Pattern formation algorithms for oblivious robots in presence of measurement errors

Kroneman, W.

Award date:
2020

Master Thesis: Pattern formation algorithms for oblivious robots in presence of measurement errors

Werner Kroneman

Abstract

Within the field of robotics, increasing interest is shown in the employment of distributed swarms of cheap, identical robots, as opposed to expensive and specialized monolithic units. Approaches vary from very practical to very theoretical; ranging from design of physical robots in the former case, to heuristics, simulations and to formal analysis in the latter. The OBLOT model has served as a de-facto standard framework for a subset of problems in the formal analysis setting.

However, while useful in the study of the theoretical boundaries of different problems and solutions, a gap exists between abstract, formal models and the practical application of knowledge gathered therein to simulations and real robots.

One such problem is that sensors may be inaccurate, seeing robots in a slightly different place than they really are. In this thesis, we explore such measurement errors in detail, and build an intuition for how to handle them by looking at how they affect specific solutions to the problems Gathering and Circle formation for three robots under the semi-synchronous scheduler, originally developed without taking measurement errors into account, and adapt them to produce a robust, bounded approximation of their original results.

Contents

1 Introduction

Traditionally, many tasks in robotics are carried out by large, complex, and specialized robots. While the use and availability of such specialized hardware has its advantages, it unfortunately proves problematic when new tasks are encountered since the development and production of such robots is a highly-expensive and labor-intensive task. Moreover, the risk of losing or damaging such a specialized piece of equipment is significant.

Over the last few decades, a trend has emerged of scientists exploring an alternative to such monolithic units: what if we could employ a large number of identical robots that, through their collective behavior, could accomplish a wide variety of tasks? Such robots could be mass-produced, and hence be a lot cheaper and replaceable individually. General-purpose units could prove to be more flexible as well by accomplishing new tasks through re-programming or only relatively minor modifications, as opposed to task-specific redesigns. They would especially be ideal in hostile and volatile environments such as disaster zones, where a swarm of robots could be programmed to accomplish the given task despite the possible loss of a few units.

Research into such alternative fields of robotics comes in many different forms, ranging from the very practical side of designing and manufacturing such robots, to the very theoretical side that explores what tasks can fundamentally be accomplished under different circumstances. Methods in the more theoretical side of the field range from more heuristics-based approaches that use simulations and experiments to highly-formalized approaches. Such formalized approaches offer the means of more easily accounting for every edge case, and are thus likely to produce highly-reliable results. These results, however, tend to be very theoretical; they can be used to inform such choices about whether it is necessary to include a particular sensor in a robot, but tend to not be easily applicable in the real world due to over-simplifications in the models used.

Among such formalized approaches, the OBLOT model [7] and variants thereof provide a de-facto standard theoretical environment for the study of pattern formation problems in space: problems where robots must either eventually form or converge to some particular geometric arrangement. In this model, robots are typically modeled as mobile punctiform entities with unlimited memory and computational power, but with extremely limited capabilities in terms of communication, mutual identification, synchronization, memory of the past, and orientation.

Such constraints will typically include:

Disorientation Robots may lack an initial consensus on a frame of reference. This disagreement may extend to the origin of the frame of reference, the orientation, the unit of measurement, or even the distinction between clockwise and counter-clockwise rotation (chirality).

Anonymity Robots may be unable to identify one another in any way other than position. Compounded by the possibility of disorientation, robots may be completely unable to designate or recognize any one particular robot. This makes tasks like assigning roles to any particular robot difficult.

Silence Robots may be unable to directly communicate with each other, or may be extremely limited in their ability to do so. While some variants of the OBLOT model allow robots a limited form of communication such as through colored lights [5], they are usually incapable of any direct kind of communication at all, such as through message-passing or even simple lights-based signaling. All that robots know about each other is their relative positions in each observer's local frame of reference.

Obliviousness Robots may lack any memory of the past, including positions of other robots, internal state, or actions taken. This prevents the use of any distributed algorithms that explicitly rely on past state, which may imply improved robustness against any transient problems or disturbances.

Reliability Robots may not always reliably perceive others or behave exactly as specified. Sometimes, a robot may experience a crash failure where they stop moving, a byzantine failure where they display arbitrary behavior, they may experience false-positives and false-negatives while observing others, have limited sensor range, or suffer from errors in their movements and in measurements of the positions of other robots.

As time goes on, robots will go through a potentially infinite sequence of *Look-Compute-Move* (LCM) cycles: they repeatedly take a "snapshot" of the positions of the robots around them, spend some time "computing" where to move based on this information, then move in accordance to this decision. The degree to which such cycles are synchronized varies, to the point where robots may even take this "snapshot" while other robots are moving, making that it has to accept a certain level of staleness or unpredictability in other robots' behavior. The OBLOT model is presented in greater detail in Section 2.1.

Typically, solutions to problems or tasks within the OBLOT model are analysed in an adversarial context: an intelligent adversary is imagined to be in control of any part of the model not fixed in advance which tries to use this control to block attempts to solve the given task. Such a setting presents a real challenge to algorithm designers, and any solutions found within such a context will therefore take into account any kind of rare, unusual, worst-case scenarios that may be missed in a more experimental approach.

The academic interest in the study of these kinds of constraints lies in discovering and understanding just how fundamentally limiting such constraints really are, and how they may be overcome. Such knowledge can then be applied in robot design: while it is technically possible to resolve some of these issues using, say, a compass sensor to enable consensus on orientation, technical solutions are sometimes problematic due to driving up the cost, size and complexity of robots, if such a thing is even technically feasible in whatever environment the robots will find themselves. It would therefore be interesting if an algorithmic solution can be found instead, mitigating the issues caused by such adversarial conditions without complicating the robot design.

In this thesis, we explore an extension to the OBLOT model: measurement errors, where a robot may perceive other robots to be in a slightly different position than they actually are. While such sensing inaccuracies have been studied previously by Cohen and Peleg [4], we build upon this work, presenting a more general overview and theoretical framework for measurement errors. We present a simplified variant of their error model. Then, through the presentation of two simple examples, we explore the notion of taking an existing algorithm that functions in an error-free setting, and adapting it such that it still functions in an environment with measurement errors.

Throughout this document, we will be led by the research question: "How do we model measurement errors, what impact to they have on simple Gathering and *Circle Formation* algorithms under SSync, and whether it possible to develop algorithms that solve these problems while being robust against measurement errors."

1.1 Related work

Distributed coordination problems with multiple identical robots have been studied extensively in literature for at least three decades. One of the oldest works available on the topic dates back to 1990, where Sugihara and Suzuki [12] describe methods to form shapes like circles and lines using a large number of distributed mobile agents. Their approach was mainly based on heuristics, and tested out through simulations rather than formal analysis. Later work, such as that by Suzuki and Yamashita [14], takes a more formal approach to forming patterns and shapes, relying on formal analysis and proofs of correctness to ensure a full understanding of methods and results. This work already uses an an earlier version of the OBLOT model, which continues to serve as a de-facto standard for the following decades, until being formalized in a more complete manner by Floccini et al. [7].

In the rest of this section, we will explore some of the literature surrounding oblivious robots within the OBLOT model and its different variants, where we single out two often-mentioned problems: Gathering (and the related problem of Convergence) and circle formation. Finally we conclude with a discussion of existing literature on so-called measurement errors, which are the subject of this thesis.

Gathering The Gathering problem, sometimes referred to as the Point Formation, Rendezvous, Aggregation or the F-point problem, is the problem where a set of n autonomous mobile robots must, within some finite amount of time, come to occupy an arbitrary single point in space. The problem was first formally presented by Suzuki and Yamashita [14], where they introduce the problem as f-point in a

setting where LCM-cycles are synchronized but not all robots are active each round (an SSYNC scheduler, see Section 2.1.5). They show that no oblivious algorithm can exist to solve the problem for $n = 2$ without additional assumptions on the set of robots activated in each round, whereas they do provide an algorithm that can solve the problem for any $n \geq 3$ under any SSYNC scheduler that guarantees that each robot is always eventually activated (such a scheduler is said to be *fair*). The problem has been extensively studied in literature under a variety of different constraints and settings, such as Agmon and Peleg [1] showing that GATHERING is unsolvable in presence of Byzantine faults under SSYNC, but that some Byzantine faults are tolerable under FSYNC, and that a limited number of crash-faults are tolerable under SSYNC. Also, Floccini et al. [9] showed that GATHERING is possible if robots have a limited visibility range as long as they have a compass. A general solution for any $n > 2$ under a fully asynchronous scheduler (ASYNC) is given by Cieliebak et al. [3].

