \sigma^i_j \cdot V^i_j\right| \;\le\; \frac{1}{N}\sum_j |\sigma^i_j| \cdot V^i_j \;\le\; \varepsilon_{\rm d} D^i_{max} \;\le\; 2\varepsilon_{\rm d} D_{cog} \;<\; 2\epsilon_0 D_{cog}. \tag{6}$$

The claim now follows from Lemma 4.6. ▮

We now turn to the $\mathcal{ERR}$ model, allowing also inaccuracies in angle measurements.

Theorem 4.8 *In the $\langle \mathcal{FSYNC}, \mathcal{ERR} \rangle$ model, if there exists a fixed ϵ_0 such that $\sqrt{2(1+\varepsilon_d)(1-\cos\varepsilon_\theta)+{\varepsilon_d}^2} \;<\; \epsilon_0 \;<\; 0.2$, then in every execution of Algorithm* RCG*, the robots converge.*

Proof: Recall that $\bar{v}^i_{cog} = \overline{V}^i_{cog} + \overline{err}^i$. By Lemma 2.2

$$\begin{aligned} err^i \;=\; |\bar{v}^i_{cog} - \overline{V}^i_{cog}| \;&<\; V^i_{cog}\sqrt{2(1+\varepsilon_{\rm d})(1-\cos\varepsilon_\theta)+{\varepsilon_{\rm d}}^2} \\ &\le\; \sqrt{2(1+\varepsilon_{\rm d})(1-\cos\varepsilon_\theta)+{\varepsilon_{\rm d}}^2}\, D^i_{max} \\ &\le\; 2\sqrt{2(1+\varepsilon_{\rm d})(1-\cos\varepsilon_\theta)+{\varepsilon_{\rm d}}^2}\, D_{cog} \;<\; 2\epsilon_0 D_{cog} \,. \end{aligned}$$

Lemma 4.6 can thus be applied, completing the proof. ▮

4.4 Analysis of RCG for measurement errors in the $\mathcal{SSYNC}$ model

Turning to the semi-synchronous model, we observe that the results of Theorem 4.8 hold true also for the $\langle \mathcal{SSYNC}, \mathcal{ERR} \rangle$ model. We start with a lemma aiming to show that in any execution of Algorithm RCG, for any move of a group of robots that produces a shift of the center of gravity, the moment of inertia decreases.

Lemma 4.9 *For a group of N robots performing Algorithm* RCG *in the $\mathcal{SSYNC}$ model, if at time step t the center of gravity has moved such that $|\bar{R}_{cog}[t+1] - \bar{R}_{cog}[t]| \ge \Delta x$ for some Δx, then $I[t] - I[t+1] \ge 2\epsilon_0(1-a)\Delta x D_{cog}[t]$ for some constant $a < 1$.*

Proof: Consider a robot i moving along a vector $\bar{y}^i$ (of size y^i) at time t, and let $\rho^i[t]$ be the calculated maximum error for this robot at time t. Fix

$$\bar{a} \;=\; \overline{err}^i[t]/\rho^i[t] \qquad \text{and} \qquad \bar{b} \;=\; \bar{v}^i_{cog}/\rho^i[t] \,.$$

We therefore have that $a \;=\; err^i[t]/\rho^i[t]$ and, since we know that the motion of the robot is of size y^i, it follows that $|\bar{b} - \bar{b}/b| = y^i/\rho^i[t]$. Thus, $b = 1 + y^i/\rho^i[t]$. By Eq. (6), $err^i \le \varepsilon_{\rm d} D^i_{max} \le \varepsilon_{\rm d} d^i_{max}/(1-\varepsilon_{\rm d})$ (with $\varepsilon' \equiv \sqrt{2(1+\varepsilon_{\rm d})(1-\cos\varepsilon_\theta)+{\varepsilon_{\rm d}}^2}$ replacing $\varepsilon_{\rm d}$ for the $\mathcal{ERR}$ model), while

$\rho^i = \epsilon_0 d^i_{max}/(1-\epsilon_0)$. This implies that $a \le \frac{\varepsilon_{\rm d}(1-\epsilon_0)}{\epsilon_0(1-\varepsilon_{\rm d})} < 1$ (where we use the fact that $\epsilon_0 > \varepsilon_{\rm d}$). Also, clearly $b \ge 1$. Hence Lemma 2.1(1) can be applied, yielding

$$(\bar{a}-\bar{b})^2 - (\bar{a}-\bar{b}/b)^2 \ge (b-1)^2 + 2(1-a)(b-1) \ge 2(1-a)(b-1) = 2(1-a)y^i/\rho^i[t].$$

Now, $\bar{a}-\bar{b} = (\bar{R}_i[t]-\bar{R}_{cog}[t])/\rho^i[t]$ and $\bar{a}-\bar{b}/b = (\bar{R}_i[t+1]-\bar{R}_{cog}[t])/\rho^i[t]$. Thus,

$$(\bar{R}_i[t]-\bar{R}_{cog}[t])^2 - (\bar{R}_i[t+1]-\bar{R}_{cog}[t])^2 \;\ge\; 2(1-a)y^i\rho^i[t] \;.$$

Note that

$$\rho^i[t] \;=\; \frac{\epsilon_0}{1-\epsilon_0} d^i_{max}[t] \;\ge\; \epsilon_0 D^i_{max}[t] \;\ge\; \epsilon_0 D_{cog}[t] \;, \tag{7}$$

and therefore $(\bar{R}_i[t]-\bar{R}_{cog}[t])^2 - (\bar{R}_i[t+1]-\bar{R}_{cog}[t])^2 \ge 2(1-a)y^i\epsilon_0 D_{cog}[t]$. The total decrease in I is therefore

$$I[t]-I[t+1] \ge I[t]-I_{\bar{x}}[t] \ge \frac{1}{N}\sum_i 2(1-a)y^i\epsilon_0 D_{cog}[t], \tag{8}$$

with $\bar{x} = \bar{R}_{cog}[t]$.