Since the GATHERING problem is not always solvable, such as with $n = 2$ robots, interest is also shown in CONVERGENCE, which is a weaker problem where robots only need to asymptotically converge to a single point. It was formally introduced as C-POINT by Suzuki and Yamashita [14], where it was pointed out that C-POINT is solvable by any algorithm that solves F-POINT (since the problem is strictly weaker), and that the $n = 2$ case can also simply be solved by having robots move to the middle point between their two positions.

In order to build some intuition for how such algorithms work, consider the following argument that moving to the midpoint indeed converges: suppose that two robots r_1, r_2 operate under an SSYNC scheduler and are controlled by an algorithm that makes any active robot move to the midpoint between the two positions. Then, during any round, the scheduler may choose to activate r_1, r_2 or both. If only one is activated, it simply moves to the point halfway between the original two positions, reducing the distance between both. If both are activated, both will move to the midpoint, reducing the distance between the two to 0. In the worst case, the distance between both is at least halved. Hence, as the rounds go on, the distance between r_1 and r_2 decreases exponentially.

Circle formation Another problem frequently-encountered in literature is the one of CIRCLE FORMATION, where all n robots must eventually form the vertices of a regular n-gon somewhere in space. Sugihara and Suzuki [12] approach circle formation from a heuristic, experimental point of view. Their algorithm mostly works, but the resulting formation sometimes approaches a Reuleaux triangle rather than a circle. This work was further improved on by Tanaka [15], where the resulting configuration was a much better approximation of a circle. The CIRCLE FORMATION problem in the semi-synchronous (SSYNC) setting was deterministically solved by Dieudonné et al. [6] for $n \geq 5$, and later on in the fully asynchronous setting by Floccini et al [8]. Surprisingly, despite the apparent simplicity of the problem, the $n = 4$ case was only solved relatively recently by Mamino and Viglietta [11] in the ASYNC setting with non-rigid movement.

Errors and Faults A lot of the work cited here does assume somewhat unrealistic conditions on the model in which robots operate: robots are always detected reliably and consistently, and always move correctly in accordance to the movement algorithm controlling them. While the work is still very useful from a theoretical perspective, these assumptions are unfortunately far from realistic, as any real-world robots will necessarily exhibit at least some degree of error in most aspects, which is likely to worsen as other design constraints such as cost or size come into play.

Some work has been done, however, to study unpredictable faults in pattern formation algorithms. As mentioned previously, there is the work by Agmon and Peleg [1] on crash-and-byzantine faults in the context of gathering. Moreover, Bouzid et al [2] show that convergence can still be achieved in the presence of a bounded number of crash faults or byzantine faults. From another angle, reliability of sensors has been addressed to some extent by Heriban and Tixeuil [10], who consider the possibility of false-positives and false-negatives, where robots are added or removed from the view in an unpredictable manner, possibly with probabilities proportional to distance from the observer.

Measurement errors Of particular interest for this thesis is the problem of *measurement errors*: robots can see the entirety of their environment, but must handle the possibility of the positions of other robots being inaccurate within a known bound. Literature is somewhat sparse on this topic, but Cohen and Peleg [4] present a formal approach to the problem: given a robot r_0 observing a robot r_1, the position of r_1 received by r_0 will be adversarially picked from anywhere within a region, proportional in size to the distance between r_0 and r_1, around the true position of r_1, see the *angle-distance* error model in Section 2.3.2 for more detail. Within this model, they demonstrate certain bounds under which a purpose-built algorithm can achieve CONVERGENCE under an SSYNC scheduler. They also come to the unfortunate conclusion that GATHERING becomes impossible. Later on, Yamamoto et al. [17] present a variant of the error model used in [4] where, within the view of one robot at a particular time, the angular error of all other robots is the same, and prove some tighter bounds within this context.

1.2 Contributions

The contributions of this thesis consist of the following: a discussion about several different error models, as well as some general observations, lemmas and comparisons surrounding these from Section 2.2 onward. The model mainly explored in this thesis, presented in Section 2.3.3, is simplified compared to the one proposed in [4] (discussed in Section 2.3.2), using only one parameter to simplify analysis. After this, we shall explore two related problems, GATHERING and CIRCLE FORMATION. For both, we analyse the problem for three robots under an SSYNC scheduler, explore relevant design possibilities, then take an example of an algorithm that solves the problem in the absence of measurement errors. Using this algorithm as a basis, we then develop a new algorithm that produces results that are a bounded approximation of the target pattern, featuring an error parameter $\epsilon_0 \geq 0$ that can be adjusted based on the expected size of measurement errors. As ϵ_0 goes to zero, the algorithms become exact. We conclude with a discussion on the results of the thesis, draw a list of general conclusions and recommendations, and present potential avenues for scaling and generalization.

2 Preliminaries

In this section, we shall present a few different concepts that are relevant to the rest of the thesis. First, in Section 2.1, we will formally present the OBLOT model as it is found in literature [7]. Then, in Section 2.2, we introduce a formal definition of a *measurement error model* as an extension to the OBLOT model, as well as a few common properties and related concepts. In Section 2.3, we introduce three error models: the *constant-radius disk* error model in Section 2.3.1, the *angle-distance* error model in Section 2.3.2 (which is the one introduced in [4]), and the *distance-proportional disk* error model in Section 2.3.3. For each of the models, we briefly discuss its strengths and weaknesses, and a few useful lemmas are presented. The section concludes with a comparison of the different error models and a recap of the notation that will be used in the remainder of the thesis.

2.1 The OBLOT model

The OBLOT model, as it is commonly found in literature [7], consists of a set R of n autonomous mobile computational entities, referred to as *robots*, which exist and move around in $\mathbb{R}^d$ (we will only consider $\mathbb{R}^2$ in this thesis). Robots are shaped as points without any mass or area (punctiform), do not carry any externally-visible identifying markers (identical), and may disagree on the reference frame in terms of origin, orientation, unit of measure and even chirality (disoriented).

Their movements are directed by some algorithm, identical for each robot, which is a deterministic function that maps from a robot's view of the current configuration to a movement vector. Robots do not have any other sources of information: the only input to the algorithm is the set of relative positions of the other robots in the local (disoriented) reference frame of the given robot at the time of recording. Robots do not communicate directly, neither through message-passing, nor central coordination; robots also do not have any memory of the past. Robots are, however, considered to possess infinite computational power in computing their movement vector during a round.

At a given point in time t, the set of locations of all robots is called the *configuration*, denoted as $C^t \in \mathcal{C}$, where $\mathcal{C}$ is the (usually infinite) set of all possible configurations, called the *configuration space*. A configuration C^t can be considered a function $C^t : R \to \mathbb{R}^d$ which maps each robot to a point in space. For any robot $r_i \in R$, the position of r_i in C^t is therefore $C^t(r_i)$, which we denote c_i^t for brevity. The superscript t may be omitted if only one point in time is considered. We call points that refer to the actual location of robots within a configuration "c-points".

2.1.1 Local reference frames

Note that robots do not necessarily agree on a frame of reference. Configurations, and the corresponding c-points, are expressed in a global reference frame that exists only as a tool of analysis for the algorithm, and is inaccessible to each robot.

As stated previously, robots cannot directly perceive any current configuration C^t because they are *disoriented*: they may disagree on properties of the frame of reference, including the origin, the orientation, unit of measure or even chirality. When expressed in the local frame of reference of a robot $r_i \in R$, we denote such a configuration, referred to as a *local configuration*, as $C_i^{t*} \in \mathcal{C}^*$. Here, $\mathcal{C}^*$ is the set of possible local configurations, given the number of robots.