Now, we know that the center of gravity has moved by at least Δx, and therefore,

$$\Delta x \le |\bar{R}_{cog}[t+1]-\bar{R}_{cog}[t]| = \frac{1}{N}\left|\sum_i y^i\right| \le \frac{1}{N}\sum_i |y^i| \;.$$

Thus, using Eq. (8), one obtains, $I[t]-I[t+1] \ge 2(1-a)\Delta x\epsilon_0 D_{cog}[t]$ ∎

We now turn to prove the main theorem for the $\mathcal{SSYNC}$ model.

Theorem 4.10 *In every execution of Algorithm* RCG *(with ϵ_0 as in Theorem 4.8) in the $\langle\mathcal{SSYNC},\mathcal{ERR}\rangle$ model, the robots converge.*

Proof: The proof of Lemma 4.5 holds for any movement of any robot, and therefore also to any partial robot group making a move. Thus, it applies also to the semi-synchronous model. In Lemma 4.6, it may happen that the robot most distant from the center of gravity is inactive and does not make a move for a number of steps, during which the situation changes and the center of gravity approaches it due to movements taken by the other robots. Hence, we have to deal with the complication arising from the possibility that the most distant robot at some time t' is no longer the most distant when its turn to move arrives at some later time t.

Suppose at time t' this robot, k, was at distance $D_{cog}[t']$ from the center of gravity. Now take $t > t'$ as the time of its next activation. Take $\delta = \frac{1}{3}\left[1-f(\epsilon_0)\right]$. If $V^k_{cog}[t] > (1-\delta)D_{cog}[t']$ and $D_{cog}[t] < (1+\delta)D_{cog}[t']$, then $V^k_{cog}[t]/D_{cog}[t] > 1-2\delta$. This implies that robot k can still make a move at time t, since

$$\rho^k[t] \;=\; \frac{\epsilon_0}{1-\epsilon_0}\cdot d^k_{max}[t] \;\le\; 2\frac{\epsilon_0}{1-\epsilon_0}(1+\varepsilon_{\rm d})D_{cog}[t] \;<\; \frac{f(\epsilon_0)}{1-2\delta}V^k_{cog}[t] \;<\; V^k_{cog}[t] \;, \tag{9}$$

where the last inequality holds as long as $f(\epsilon_0) < 1$. We may now use Lemma 2.1(1), with $b = V^k_{cog}[t]/\rho^k[t] > (1-2\delta)/f(\epsilon_0)$, leading to

$$(\bar{R}_k[t] - \bar{R}_{cog}[t])^2 - (\bar{R}_k[t+1] - \bar{R}_{cog}[t])^2 \;\geq\; (v^k_{cog}[t] - \rho^k)^2 \geq \left(\frac{1-2\delta}{f(\epsilon_0)} - 1\right)^2 \rho^k[t] \;.$$

Eq. (7) implies that $\rho^k[t] \geq \epsilon_0 D_{cog}[t] \geq (1-2\delta)\epsilon_0 D^k_{cog}[t']$, and thus

$$I[t+1] \;\leq\; \left[1 - \frac{1}{N}\left(\frac{1-2\delta}{f(\epsilon_0)} - 1\right)^2 [(1-2\delta)\epsilon_0]^{-2}\right] I[t'].$$

If one of the requirements is violated, this means that the center of gravity has moved at least a distance of $\delta D_{cog}[t']$ in some time interval $[t', t_1] \subset [t', t]$. Choose t_1 to be the first time at which the center of gravity has moved at least $\delta D_{cog}[t']$. Then, for every time $t^* \in [t', t_1 - 1]$, $D_{cog}[t^*] \geq (1-\delta)D_{cog}[t']$. Denote by $\overline{\Delta x}[t^*]$ the change in the center of gravity between cycles t^* and $t^* + 1$. We have

$$\sum_{t^* \in [t', t_1 - 1]} \Delta x[t^*] \;\geq\; \left| \sum_{t^* \in [t', t_1 - 1]} \overline{\Delta x}[t^*] \right| \;\geq\; \delta D_{cog}[t'] \;.$$

By Lemma 4.9 we have that for all t^*,

$$I[t^*] - I[t^* + 1] \geq 2(1-a)\Delta x[t^*]\epsilon_0 D_{cog}[t^*] \geq 2(1-a)\Delta x[t^*]\epsilon_0(1-\delta)D_{cog}[t'].$$

Therefore, the total improvement in I is

$$\begin{aligned}
I[t'] - I[t] \quad &\geq \quad I[t'] - I[t_1] \;\geq \sum_{t^* \in [t', t_1 - 1]} 2(1-a)\Delta x[t^*]\epsilon_0(1-\delta)D_{cog}[t'] \\
&\geq \quad 2(1-a)\delta D_{cog}[t']\epsilon_0(1-\delta)D_{cog}[t'] \;\geq\; 2(1-a)\delta\epsilon_0(1-\delta)I[t'] \;,
\end{aligned}$$

hence $I[t] \leq (1-\eta)I[t']$ for some constant $\eta > 0$. Thus, the robots converge. ▮

5 Handling movement and calculation errors

We now turn to treat the case of movement errors. Assume the robots' motion is inaccurate, where the robot aiming to move a distance d_0 at an angle α_0 will actually move a distance d satisfying $ad_0 < d < bd_0$ at an angle α satisfying $|\alpha - \alpha_0| < \delta$, for constants $0 < a < 1$, $b > 1$ and $\delta > 0$.

As mentioned earlier, another possible type of movement error discussed in the literature is that the robot might halt its movement prematurely (it is commonly assumed that the movement is guaranteed to traverse at least some constant distance s).

Movement distance errors may be readily treated as measurement errors. In fact, movement distance errors are somewhat less severe, as they may not cause the robot to move in the wrong direction, but only move an inaccurate distance in the correct direction. Therefore, no limit on the size of the error is needed (other than b being finite and $a > 0$), and the only requirement necessary for ensuring convergence of the algorithm is to set the calibration parameter β to $\beta = 1/b$. We have the following.

Theorem 5.1 *Robots having movement errors performing Algorithm* `RCG` *will converge to a point.*

Proof: A robot having an angular movement error of $\alpha - \alpha_0 = \theta$ performing algorithm `RCG` will make exactly the same movement as a robot having accurate movement and viewing all other robots with an error of θ in the angle measurement (possibly in addition to any measurement errors included in the model). Therefore, a model with motion angle errors is reducible to a model with $\varepsilon_\theta = \varepsilon_\theta^{\text{mv}}$. In general, if both measurement and movement errors occur, then the model is reducible (for Algorithm `RCG`) to a model with total angle measurement inaccuracy of $\varepsilon_\theta + \varepsilon_\theta^{\text{mv}}$.

To adjust for the motion distance error, the parameter β is modified to $\beta = 1/b$ (or any lower value). Thus, every robot attempts to move to the point $\bar{c}_i = \frac{1}{b}(1 - \rho^i/v_{cog}^i) \cdot \bar{v}_{cog}^i$. This guarantees that the distance travelled is always less than $b\bar{c}_i = (1 - \rho^i/v_{cog}^i) \cdot \bar{v}_{cog}^i$, and at least $a\bar{c}_i = \frac{a}{b}(1 - \rho^i/v_{cog}^i) \cdot \bar{v}_{cog}^i$, a constant multiple of the distance travelled in the original algorithm. Furthermore, since the calculation of ρ^i is not influenced by β, it is guaranteed that any robot that can move in the original (accurate movement) model can also move in the inaccurate movement model. Therefore, using Lemma 2.3, one determines that whenever in the accurate movement model a robot decreases its squared distance from the center of gravity by some amount c, in the inaccurate movement model its squared distance from the center of gravity will decrease by at least ca/b, guaranteeing a constant factor improvement in I. The case presented above in the model definition is equivalent to the case $b \equiv 1 + \varepsilon_{\text{d}}^{\text{mv}}$, $a \equiv 1 - \varepsilon_{\text{d}}^{\text{mv}}$. ∎

The second possible form of movement inaccuracies is the "premature stopping" model, presented above, in which the robot may fail to move the full distance determined by the algorithm, and stop prematurely. This model guarantees, however, that the robot will complete a movement of at least some constant distance, s, unless the distance determined by the algorithm is less than s, in which case the robot is guaranteed to complete its movement.

Theorem 5.2 *In the $\mathcal{FSYNC}$ and $\mathcal{SSYNC}$ models, a group of N robots performing algorithm* `RCG` *with premature stopping will converge to a point.*

Proof: We start with the $\mathcal{FSYNC}$ model. Take the robot k farthest from the center of gravity. There are two possibilities for this robot's movement.

1. Robot k completes its movement. This case is analyzed exactly the same as in Theorem 4.7 or Theorem 4.8 (depending on the error model), and I can be shown to decreased by a constant multiplative factor.

2. Robot k travels some distance $s^* \geq s$ at time t. Assume the robot would have completed

the move determined by the algorithm, of some distance $d \le D_{cog}[t]$. Then, by Theorem 4.7 or Theorem 4.8, $I[t] - I[t+1] \ge \eta(N) I[t]$ for some $\eta(N)$. The ratio μ between the distance traveled by the robot and the distance determined by the algorithm satisfies

$$\mu \;=\; \frac{s^*}{d} \;\ge\; \frac{s}{d} \;\ge\; \frac{s}{D_{cog}[t]}\ .$$

By Lemma 2.3 the decrease in robot k's distance to the center of gravity by the actual movement is at least μ times the decrease due to the desired movement. Thus

$$I[t] - I[t-1] \;\ge\; \mu^2 \eta(N) I \;\ge\; \frac{s^2}{D_{cog}[t]^2}\eta(N) I[t]\ .$$

Since $I[t] \le D_{cog}[t]^2$, it follows that $I[t] - I[t-1] \ge \eta(N) s^2$.

Thus, at evey time step, I is decreased by either a multiplicative or an additive constant. The theorem follows.