Correspondingly, if r_i observes another robot $r_j \in R$ in the local configuration C_i^{t*}, we denote the local position of r_j as $c_{j,i}^{t*}$. We refer to such points as c^*-points. Like their counterparts in the global reference frame, the time index t and observer index i can be dropped if these can be inferred from context. See Figure 1 for an example.

An important thing to note there is that, while the local reference frame of a robot can generally be considered unpredictable, the relationship of some global configuration C^t and a local representation thereof C_i^{t*} (for some observer $r_i \in R$) is simply a combination of translation, rotation, mirroring and uniform scaling. Hence,

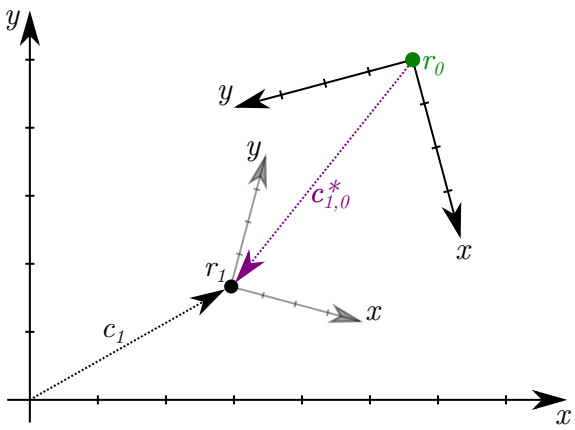


Figure 1: Example of two robots observing each other: r_0 in green on the top right observing r_1 in the bottom left. Both robots feature a local frame of reference, with different orientation and scaling (represented by the axes originating from the robots), with a global frame of reference, known only to an outside observer, depicted by the large axes. In this example, r_0 observes r_1 as the relative vector $c^*_{1,0}$ in its local frame of reference (purple arrow), whereas the position of r_1 can also be expressed in the global frame of reference through the vector c_1.

distance ratios and angles are preserved.

Also, a robot will always observe itself at the origin of the configuration when in the local reference frame. That is, for any observer $r_i \in R$, $c^*_{i,i} = (0, 0)$.

2.1.2 Adversarial context

As can be noted in the previous section, the relative transformation between reference frames is not fixed in advance, nor are many other parts of the model entirely deterministic. Different constraints of different strengths are given, but significant uncertainties will remain.

We approach these unknowns from an *adversarial* point of view. Namely, any solution to a problem expressed in this model must accomplish its' task under all possible different values of any variables or parameters not specified in advance.

From an intuitive point of view, we could see this as an intelligent agent intentionally choosing non-predetermined aspects of the modeling environment in such a way that, if at all possible, the solution to a given problem fails. A solution that succeeds in an adversarial context will thus take into account every possible value of the non-determined variables and parameters of the model.

2.1.3 Robot control algorithm

Within the context of this thesis, we will frequently refer to the following:

Definition 1. A *robot control algorithm* (or *algorithm* for short) is a deterministic function that maps from configurations in the local references frame of robots to movement vectors. This algorithm is identical for all robots, which otherwise do not share memory or other such means of communication.

To be more specific, suppose that all robots are controlled by some algorithm A, and let $r_i \in R$ be an arbitrary robot among them. Every time t where r_i observes the *look* phase in its activation cycle (see Section 2.1.4), it will receive a snapshot of the configuration C^{t*}_i in its local reference frame. (Note: In later sections, S^{t*}_i will designate snapshots that contain measurement errors.)

Then, from that snapshot, it will compute $A(C_i^{t*})$, which will produce a movement vector m_i^* in local space. After compensating for the local orientation of the robot, we obtain the movement vector $m_i \in \mathbb{R}^d$ in the global frame of reference, which we then use in the analysis of how r_i will behave during the subsequent *move* phase.

As a consequence of the memoryless and deterministic nature of such robot control algorithms, such algorithms are generally self-stabilizing: since they do not rely on any preservation of state or memory, most finite transient disturbances made to the configuration, such as a robot being moved around, will eventually result in the problem still being solved. As we will see in Section 4.1, configurations that contain dense points or other types of symmetries may make solving certain problems impossible, implying that disturbances that create such symmetries often cannot be recovered from.

2.1.4 Look-Compute-Move cycles

As time passes, each robot goes through a (possibly infinite) sequence of *activation cycles* (or *cycles* for short) that consist of the following stages:

Sleep The robot is (optionally) immobile some finite amount of time.

Look The robot takes an instantaneous snapshot C^{t*} of its surroundings in its reference frame (see Section 2.1.1), where t is the current time.

As a modification to the standard OBLOT model used in literature, snapshots may also contain adversarial noise in accordance to an error model, in which case we will instead denote them as S^{t*} (see Section 2.2).

Compute The robot is (optionally) immobile for some finite amount of time, during which time the snapshot is retained but not acted upon.

Intuitively, this stage is meant to represent the period of time wherein a physical robot runs the algorithm and computes its movement vector.

Move The robot moves in accordance to the previously-determined movement vector m^* (or simply m if we consider the global reference frame), possibly under the risk of interruption, see Definition 2 about *movement rigidity*.

As noted in Section 2.1.3, the robots are *oblivious*, meaning that no memory is retained in between activation cycles. The only state preserved in between cycles is the global position of robots, to which they have no direct access.

With regards to scheduling of the *Move* cycle, we generally distinguish *rigid* from *non-rigid* movement as follows:

Definition 2. *Movement rigidity* refers to whether a robot is guaranteed to complete the movement vector calculated by its algorithm.

Rigid In each *Move* cycle, the robot completely executes the movement decision made by the algorithm, with the cycle ending only when the target it reached.

Non-rigid In each *Move* cycle, the robot may be deactivated at any time, possibly cutting short the movement decision made by the algorithm. However, the *Move* cycle is assumed to at least last for some positive constant amount of time, in which the robot will move at least some constant distance.

2.1.5 Scheduling

When these cycles occur and how long they take is determined by the *scheduler*, an adversarially-determined activation policy. Some types of schedulers have stronger assumptions than others, but there are three that

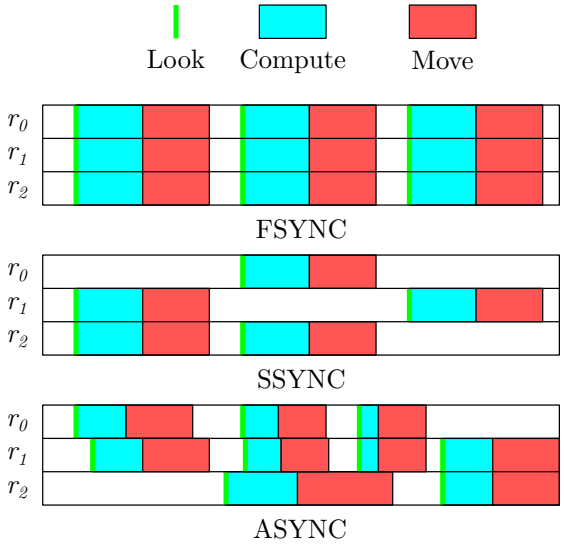


Figure 2: Example of part of an activation schedule for three robots r_0, r_1, r_2 for each scheduler. Time is represented horizontally, with each LCM cycle starting with an instantaneous *Look* phase (green bar), a *Compute* phase in blue, and a *Move* phase in red. Sleep is represented in white. Note the different levels of synchronization in each schedule.

are found somewhat frequently in literature. We define these below, and provide a graphical comparison in Figure 2.

Fully synchronous (FSync) The scheduler considered to have the strongest assumptions of the three. With this scheduler, time is subdivided into distinct regions called *rounds*. During each round, all robots are activated, and the LCM-cycles of all robots are synchronized.

Consequently, robots never observe the others while they are moving, and snapshots cannot be out of date. For all intents and purposes, the robots can be considered as moving instantaneously, albeit that their movements may or may not be interrupted, depending on *movement rigidity* (see Definition 2).

Under this scheduler, time may be considered discrete, with the configurations in the *look* phase of each round being denoted as the sequence $C^0, C^1, C^2 \ldots$

Semi-synchronous (SSync) The SSync scheduler is a weakened version of the FSync scheduler where, in each round, only an adversarially-chosen subset of robots is activated. However, the scheduler is still generally assumed to be *fair*: at any point in time, every robot must always eventually be activated after some finite amount of time.