In the case of the $\mathcal{SSYNC}$ model, as in the proof of Theorem 4.10 either the farthest robot k is at a distance at least $(1-\delta) D_{cog}[t']$ from the center of gravity when it performs its next move after time t', in which case it induces a multiplicative or an additive constant improvement in I (as in the $\mathcal{FSYNC}$ model above), or the center of gravity has moved a distance of at least $\delta D_{cog}[t']$, in which case I has improved by a multiplicative factor. ▮

Calculation errors and limited accuracy can be modeled in several ways. The most commonly used model is assuming that each operand, x_i, in each operation can include an additional error term, ε_{ci} which satisfies $|\varepsilon_{ci}| \le \varepsilon_c x_i$ for some ε_c.

We begin with a technical lemma relating inaccuracies in a vector's cpmponent with angle inaccuracies.

Lemma 5.3 *Let $\bar{a}$ be a vector with components (x, y), and $\bar{b}$ be a vector with components (x', y'), where $(1-\varepsilon)x < x' < (1+\varepsilon)x$ and $(1-\varepsilon)y < y' < (1+\varepsilon)y$ for some $0 < \varepsilon < 1$. The angle between $\bar{a}$ and $\bar{b}$ is at most $(\ln(1+\epsilon) - \ln(1-\epsilon))/2$ radians.*

Proof: If $x = 0$ or $y = 0$ then $x' = 0$ or $y' = 0$ respectively, so the angle between $\bar{a}$ and $\bar{b}$ is 0 and the theorem follows immediately. Assume now that $x \ne 0$ and $y \ne 0$ and denote $x' = (1+\alpha)x$ and $y' = (1+\beta)y$, where $-\epsilon < \alpha, \beta < \epsilon$. Notice that $\bar{a}$ and $\bar{b}$ are in the same quadrant. Now choose θ as the angle between $\bar{a}$ and the appropriate axis by the rule

$$\tan\theta = \begin{cases} |x|/|y|, & \alpha > \beta, \\ |y|/|x|, & \alpha \le \beta, \end{cases}$$

and set θ' to be the angle between the same axis and $\bar{b}$ defined by

$$\tan\theta' = \begin{cases} |x'|/|y'|, & \alpha > \beta, \\ |y'|/|x'|, & \alpha \le \beta. \end{cases}$$

Thus, $\tan\theta' = \frac{1+\alpha}{1+\beta}\tan\theta$ if $\alpha > \beta$ and $\tan\theta' = \frac{1+\beta}{1+\alpha}\tan\theta$ otherwise. Either way, we have $\tan\theta \le \tan\theta' \le \frac{1+\epsilon}{1-\epsilon}\tan\theta$, or

$$\ln\tan\theta \le \ln\tan\theta' \le \ln\tan\theta + \ln(1+\epsilon) - \ln(1-\epsilon). \tag{10}$$

Denote $\theta' = \theta + \Delta\theta$. Since tan is a continuous function with continuous derivative inside each quadrant, we have

$$\begin{aligned} \ln\tan(\theta+\Delta\theta) &= \ln\tan\theta + \left(\frac{d\ln\tan\theta}{d\theta}\right)_{\theta=\theta^*} \cdot \Delta\theta \;=\; \ln\tan\theta + \frac{1}{|\cos^2\theta|\tan\theta}\,\Delta\theta \\ &= \ln\tan\theta + \frac{2}{|\sin 2\theta|}\Delta\theta \;\ge\; \ln\tan\theta + 2\Delta\theta, \end{aligned} \tag{11}$$

for some $\theta \le \theta^* \le \theta + \Delta\theta$. From Eqs. (10) and (11) it follows that the angle $\Delta\theta$ between $\bar{a}$ and $\bar{b}$ satisfies $|\Delta\theta| \le (\ln(1+\epsilon) - \ln(1-\epsilon))/2$. ▮

Theorem 5.4 *A group of robots having inaccurate or limited accuracy calculations with $\varepsilon_c < 0.125$ (assuming accurate measurement and movement) executing Algorithm* `RCG` *will converge to a point.*

Proof: We use uppercase letters to represent real quantities and lowercase to represent calculated quantities. For simplicity of notation we assume exact measurements and inaccurate calculations. A combination of the two with limited errors is expected to behave similarly using the same lines of argument.

In step 1 of the calculation in Algorithm `RCG`, the additional errors are equivalent to inaccurate measurements, since each $\bar{v}^i_j$ is augmented with an error term of at most $\varepsilon_c \bar{v}^i_j$. To bound the error, we notice that the calculated $\bar{v}^i_{cog}$ for each robot i is

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

It is known that $\sigma^i_j \le \varepsilon_c v^i_j \le (1+\varepsilon_d)\varepsilon_c V^i_j$. Defining $\varepsilon_t = (1+\varepsilon_c)(1+\varepsilon_d) - 1$, and assuming the calculation is done by components (so the error bound holds for each component and therefore also to the vector size), it follows that

$$\left|\bar{v}^i_{cog} - \overline{V}^i_{cog}\right| \;=\; \frac{1}{N}\left|\sum_j (\overline{err}^i + \bar{\sigma}^i_j)\right| \;\le\; \frac{1}{N}\sum_j (|\overline{err}^i| + |\bar{\sigma}^i_j|) \;\le\; \frac{1}{N}\sum_j \varepsilon_t V^i_j \;\le\; \varepsilon_t D_{max}\ .$$

This implies that the inaccuracy in the calculated center of gravity appears in exactly the same form as in inaccurate measurements.

Step 2 of Algorithm `RCG` involves a comparison between vectors, which may be considered accurate up to the limited accuracy in the calculation of the vector sizes. This may lead to the calculated d^i_{max} having a relative error of at most epsilon. In the calculation performed in step 3 of the algorithm, another relative error of ε_c may be introduced, and yet another may occur in the calculation of ρ^i/v^i_{cog} in step 4. Therefore, it can be concluded that

$$\frac{\epsilon_0}{1-\epsilon_0}(1-\varepsilon_c)^3 d^i_{max} \;\le\; \rho^i \;\le\; \frac{\epsilon_0}{1-\epsilon_0}(1+\varepsilon_c)^3 d^i_{max} \;.$$

The choice of ϵ_0 should be large enough to ensure that the value of ρ^i resulting from the calculation of step 3 will always be greater than the total error in the target location (discussed below), but small enough to ensure that at least one robot can move at every configuration, i.e., such that the calculated ρ^i satisfies $\rho^i < D_{cog}$. This imposes the conditions $\frac{\epsilon_0}{1-\epsilon_0}(1-\varepsilon_c)^3 > \frac{\varepsilon_c}{1-\varepsilon_c}$ and $\frac{\epsilon_0}{1-\epsilon_0}(1+\varepsilon_c)^3 d^i_{max} < D_{cog}$ for at least one i. Since for at least one robot $d^i_{max} \ge 2(1-\varepsilon_d)D_{cog}$, it follows that we require $\frac{\epsilon_0}{1-\epsilon_0}(1+\varepsilon_c)^3(1-\varepsilon_d) < [2(1-\varepsilon_d)]^{-1}$. That is, we must choose ϵ_0 such that this condition holds.

Errors in step 4 of the algorithm are relative errors in the motion vector, and therefore are identical to movement errors, treated in Theorem 5.1. Since the calculated motion vector may contain a relative error of ε_c, leading to a factor of $1+\varepsilon_c$ in the motion vector size, β should be chosen such that $\beta < (1+\varepsilon_c)^{-1}$. By Lemma 5.3, an error of at most $\theta_c = (\ln(1+\varepsilon_c) - \ln(1-\varepsilon_c))/2$ may occur in the angle of motion, which is similar to the error in Theorem 5.1.

The total error in the aimed location of the center of gravity, using Lemma 2.2, is therefore

$$\sqrt{2(1+\varepsilon_t)(1-\cos(\theta_c+\varepsilon_\theta))+{\varepsilon_t}^2}\; D_{max} \;.$$

In the case of exact measurements, $\varepsilon_d = 0$, ϵ_0 should satisfy

$$\frac{\sqrt{2(1+\varepsilon_c)(1-\cos\theta_c)+{\varepsilon_c}^2}}{(1-\varepsilon_c)^3} \;<\; \frac{\epsilon_0}{1-\epsilon_0} \;<\; \frac{1}{2(1+\varepsilon_c)^3} \;.$$

For these inequalities to allow a solution, the upper bound should be higher than the lower bound. Thus, ε_c should satisfy $\varepsilon_c < 0.126\ldots$. When $\varepsilon_d > 0$, the maximum value of ε_c should be lower accordingly. ∎

It should be noted that unlike the measurement and movement inaccuracies discussed earlier, calculation errors are rather negligible. In particular, commercially available prcessors, even low-end ones, can be assumed to introduce very small numerical errors, hence it is safe to assume that for any practical purpose $\varepsilon_c < 10^{-3}$ in any real system. Consequently, the limit above poses no problem in practical applications.

6 Analysis of RCG in the fully asynchronous model

6.1 Convergence in the one-dimensional case

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.

Following are some technical lemmas and definitions used to prove the result, when the robots are on a straight line. Define $H[t]$ as the minimum segment containing all robots and destinations at time t.

Lemma 6.1 *In the* $\langle \mathcal{ASYNC}, \mathcal{ERR}^- \rangle$ *model,* $H[t] \subseteq H[t_0]$*, for all times* $t \geq t_0$*.*

Proof: By the proof of part 2 of Lemma 4.5 it follows that every robot approaches the true center of gravity, *i.e.*, in the one dimensional case, it always moves in the correct direction toward the center of gravity. Furthermore, the size of its motion is always an underestimate to the true distance. Therefore, the calculated destination resides between the current location of the robot and the real center of gravity, which is always in the segment. Thus no destination is calculated outside the segment. Therefore, no robot leaves the segment either. ▮

Next, we define the following quantities.

$$\begin{aligned}
\phi_1[t] &= \sum_{i=1}^{N} |\bar{R}_{cog}[t] - \bar{c}_i[t]| \ , \\
\phi_2[t] &= \sum_{i=1}^{N} |\bar{c}_i[t] - \bar{R}_i[t]| \ , \\
\phi[t] &= \phi_1[t] + \phi_2[t] \ , \\
h[t] &= |H[t]| \ , \\
\psi[t] &= \frac{\phi[t]}{2N} + h[t] \ .
\end{aligned}$$

We now claim that ϕ, h and ψ are non-increasing functions of time.