A scheduler that is not *fair* could be seen as simulating crash faults, where a robot crashes and stops moving indefinitely.

Asynchronous (ASync) Typically considered the scheduler with the weakest assumptions, it will cause the stages of the activation cycle of a robot to occur at unpredictable times and last for durations of unpredictable length. The LCM cycles of the robots are not synchronized, and may even overlap or contain each other between different robots.

Like the SSync scheduler, it is also assumed to be *fair*. Moreover, it is also often assumed that a robot's *move* cycle takes at least some constant amount of time, and may or may not allow a robot to complete its planned movement, depending on *movement rigidity* (see Definition 2).

This scheduler is considered the most adversarial scheduler among those presented here, as robots can observe their environment while others are moving, and because the snapshot will be arbitrarily stale

by the time the robot actually starts moving.

Furthermore, for the purpose of performance analysis, we introduce the notion of *full rounds*:

Definition 3. A *full round* is the smallest span of time in which every robot completes a full activation cycle at least once.

The performance of an algorithm can be expressed as a function of the number of such rounds. In case of the FSYNC scheduler, a *full round* take at most one (simple) round, whereas for the SSYNC scheduler, a single *full round* will contain a non-zero, arbitrary number of rounds since SSYNC only guarantees to eventually activate each robot.

2.2 Modeling measurement errors

In the original OBLOT model, the *snapshots* observed by robots (see Sections 2.1.1 and 2.1.4) are simply views of the current configuration at the time of the *look* phase.

Drawing upon earlier definitions, for the purpose of modeling measurement errors, we modify the OBLOT model by introducing a measurement error function e, which will serve to formally determine the range of options available to the adversary, defined as follows:

Definition 4. An *error function* e is a function from a configuration to a set of possible configurations (snapshots) observed by some robot. The function e is typed as follows:

$$e : R \times \mathcal{C} \to \mathcal{P}(\mathcal{C})$$

where R is the set of all robots, $\mathcal{C}$ is the global configuration space, and $\mathcal{P}$ is the power set function.

Note: the formal error model describes measurement errors in the global reference frame. Given that robots are disoriented, adding noise to the configuration must be done before transforming it to the robot's local reference frame. As we will see in Section 2.3, whether this matters depends on the error model.

In other words, suppose we are given a configuration $C \in \mathcal{C}$ and a robot $r_i \in R$ that observes it. By applying the error function $e(r_i, C)$, we obtain a set of possible *noisy snapshots*: configurations derived from C by adding measurement errors. The adversary picks one of these, denoted S_i, which is then transformed to r_i's local reference frame, yielding S_i^*, and provided as the input for the algorithm. Within this context, the observed location of some robot r_j in S_i is denoted $s_{j,i}$, or $s_{j,i}^*$ for that same position in S_i^*, in the local frame of reference of the observer r_i. Moreover, in the local frame of reference, note that for a robot observing itself, we have that $c_{i,i}^* = s_{i,i}^* = (0, 0)$.

Many different variations of such a function e could exist, consistent with a wide range of possible sensor errors. However, we consider the following to be reasonable assumptions on any such error function e:

- The set returned by e will at least contain a snapshot without noise. That is, through luck, it is possible for sensors to perceive robots where they actually are within the *observer*'s frame of reference. This also meshes with the idea that algorithms can be built with certain safety margins in mind that are often over-estimations of the actual magnitude of errors, and ideally should take advantage of this fact to produce higher-quality output if given the chance.

  Formally, for all observers $r_i \in R$ and configurations $C \in \mathcal{C}^*$, we have that $C \in e(r_i, C)$.

- The set returned by e is continuous in the sense that, if we look at the set of possible positions of a robot $r_j \in R$ across all possible noisy snapshots that can be observed by an observer $r_i \in R$, this set is a continuous region in space, called the *noise region* (see Definition 6) of r_j. In a more intuitive sense, this means that an inaccurate or noisy sensor can perceive a robot anywhere within a given, continuous area around a robot's actual position.

Note: The noise region *is not to be confused with the uncertainty region (Definition 7) of a robot, which is the set of possible actual relative locations of that robot within C^* given its observed position (which includes measurement errors). See Figure 3.*

- To model that some noise is always to be expected, even with the best of sensors, the set returned by e will generally contain infinitely many configurations (as a result of being continuous and containing more than one configuration). Note that some exceptions may occur here, for instance if we consider a measurement error that goes to zero as other robots approach the observer.

Independence of errors between robots As can be noted in Definition 4, an error function is defined as operating on whole local configurations at a time, rather than on individual observed robots. Therefore, we introduce the following:

Definition 5. Measurement errors are considered *independent* if, for a robot $r_i \in R$ observing robots $r_j, r_k \in R$, the error in the observed position of r_j provides no additional constraints on the error in the observed position of r_k. Otherwise, they are *dependent*.

An example of dependent measurement errors can be observed in [17], where the angular error is the same for all robots within a snapshot. Such whole-configuration error models may also be useful in modeling strange effects where one robot's position affects the way another robot is measured. However, measurements errors are usually simply modeled as independent from one another.

Noise region and uncertainty region Mainly for *independent* measurement errors (and perhaps, to a more limited degree, *dependent* measurement errors), it helps to understand the range of measurement errors that can occur given a real-world position of an individual robot and vice-versa, which is why we here define the concept of *noise region* and its dual *uncertainty region*.

Definition 6. With $r_i \in R$, located at some point c_i, observing a robot r_j with known global (note: not local) position c_j, the *noise region* of r_j is the set of all possible locations s_j where r_i may observe s_j after measurement errors are applied.

Definition 7. With $r_i \in R$ observing a robot r_j with measured relative (not global) location s_j^*, the *uncertainty region* of f_j is the set of all possible actual locations c_j^* of r_j in the local frame of reference of r_i may yield such an observed location s_j^*.

It should be noted, here, that the *noise region* is defined in the global coordinate space, because the amount of noise in a measurement will typically be determined by the actual relative positions between robots. The *uncertainty region*, on the other hand, is defined in terms of the local coordinate space of the observer, because it is based on the knowledge of the observer, which is unaware of the global frame of reference. Depending on the error model used, noise regions and uncertainty regions may translate across reference frames.

Fairness of measurement errors and termination Depending on the task at hand, it may be desirable for robots to eventually terminate. Unfortunately, without any further constraints, the adversary that controls measurement errors may make this impossible. Take, for instance, the problem of forming an equilateral triangle. Any algorithm that solves this problem will probably include some way of detecting the existence of an equilateral triangle, and then decide to move or not to move based on the result of that test. Due to measurement errors, the adversary can make any equilateral triangle look like a non-equilateral triangle, and make a range of non-equilateral triangles look exactly equilateral. An initial attempt to solve this problem would be to add a threshold value such that any configuration within that threshold is considered "good enough", and the robots won't move if such a case is detected.

However, consider Figure 4, where an algorithm designer has implemented such a threshold. No matter what threshold they pick, the adversary could place robots near the boundary of the approximation threshold,

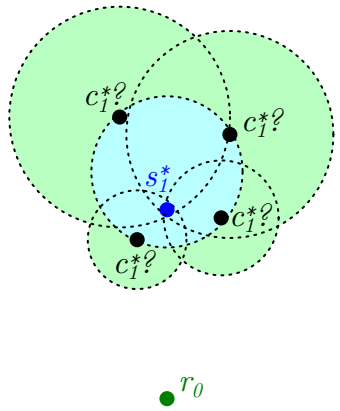


Figure 3: Illustration of the difference between the *noise region* and *uncertainty region*. Robot r_0 observes another robot r_1 at position s_1. The uncertainty region of r_1 is illustrated as the light blue disk, with center u_1. Different possible positions of r_1 with the hypothetical noise region (light green) are displayed in accordance to the *distance-proportional disk* model (see Section 2.3). Notice how s_1 lies in the interior of each of these regions.