Lemma 6.2 *For every* $t_1 > t_0$*,* $\phi[t_1] \leq \phi[t_0]$*.*

Proof: Examine the change in ϕ due to the various robot actions. If a Look operation is performed by robot i at time t, then the destination $\bar{c}_i[t]$ is between the robot and the real center of gravity. Therefore $|\bar{R}_{cog}[t] - \bar{c}_i[t]| + |\bar{c}_i[t] - \bar{c}_i[t]| = |\bar{c}_i[t^*] - \bar{R}_{cog}[t]|$ for any $t^* \in [t', t]$, where t' is the end of the last move performed by robot i. Therefore, ϕ is unchanged by the Look performed.

Now consider some time interval $[t'_0, t'_1] \subseteq [t_0, t_1]$, such that no look operations were performed during $[t'_0, t'_1]$. Suppose that during this interval each robot i moved a distance Δ_i (where some of these distances may be 0). Then ϕ_2 decreased by $\sum_i \Delta_i$, the maximum change in the center of gravity is $|\bar{R}_{cog}[t_1] - \bar{R}_{cog}[t_0]| \leq \sum_i \Delta_i / N$, and the robots' calculated centers of gravity have not changed. Therefore, the change in ϕ_1 is at most $\phi_1[t_1] - \phi_1[t] \leq \sum_i \Delta_i$. Hence, the sum $\phi = \phi_1 + \phi_2$ cannot increase. ▮

Lemma 6.3 *ψ is a non-increasing function of time.*

Proof: By Lemma 6.2, ϕ is non-increasing. By Corollary 6.1, h is non-increasing. Therefore their sum is also non-increasing. ∎

Lemma 6.4 $h \leq \psi \leq 2h$.

Proof: The lower bound is trivial. For the upper bound, notice that ϕ is the sum of $2N$ summands, each of which is at most h (since they all reside in the segment). ∎

We now state a Lemma which allows the analysis of the change in ϕ (and therefore also ψ) in terms of the contributions of individual robots.

Lemma 6.5 *If by the action of a robot i separately, in the time interval $[t_0, t_1]$ its contribution to ϕ is δ_i, then $\phi[t_1] \leq \phi[t_0] + \delta_i$.*

Proof: Lemma 6.2 implies that "look" actions have no effect on ϕ and therefore can be disregarded. A robot moving a distance Δ_i will always decrease its term in ϕ_2 by Δ_i, and the motions of other robots have no effect on this term. Denote by Δ_j the motions of the other robots. Notice that

$$\left|\bar{R}_{cog} + \frac{\Delta_i}{N} + \frac{1}{N}\sum_{j \neq i} \Delta_j - \bar{c}_k\right| \leq \left|\bar{R}_{cog} + \frac{\Delta_i}{N} - \bar{c}_k\right| + \frac{1}{N}\sum_{j \neq i} |\Delta_j|.$$

The function ϕ_1 contains N summands, each of which contains a contribution of at most $\frac{1}{N}|\Delta_j|$ from every robot $j \neq i$. Therefore, the total contribution of each robot to ϕ_1 is at most $|\Delta_j|$, which is canceled by the negative contribution of $|\Delta_j|$ to ϕ_2. ∎

Lemma 6.6 *If at some time t_0, for some $0 < A < 1$, the convex hull of the robot group $H'[t]$ satisfies $|H'[t_0]| \leq (1 - A)h[t_0]$, then there exists a time $t > t_0$ such that $\psi[t] \leq \left(1 - \frac{A}{4N^2}\right)\psi[t_0]$.*

Proof: Take t^* to be the time after t_0 where all robots finished a complete Look–Compute–Move cycle. If for some time $t_1 \in [t_0, t]$ it held that $h[t_1] \leq (1 - \frac{A}{2})h[t_0]$, then $\psi[t_1] < \left(1 - \frac{A}{4N^2}\right)\psi[t_0]$ and we are done. If no such time (t_1) existed, then at time t_0 at least one robot had its calculated destination at a distance $\frac{A}{4}h[t_0]$ from the convex hull of the robot group. Suppose robot k had the most distant destination from the group, $\Delta_k \geq \frac{A}{4}h[t_0]$. By the movement of robot k alone, the real center of gravity moves a distance $\frac{\Delta_k}{N}$ towards its center of gravity. Therefore, by its motion ψ decreases by $\frac{2\Delta_k}{N^2}$. By Lemma 6.5 the decrease of ψ by the motion of all robots is at least the same, thus proving the claim. ∎

We now proceed to prove the main theorem regarding convergence in the asynchronous model.

Lemma 6.7 *For all times t_0 there exists a time $t_1 > t_0$ and a constant δ such that $\psi[t_1] \leq \left(1 - \frac{\delta}{16N^2}\right)\psi[t_0]$.*

Proof: We have established that ψ is a non-increasing function of time. Now assume that at time t_0, $H[t_0]$ is the convex hull of the locations of the robots and their destinations. Without loss of generality we assume $H[t_0] = [0,1]$. Take $t^* > t_0$ to be the time when all robots completed at least one full cycle. If at some time $t \in (t_0, t^*]$ the size of the interval occupied by the robots is $h[t^*] \le 1 - \delta$ we are done by Lemma 6.6.

Suppose now that at all times $t \in (t_0, t^*]$ the robots' interval never shrinks to $1-\delta$. This implies that there exist robots in the intervals $[0, \delta]$ and $[1-\delta, 1]$. Now, for each robot, i, there exists at least one robot, j, with distance at least $V_j^i \ge \frac{1-\delta}{2}$ from it and this robot is viewed at a distance at least $v_j^i \ge (1-\varepsilon_c)\frac{1-\delta}{2} \ge (1-\epsilon_0)\frac{1-\delta}{2}$. Therefore, it follows that $\rho^i \ge \frac{\epsilon_0(1-\delta)}{2} \ge \frac{\epsilon_0}{2}$, and thus no robot calculates its destination to within $\frac{\epsilon_0}{2}$ of $\{0,1\}$. If we take $\delta < \frac{\epsilon_0}{2}$ then no robot approaches a distance δ from the boundary.

Now take the leftmost and rightmost robots $k_l = \arg\min_i \bar{R}_i$ and $k_r = \arg\max_i \bar{R}_i$ and take $t_l > t^*$ and $t_r > t^*$ to be their next Look times. By assumption $\bar{R}_{k_l}[t^*] \in [0, \delta]$ and $\bar{R}_{k_r}[t^*] \in [1-\delta, 1]$. Assume, without loss of generality that $t_l \le t_r$. Then, for robot k_l, either $\bar{c}_{k_l}[t^*] \ge \delta$ or $\bar{c}_{k_l}[t_l] \ge \delta$, or otherwise robot k_l decided to maintain its position. This means that the real center of gravity at time t_l was within a distance $\rho_{k_l} + err_{k_l}$ of $\bar{R}_{k_l}$. For $\epsilon_0 < 0.2$, $\rho_{k_l} + err_{k_l} \le \left(\epsilon_0 + \epsilon_0 \frac{1+\epsilon_0}{1-\epsilon_0}\right) D_{max} < \frac{1}{2}$. Take $2\delta < \frac{1}{2} - \epsilon_0 - \epsilon_0 \frac{1+\epsilon_0}{1-\epsilon_0}$. Since $\bar{R}_{k_l} \in [0, \delta]$ it follows that the center of gravity at time t_l was in the interval $\bar{R}_{cog}[t_l] \in [0, \frac{1}{2} - \delta]$. Now again for k_r either $\bar{c}_{k_r}[t^*] \le 1 - \delta$ or $\bar{c}_{k_r}[t_r] \le 1 - \delta$ or the center of gravity has moved to the interval $[\frac{1}{2} + \delta, 1]$, in which case the center of gravity moved by at least 2δ approaching $\bar{c}_{k_r}\bar{R}_{k_r}$, leading to the desired decrease in ψ. Thus, no robot enters the δ environment of the boundary, and either one of the robots in this environment has left or ψ decreased by the desired amount. Since the number of robots is finite, after a finite number of such steps it is guaranteed that either ψ decreased by the desired amount, or the interval shrunk by at least δ, leading again to the desired decrease in ψ. ∎

Lemma 6.7 yields the theorem,

Theorem 6.8 *In the* $\langle \mathcal{ASYNC}, \mathcal{ERR}^- \rangle$ *model,* N *robots performing Algorithm* `RCG` *converge on the line.*

We make the following more general conjecture.

Conjecture 6.9 *Algorithm* `RCG` *converges in the* $\langle \mathcal{ASYNC}, \mathcal{ERR} \rangle$ *model for sufficiently small error in the angle and distance measurements.*

6.2 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 4.2.

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 6.8 yields the claimed separation between the two algorithms.

Theorem 6.10 *In the $\langle \mathcal{ASYNC}, \mathcal{ERR}^{-} \rangle$ model, for every ε_d and $N > 1/\varepsilon_d$ there exists an activation schedule for which Algorithm* `Go_to_COG` *does not converge, even when the robots are restricted to a line.*

Proof: Start with a configuration in which the first robot is at $\bar{R}_1 = 0$ and the other $N-1$ robots are located at $\bar{R}_i = 1$, for $i = 2 \ldots N$. Robot 1 makes a look, and sees the other robots at location $1 + \varepsilon_{\rm d}$. While robot 1 is at its Compute phase, the other robots go through a long sequence of cycles, leading them to a distance $\delta \ll \varepsilon_{\rm d}$ from robot 1. Robot 1 now finishes its compute phase, concludes that the center of gravity is at location $\frac{N-1}{N}(1+\varepsilon_{\rm d})$ and moves to this location. The robots are now at a similar setup to the initial setup, but with a distance of $1 - \frac{1}{N} + \frac{N-1}{N}\varepsilon_{\rm d} - \delta > 1$ between robot 1 and the rest of the robots. Repeating this process leads to the divergence of the algorithm. ▮

7 Conclusions

We have discussed the feasibility of robot swarm convergence under conditions of inaccuracy in the robots' sensors, calculations and movement. We have presented several impossibility results under various conditions, and have discussed the inadequancy of existing algorithms. We then presented an algorithm based on restricted movement to the center of gravity of the robot swarm, and have proven its correctness under a range of the inaccuracy parameters.

We have also shown that in the case of the one dimensional asynchronous model the restricted center of gravity algorithm guarantees convergence, while the standard center of gravity algorithm fails to converge under certain circumstances.