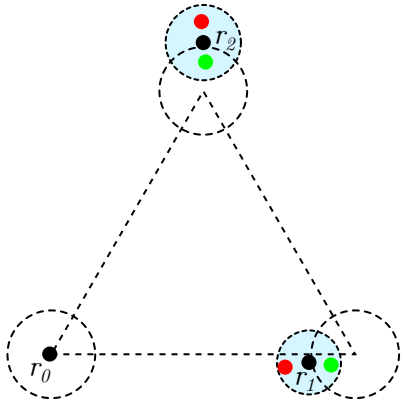


Figure 4: Illustration of the impossibility of termination without some kind of fairness criterion. Robot r_0 observes r_1 and r_2, where the adversary may choose to display the two robots at either the green points, within the threshold boundary, or at the red points, outside the threshold boundary. Hence, the adversary can choose when to make the robots detect certain properties about the configuration. For example, whether the robots form the vertices of an equilateral triangle.

and then simply choose to either display a configuration that does conform to the threshold, or one that does not. Given that the adversary can simply wait indefinitely to reveal the fact that the configuration does not conform to the threshold, termination becomes somewhat ill-defined.

As a result, a certain fairness criterion emerges. An algorithm designer may make any of a number of choices here, either assuming that the measurement errors eventually stop changing, hence guaranteeing that a situation like the one previously described cannot occur, or weakening the success criterion to one where a result is considered "good enough" at some point without entirely guaranteeing eventual stability.

2.3 Examples of error models

Up to this point, we explored a general definition of *measurement errors*. In this section, we will look at three specific examples of error models: the *constant-radius disk* error model, the *angle-distance* error model, and the *distance-proportional disk* error model. For each model we will define it formally through the error function, explore some strengths and weaknesses, and establish a few useful lemmas about them.

2.3.1 Constant-radius disk error model

An intuitive first attempt at modeling measurement errors is what we will refer to as the *constant-radius disk* error model. In this error model, the noise region of every robot, regardless of the observer, is simply a disk of some constant radius $\epsilon_0 \geq 0$ centered at the robot.

Formally, given an observer r_i and a configuration $C \in \mathcal{C}$, the error function is defined as follows:

$$e(C, r_i) = \{\forall r_j \in R : \textbf{ if } j \neq i \textbf{ then } |c_j, s_j| \leq \epsilon_0 \textbf{ else } c_j = s_j \mid S \in \mathcal{C}\},$$

where $|p, q|$ is the Euclidean distance function between two points p and q.

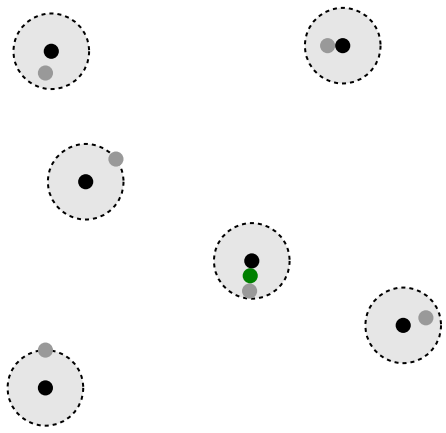


Figure 5: Illustration of the constant-disk error model. The different robots are being observed by the robot in green, with their actual position in black and observed position in light gray, the noise region is also depicted in light gray with black dashed border. Note how the size of the noise region does not change, and that if another robot is sufficiently close to the observer, the observer can even find itself inside the noise region of that other robot.

Now, we follow up with a lemma formalizing the noise region:

Lemma 1. Given a robot $r_i \in R$ observing a robot $r_j \in R$, the *noise region* of r_j is a disk of radius ϵ_0 centered on c_i if $r_i \neq r_j$, otherwise the noise region is a singleton containing c_i.

Proof. The definition of the error function gives the constraint $|c_j, s_j| \leq \epsilon_0$, the locus of which corresponds exactly to the proposed shape of the *noise region*. □

However, there is a major drawback to this model if robots do not agree on a unit of measurement:

Lemma 2. For any robot $r_i \in R$ observing another distinct robot $r_j \in R$, the uncertainty region of r_j is the entire plane.

Proof. Since the size of the radius of the region depends on an absolute length, and robots do not know the global unit of measurement, the size of the noise region can be arbitrarily large in the local frame of reference of the observer. By consequence, there are no meaningful constraints on the actual location of another robot based on where it is observed. □

Consequently, this introduces the problem that a robot can never be quite certain about anything, and has to effectively "go blind"; the *uncertainty region* is effectively useless and cannot be used to make assessments about the current configuration. Analysis can only be done with knowledge of the actual configuration, information that the robot itself does not have.

2.3.2 Angle-distance error model

The *constant-radius disk* error model is simple, and serves as a good starting point, but might be unrealistic if we consider sensors whose accuracy drops as the object being observed is further away. Moreover, given the aforementioned issues with unit of measurement, a measurement error that scales with the distance between robots would be better, since the uncertainty region would be knowable to robots despite non-consensus on unit of measure.

A different error model, more complex, but perhaps more realistic and practically useful, is what we will call the *angle-distance* error model. Parameterized on an angular error parameter $\theta_0 \in [0, \pi]$ and a distance-proportional error parameter $\epsilon_0 \geq 0$, the error function e is defined such that the noise range for any observed robot $r_i \in R$ is a sector of an annulus centered on c_0 with inner radius $(1 - \epsilon_0)|c_0, c_i|$ and outer radius $(1 + \epsilon_0)|c_0, c_i|$, the arc extending by an angle of θ_0 on either side of c_i (total amplitude is 2 θ_0). This model is also found in literature, see [4], with a *dependent* (see Definition 5) variant presented in [17].

More formally, for any observer $r_i \in R$ and configuration $C \in \mathcal{C}$, this is a measurement error model characterized by the following error function:

$$\begin{aligned} e(C, r_i) = \{\forall_{r_j \in R} : \textsc{InRange}(r_j, S) \wedge \textsc{InAngle}(r_j, S) \mid S \in \mathcal{C}\}, \text{where} \\ \textsc{InRange}(r_j, S) = ||c_i, c_j| - |c_i, s_j|| \leq \epsilon_0 \cdot |c_i, c_j|, \\ \textsc{InAngle}(r_j, S) = |\angle(c_j, c_i, s_j)| \leq \theta_0, \end{aligned}$$

where $|p, q|$ is the euclidean distance function between some points p and q and $\angle o, p, q$ is the smallest angle formed by points o, p and q. Here, as mentioned previously, points are given in the global reference frame, and a transformation therefore has to first be applied to understand how the error model works in the local reference frame. This is not very difficult: the error is proportional to the distance between robots, a measure that is meaningful after such a transformation.

Lemma 3. For a robot $r_i \in R$ observing another robot $r_j \in R$ at relative position s_j^*, the *uncertainty region* under the *angle-distance* error model with distance and angle error parameters ϵ_0, θ_0 is a sector of an annulus centered on s_j^* with inner radius $|s_i^*, s_j^*|/(1 - \epsilon_0)$ and outer radius $|s_i^*, s_j^*|/(1 + \epsilon_0)$, the sector extending by an angle of θ_0 on either side.

Proof. For a robot $r_i \in R$ observing itself at position s_i^*, coinciding with c_i^*, another robot $r_j \in R$ at position s_j^*, with c_j^* unknown. The error model provides us with the following constraints:

$$\begin{cases} ||c_i, c_j| - |c_i, s_j|| \leq \epsilon_0 \cdot |c_i, c_j| \\ |\angle(c_j, c_i, s_j)| \leq \theta_0 \end{cases}$$

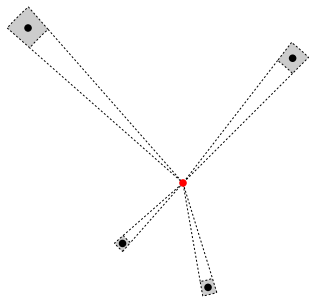


Figure 6: A configuration of robots, with an observer in red, *noise range* of other robots, under *angle-distance* error model, shown in gray area.

where c_i, c_j, s_i, s_j are $c_i^*, c_j^*, s_i^*, s_j^*$ in the global frame of reference.

Taking into account that angles and distance ratios are preserved across local and global reference frames, we can conclude that these constraints translate to the local s^* and c^* points as

$$\begin{cases} \left||c_i^*, c_j^*| - |c_i^*, s_j^*|\right| \leq \epsilon_0 \cdot |c_i^*, c_j^*| \\ |\angle(c_j^*, c_i^*, s_j^*)| \leq \theta_0 \end{cases}.$$

By looking at the two extreme cases, the first constraint can be re-written into the following:

$$\frac{|c_i^*, s_j^*|}{1+\epsilon_0} \leq |c_i^*, c_j^*| \leq \frac{|s_i^*, s_j^*|}{1-\epsilon_0}.$$

The second condition is actually symmetric: the minimal angle between c_j^* and s_j^* (centered on c_i^*) is the same regardless of the order of these two points. Hence, the angular uncertainty is also simply θ_0 on either side of the observed position. The locus of these inequalities is the described region. □

The strengths of this model include that it is far more realistic than the *uniform circle* model presented in Section 2.3.1: the *uniform circle* model, for instance, predicts measurement errors occurring that could cause a robot to perceive another robot 180 degrees in the wrong direction, should it be close enough. This is not the case if the distance error is proportional to the distance between robots, cases with large θ_0 notwithstanding. Secondly, it makes intuitive sense that, as signals scatter and weaken with distance, that sensors become less accurate as distances increase.

However, it does suffer from one weakness, especially at higher values of θ_0: the noise and uncertainty regions are concave. The shape is, moreover, relatively complex and features sharp angles, which likely complicates analysis. See Figure 7 for an extreme example.

2.3.3 Distance-proportional disk error model

With the aim of combining the simplicity of the *constant-radius disk* model and the distance-proportional elements of the *angle-distance* model, we present a third option: the *distance-proportional disk* error model, where the noise range of the each observed robot is a disk centered on the observed robot's actual position, with a radius proportional to the distance between the two robots by a coefficient parameter ϵ_0.

More formally:

Definition 8. The *distance-proportional disk* error model is a measurement error model with an error function e defined such that for some parameter $0 \leq \epsilon_0 \ll 1$ and an observer $r_i \in R$:

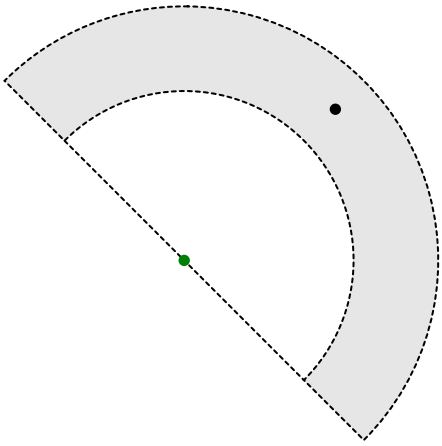


Figure 7: Illustration of the noise range at $\theta_0 = \pi/2$ radians, yielding a highly concave shape. The observer is depicted in green, with the actual position of the observed robot in black. While it may technically be possible for sensors to behave in such a manner, it complicates analysis and would, at least intuitively, appear to be somewhat unrealistic.

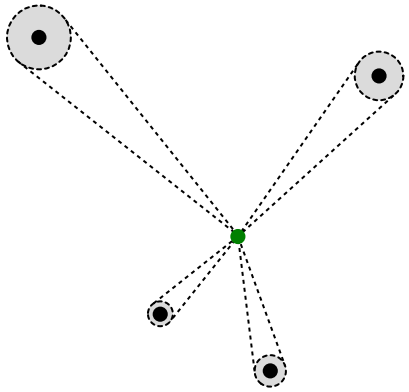


Figure 8: Illustration of the *distance-proportional disk* error model, where the noise range is centered on the position of the other robots as observed by the green robot in the middle, the size of each disk proportional to the distance between the observer and the observed.

$$e(C, r_i) = \left\{ \forall_{r_j \in R} : |s_j, c_j| \leq \epsilon_0 \cdot |c_i, c_j| \mid S \in \mathcal{C} \right\}$$

Next, we follow up with a lemma on the noise region and uncertainty region uncertainty region in this model:

Lemma 4. Given robots $r_i, r_j \in R$, where r_i observes r_j, the noise region of r_j is a disk centered on c_j with radius $\epsilon_0 |c_i, c_j|$.

Proof. Given the definition of the error function, we know that $|s_j, c_j| \leq \epsilon_0 \cdot |c_i, c_j|$, which directly implies the proposed noise region. □

Lemma 5. For a robot $r_i \in R$ observing another robot $r_j \in R$, the *uncertainty region* (see Definition 7) of r_j under the *distance-proportional disk* error model with parameter ϵ_0 is a disk of radius $|s_i^*, s_j^*| \epsilon_0 / (1 - \epsilon_0^2)$ centered on $(s_j^* - s_i^*)/(1 - \epsilon_0^2) + s_i^*$.

(Note: an alternative notation that is more convenient is provided at the end of the proof.)

Proof. Let $r_i, r_j \in R$, where r_i observes r_j, measuring it at position s_j^*. By definition, the *uncertainty region* is the set of possible points c_j^* where r_j can be located (relative to r_i) such that it is perceived to be at s_j^*. From the error model, after translating to the local frame of reference of r_i, for all candidate points c_j^* it holds that

$$|s_j^*, c_j^*| \leq (\epsilon_0 |c_i^*, c_j^*|).$$

From there, we can derive (see Appendix A.1) that

$$|s_j^*, (s_j^* - s_i^*)/(1 - \epsilon_0^2) + s_i^*| \geq \epsilon_0 |c_i^*, s_j^*| / (1 - \epsilon_0^2).$$

In other words: c_j^* exists in a disk of radius $\epsilon_0 |s_i^*, s_j^*| / (1 - \epsilon_0^2)$ centered on $(s_j^* - s_i^*)/(1 - \epsilon_0^2) + s_i^*$. □

For convenience of notation, for a robot $r_i \in R$ observing another robot $r_j \in R$ at position $s_{j,i}^{t*}$, we denote the center of the uncertainty region in r_i's frame of reference as $u_{j,i}^{t*}$, where the superscript t and observer index i are optional if they are implied from context. Formally

$$u_{j,i}^{t*} = \frac{s_{j,i}^{t*} - s_{j,i}^{t*}}{1 - \epsilon_0^2} + s_{i,i}^{t*}$$

where, technically $s_{i,i}^{t*} = (0, 0)$ since we are working in the local frame of reference of r_i.

This also conveniently allows us to re-write the radius of the uncertainty region as:

$$\epsilon_0 |u_{j,i}^{t*}, u_{i,i}^{t*}|$$

where $u_{i,i}^{t*}$ coincides with $c_{i,i}^{t*}$ and $s_{i,i}^{t*}$.

Furthermore, we can also rewrite distance estimates as follows:

Lemma 6. For robots $r_i, r_j \in R$, where r_i observes r_j we have:

$$\frac{|c_i^*, c_j^*|}{1 + \epsilon_0} \leq |u_i^*, u_j^*| \leq \frac{|c_i^*, c_j^*|}{1 - \epsilon_0}$$

and

$$(1 - \epsilon_0) |u_i^*, u_j^*| \leq |c_i^*, c_j^*| \leq |u_i^*, u_j^*| (1 + \epsilon_0).$$

Proof. Follows directly from $|u_i^*, u_j^*| = \frac{|s_i^*, s_j^*|}{1 - \epsilon_0^2}$, and $(1 - \epsilon_0) |c_i^*, c_j^*| \leq |s_i^*, s_j^*| \leq (1 + \epsilon_0) |c_i^*, c_j^*|$. □

Linear combinations and uncertainty Throughout this thesis, comparisons are frequently made using either the positions of robots themselves, the center of gravity (COG) or other kinds of linear combinations. Luckily, the idea of a *noise region* and *uncertainty region* can easily be extended to linear combinations of robots positions.