Establishing tight bounds for the convergence rate remains an open problem for future study. Our proofs provide only some trivial bounds stemming from the time required to traverse the distance between the two farthest robots. In the model allowing possible premature stopping of a robot after traversing a distance of at least S, this time (in terms of number of steps) is at least the distance between the two farthest robots divided by $2S$. Another trivial bound on the convergence rate of the center of gravity algorithm `Go_to_COG` follows from the ability of the adversary to increase or decrease the distance viewed from a robot to all other robots by a factor of $1 \pm \varepsilon_{\rm d}$. This implies that even in the fully synchronous model, instead of meeting at the center of gravity, after every step the robots may reach a configuration similar to their previous one, except with all distances multiplied by $\varepsilon_{\rm d}$

References

[1] N. Agmon and D. Peleg. Fault-tolerant gathering algorithms for autonomous mobile robots. In *Proc. 15th ACM-SIAM Symp. on Discrete Algorithms*, pages 1063–1071, January 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 ACM-SIAM Symp. on Discrete Algorithms*, 2002.

[4] T. Balch and R. Arkin. Behavior-based formation control for multi-robot teams. *IEEE Trans. on Robotics and Automation*, 14, December 1998.

[5] G. Beni and S. Hackwood. Coherent swarm motion under distributed control. In *Proc. DARS'92*, pages 39–52, 1992.

[6] Y.U. Cao, A.S. Fukunaga, and A.B. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4(1):7–23, March 1997.

[7] M. Cieliebak, P. Flocchini, G. Prencipe, and N. Santoro. Solving the robots gathering problem. In *Proc. 30th Int. Colloq. on Automata, Languages and Programming*, pages 1181–1196, 2003.

[8] M. Cieliebak and G. Prencipe. Gathering autonomous mobile robots. In *Proc. 9th Int. Colloq. on Structural Information and Communication Complexity*, pages 57–72, June 2002.

[9] E.J. Cockayne and Z.A. Melzak. Euclidean constructibility in graph minimization problems. *Math. Magazine*, 42:206–208, 1969.

[10] R. Cohen and D. Peleg. Robot convergence via center-of-gravity algorithms. In *Proc. 11th Colloq. on Structural Information and Communication Complexity*, volume LNCS 3104, pages 79–88. Springer-Verlag, 2004.

[11] R. Cohen and D. Peleg. Convergence properties of the gravitational algorithm in asynchronous robot systems. *SIAM J. on Computing*, 34:1516–1528, 2005.

[12] 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*, pages 97–104. ACM Press, 2002.

[13] 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 Algorithms and Computation*, pages 93–102, 1999.

[14] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Gathering of autonomous mobile robots with limited visibility. In *Proc. 18th STACS*, pages 247–258, 2001.

[15] B. V. Gervasi and G. Prencipe. Coordination without communication: The case of the flocking problem. *Discrete Applied Mathematics*, 143:203–223, 2004.

[16] H. Goldstein. *Classical Mechanics.* Addison-Wesley, Reading, MA, USA, 1980.

[17] D. Jung, G. Cheng, and A. Zelinsky. Experiments in realising cooperation between autonomous mobile robots. In *Proc. Int. Symp. on Experimental Robotics*, June 1997.

[18] Y. Kawauchi, M. Inaba, and T. Fukuda. A principle of decision making of cellular robotic system (CEBOT). In *Proc. IEEE Conf. on Robotics and Automation*, pages 833–838, 1993.

[19] M.J. Mataric. *Interaction and Intelligent Behavior.* PhD thesis, MIT, 1994.

[20] S. Murata, H. Kurokawa, and S. Kokaji. Self-assembling machine. In *Proc. IEEE Conf. on Robotics and Automation*, pages 441–448, 1994.

[21] L.E. Parker. Designing control laws for cooperative agent teams. In *Proc. IEEE Conf. on Robotics and Automation*, pages 582–587, 1993.

[22] L.E. Parker. On the design of behavior-based multi-robot teams. *J. Adv. Robotics*, 10, 1996.

[23] L.E. Parker and C. Touzet. Multi-robot learning in a cooperative observation task. In *Distributed Autonomous Robotic Systems 4*, pages 391–401, 2000.

[24] L.E. Parker, C. Touzet, and F. Fernandez. Techniques for learning in multi-robot teams. In *Robot Teams: From Diversity to Polymorphism, T. Balch and L.E. Parker, Eds.* 2001.

[25] L.E. Parker, C. Touzet, and F. Fernandez. Techniques for learning in multi-robot teams. In T. Balch and L.E. Parker, editors, *Robot Teams: From Diversity to Polymorphism.* A. K. Peters, 2001.

[26] SensComp Inc. *Specification of 6500 series ranging modules.* http://www.senscomp.com.

[27] K. Sugihara and I. Suzuki. Distributed algorithms for formation of geometric patterns with many mobile robots. *Journal of Robotic Systems*, 13(3):127–139, 1996.

[28] I. Suzuki and M. Yamashita. Distributed anonymous mobile robots - formation and agreement problems. In *Proc. 3rd SIROCCO*, pages 313–330, 1996.

[29] I. Suzuki and M. Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM J. on Computing*, 28:1347–1363, 1999.

[30] M. Sztainberg, E. Arkin, M. Bender, and J. Mitchell. Analysis of heuristics for the freeze-tag problem. In *Proc. 8th Scandinavian Workshop on Algorithm Theory*, pages 270–279, 2002.

[31] I.A. Wagner and A.M. Bruckstein. From ants to a(ge)nts. *Annals of Mathematics and Artificial Intelligence*, 31, special issue on ant-robotics:1–5, 1996.

[32] A. Weber. *Theory of the location industries.* The Univ. of Chicago Press, 1937.