Definition 9. Suppose that we are given a set of robot positions $c_0, c_1, c_2 \ldots$, and a corresponding set of weights $a_0, a_1, a_2 \ldots$, such that we can construct a linear combination p_C as

$$p_C = \sum_{r_i \in R} a_i c_i \text{ where } a_i \in \mathbb{R}.$$

Then, given an observer $r_i \in R$, the *noise region* of p_C is the set of all points p_S that could result from taking all possible observed (global) positions $s_0, s_1, s_2 \ldots$, through the following construction:

$$p_S = \sum_{r_i \in R} a_i s_i \text{ where } a_i \in \mathbb{R}.$$

The noise region of such a linear combination under the *distance-proportional disk* is quite straightforward:

Lemma 7. Under the *distance-proportional disk* error model with parameter ϵ_0, with a set of n robots R positioned according to some configuration C, observed by some robot $r_0 \in R$ as snapshot S, the *noise region* of a linear combination of $c_0, c_1, c_2 \ldots$ with weights $a_0, a_1, a_2 \ldots$ is a disk with radius $\sum_{r_i \in R} \epsilon_0 a_i |c_i, c_0|$ centered on p_C.

Proof. By the definition of the error model, the noise range of each $r_i \in R$ is a disk of radius $\epsilon_0 |c_i, c_0|$ and centered on c_i.

If every $c_i = s_i$, then the proposed linear combination of the positions of the centers of the uncertainty regions of each robot exactly corresponds to the center of the proposed uncertainty region disk.

Then, as the position of each c_i varies within the noise region for that robot, the position of the linear combination varies proportionally, weighted by a_i. Adding up all of these movements effectively yields a movement of a distance of at most the radius of the proposed noise region. □

Then, we can apply similar logic in reverse to yield the *uncertainty region* of a linear combination of the positions of the robots:

Definition 10. Suppose we are given a set of local observed positions $s_0^*, s_1^*, s_2^* \ldots$, and a corresponding set of weights $a_0, a_1, a_2 \ldots$, such that we can construct a linear combination p_S^* as follows:

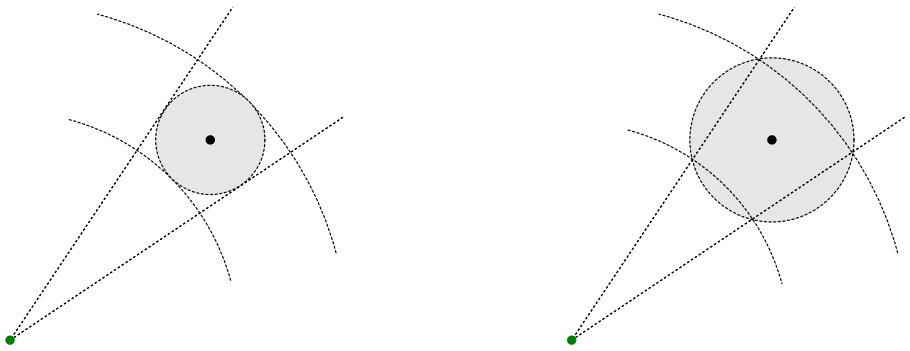
$$p_S^* = \sum_{r_i \in R} a_i s_i^*.$$

Then, the *uncertainty region* of p_S^* is the set of all possible local points p_C^* that, given all possible values of $c_0^*, c_1^*, c_2^* \ldots$ could have been constructed as

$$p_C^* = \sum_{r_i \in R} a_i c_i^* \text{ where } a_i \in \mathbb{R}.$$

Lemma 8. Under the *distance-proportional disk* error model with parameter ϵ_0, a set of n robots R positioned according to some configuration C, observed by some robot $r_0 \in R$ as snapshot S, the *uncertainty region* of a point p_S that is a linear combination of the positions of each robot in S with weights $a_0, a_1, a_2 \ldots$ is a disk with radius $\sum_{r_i \in R} \epsilon_0 a_i |u_i, u_0|$ and centered on: $\sum_{r_i \in R} a_i u_i$

Proof. Lemma 5 states that the uncertainty region for any robot r_i is a disk of radius $\epsilon_0 a_i |u_i, u_0|$ and centered on $a_i(u_i - u_0)$.



(a) Example of a distance-proportional disk noise range fitting inside an angle-distance noise range.

(b) Example of a distance-proportional disk noise range fitting inside an angle-distance noise range.

Figure 9: Comparison of angle-distance and distance-proportional-disk error models.

If every $c_i = u_i$, then the proposed linear combination of the positions of the centers of the uncertainty regions of each robot exactly corresponds to the center of the proposed uncertainty region disk.

Then, as the position of each c_i varies within the uncertainty region for that robot, the position of the linear combination varies proportionally, weighted by a_i. Adding up all of these movements effectively yields the radius of the proposed uncertainty region. □

Comparison to angle-distance model The *distance-proportional disk* error model (Section 2.3.3) and the *angle-distance* error model (Section 2.3.2) are directly comparable.

Theorem 1. If an algorithm solves a problem under the *angle-distance* error model with distance error coefficient ϵ_0 and an angle error parameter θ_0, where $\sin\theta_0 \geq \epsilon_0$, then that same algorithm also solves that problem under the *distance-disk* error model with the same distance error coefficient ϵ_0

Proof. The noise range in the angle-distance model is a sector of an annulus (see Lemma 3): the intersection of the interior area of an angle of amplitude θ_0 and an annulus of inner radius $(1 - \epsilon_0)$ and outer radius $(1 - \epsilon_0)$ times the distance between the observer and the observed robot.

As illustrated in Figures 9a and 10, $\sin\theta_0 \geq \epsilon_0$ guarantees that the noise range disk fits inside the interior area of the angle, whereas the distance uncertainty being the same makes sure that the disk fits inside the annulus.

As a result, the number of possible positions that a robot may be observed in will be a subset of those under the other error model, meaning that the adversary is strictly weaker and the algorithm still works. □

Theorem 2. If an algorithm solves a problem under the *distance-disk* error model with distance error coefficient $\epsilon_0 < 1$, then that same algorithm also solves that problem under the *distance-disk* error model as long as the parameters fit the sector of the annulus inside the noise range disk. See Figure 9a.

Proof. Same as with Theorem 1, if the noise region of a robot under one error model is always contained within the noise region under the other error model, the algorithm for for the larger noise region will work for the smaller since the adversary is weaker. □

The exact relation between the parameters is left as an exercise to the reader.

The fact that the two models are comparable shows that the results from [4] are directly transferable to the *distance-proportional disk* error model. This includes an impossibility result from that paper that states that CONVERGENCE is impossible if $\theta_0 \geq \pi/3$ under the *angle-distance* error model, which effectively becomes:

Theorem 3. Convergence with 3 robots is impossible when $\epsilon_0 \geq \sqrt{3}/2$ under *Ssync* in the *distance-disk* model.

Proof. By Theorem 1, if $\epsilon_0 \geq \sqrt{3}/2$, the error model effectively reduces to the angle-distance model with $\theta_0 \geq \pi/3$ (and undefined ϵ_0 for the distance measurement), see Figure 10. As proven in the work by Cohen and Peleg [4] convergence then becomes impossible. □

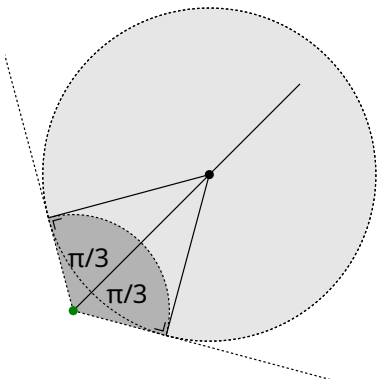


Figure 10: Illustration of Theorem 3. Note how the size of the *noise range* of the observed robot induces an angular uncertainty of $\pi/3$ on either side. Following [4], this makes it possible for the adversary to distort the triangle such that the algorithm can be manipulated into never converging.

2.4 Recap on notation

Throughout this section, a lot of new notation has been introduced. For convenience, it is summarized below.

The set of robots is designated as R, individually represented as r_i where i is an index. Note that a distinct index does not necessarily imply that the robots themselves are distinct. Also, note that while this notation treats robots as distinct, robots themselves cannot tell each other apart by any other means than relative position within their local frame of reference.

C^t: the configuration of the robots at some time t, in the global frame of reference. A configuration is effectively a mapping of robots to positions in $\mathbb{R}^2$. For a robot r_i, c_i^t designates the position of r_i for the current configuration C in the global frame of reference. The superscript t may be omitted if it is implied from context.

C_i^{t*}: the configuration of the robots at some time t, in the local frame of reference of some observing robot $r_i \in R$. For another robot $r_j \in R$, $c_{j,i}^{t*}$ designates the position of r_j in the local frame of reference of r_i. The time index t and observer index i may be omitted if implied from context.

For each notation C^t, C_i^{t*}, c_i^t and $c_{j,i}^{t*}$, we further denote S_i^t, S_i^{t*}, s_i^t and $s_{j,i}^{t*}$ as the corresponding concept after the application of some adversarially-determined measurement errors, relative to some observer $r_i \in R$. The same rules apply for omission of time and observer indices.

Finally, specifically for the *distance-proportional disk* error model, for an observed position $s_{j,i}^t$, $u_{j,i}^t$ (or $u_{j,i}^{*t}$ for the local frame of reference) designates the center of the uncertainty region disk, meaning the center of the region in which r_j may actually be located given only the information available to r_i (see Section 2.3.3). Indices and superscript may be omitted under the same conditions as for s-points.

2.5 Making algorithms robust

In the following sections, we will deal with the notion of "robustness against measurement errors".

Unfortunately, this does not mean that the algorithms, after adaptation, will be able to provide exact solutions to pattern formation problems despite measurement errors. For instance, as discussed in [4, 3], GATHERING is impossible in the context of measurement errors, only CONVERGENCE.

As we will see in Section 4, this expands to any pattern that requires exact relative positions between robots: since robots are always uncertain about exact relative positions, they are always uncertain about whether they have to move and where to form the requested pattern.

However, there exists a common practice in programming, when dealing with floating point arithmetic to simply add a small margin of error to equality comparisons, and while the practice is usually applied somewhat haphazardly, the result is often "good enough" to the point that the average user does not really notice. This approach would seem to offer a way out for dealing with measurement errors.

The inspiration of this thesis, however, is a case where this approach unfortunately failed: a group of students attempted to implement a pattern formation algorithm in a simulator. Likely due to errors with regard to determining the position of other robots and floating point arithmetic, the robots exhibited erratic behavior that resisted attempts at resolution.

While the process is therefore apparently not entirely straightforward, there is still value in being able to adapt an existing pattern formation algorithm that yields a bounded approximation of the desired pattern.

3 Gathering and convergence

GATHERING, by the definition used within this thesis, is the problem where a set of robots R in the OBLOT model must eventually reach a configuration where all robots occupy a single point. The robots must then remain in that configuration indefinitely.

A related problem is that of CONVERGENCE. Strictly weaker than GATHERING, it requires that robots asymptotically converge towards a single point. This can be formalized by requiring, for instance, that the radius of the smallest enclosing circle of the robots must always eventually decrease over time.

As mentioned in *related work* (Section 1.1), extensive literature is available on GATHERING and CONVERGENCE, ranging over different numbers of robots, schedulers, guarantees of movement rigidity, and models. However, with regards to measurement errors, only two contributions are known at the time of writing: [4, 17]. Both of these use a custom-built algorithm that achieves CONVERGENCE, while concluding that GATHERING is impossible. For instance, [4] provides a solution to CONVERGENCE under an SSYNC scheduler and the *angle-distance* error model (see Section 2.3.2) under the condition that $0.2 > \sqrt{2(1-\varepsilon_0)(1-\cos\theta_0+\epsilon_0^2)}$ where ϵ_0 and θ_0 are the maximum angle and distance error parameters, respectively. Wider ranges on these parameters become possible when additional constraints are placed on the model, such as in making errors *dependent* [17] (see Definition 5).

As a side-result of this thesis we observe the following. Table 1 in [17] states that the case where $\epsilon_0 \geq 1$ is an open problem. However, with such a large value for ϵ_0, we argue that it is possible for the distance error to be so large that other robots will appear exactly on top of the observing robot. Hence, the adversary can make any configuration look like any other, and just about any problem becomes unsolvable.

In the following sections, we will build upon this work by revisiting the idea of GATHERING being impossible under presence of measurement errors. Obviously, we do not invalidate the results from literature, but we instead explore the idea of taking an existing GATHERING algorithm and adapting it to function in the presence of measurement errors. The new algorithm shall provide a similar operating principle and performance characteristics to the original algorithm, with results that are a bounded approximation of the original result.

Specifically, in order to familiarize ourselves with the problem and build an intuition for it while avoiding overly-complex analysis, we will take a simple GATHERING algorithm for three robots under SSYNC with rigid movement, and adapt it such that it works under the *distance-proportional disk* error model. These ideas are formalized in Section 3.3.

3.1 Design space analysis

There are many possible choices when it comes to designing an algorithm that gathers a set of three robots to a single point. Recall that any such algorithm would be a deterministic function that, for each robot $r \in R$, receives a snapshot of the configuration in its local (disoriented) reference frame and produces a movement vector. This nature, the given goal, and the significant adversarial behavior on part of the scheduler restrict the design space for such an algorithm to a surprising degree.

We will explore some of these design restrictions below in case of an SSYNC scheduler with rigid movement, drawing inspiration from [13, 14], and detailing some of the common characteristics that such algorithms are going to exhibit under different classes of configurations in an error-free environment. Then, in later sections, we will explore how the addition of measurement errors impacts such an algorithm.

Among all different possible configurations, we distinguish a few different possible types, and argue how GATHERING could be achieved from there:

Robots form an equilateral triangle One feature to be noted here, is that such a system exhibits 3-way rotational symmetry around the center of gravity, which coincides with the center of the smallest enclosing circle of the three robots. As a result, the adversary can disorient the robots such that they all perceive an identical snapshot of their environment, and the same movement vector (in their local

reference frame) is thus produced by the algorithm for all three robots. If the scheduler adversary then activates all three robots at the same time, they will thus all move by the same vector, relative to their orientation. If this movement vector is any other vector than one that makes the robot move to the center of gravity, thus achieving gathering, the robots will form another equilateral triangle, again with all robots possibly perceiving exactly the same snapshot. Thus, if the algorithm does anything other than move the robots to the COG, care must be taken that the adversary cannot simply repeat this behavior indefinitely. Especially with regards to the fact that robots can have arbitrary unit length, it may be difficult if not impossible to distinguish one equilateral triangle from another, so it is not clear at the time of writing if any option other than moving to the COG is even possible.

Robots form an isosceles triangle Here, the configuration exhibits mirror symmetry along the bisector of the angle formed at the apex of the isosceles triangle. Also, one robot is clearly distinct, namely the one at the apex, while the other two robots, the ones at the base of the triangle, are indistinguishable from each-other due to non-consensus about chirality. Any algorithm has a choice here: move the robot at the apex, move any of the robots at the base, or all three.

On the one hand, the robot at the apex seems a natural choice: it is uniquely distinguishable, implying that under any fair scheduler, it is possible for the algorithm to eventually move this and only this robot. Should an algorithm decide to move this robot, it can make a number of choices:

- It can move along the bisector of the apex. This either maintains the isosceles triangle while changing the height, or hits special cases such as the robots either becoming co-linear or forming an equilateral triangle.

  It must be noted that forming an equilateral triangle may be problematic, as the scheduler can choose to activate just one robot on the next round, have it move to the COG, and re-form an isosceles triangle. This could cause an infinite loop.
- It can also move away from the bisector of the apex. This either forms a scalene triangle, a line, or the robot moves exactly to the position of one of the robots at the base of the triangle, creating a dense point where two robots occupy the same position.

On the other hand, it is also possible to move the robots at the base of the triangle. The simplest option is to instruct the robots to move to the apex position. If only one robot is active, a dense point with 2 robots is formed. In case both robots are activated, both move to the apex, and gathering is achieved this way. If the robots are instructed to move anywhere else, the scheduler may decide to activate either one robot or both in a way that is likely non-trivial to analyse.

Robots form a scalene triangle In configurations where the robots form the vertices of a scalene triangle, since angles are of a different size and have a unique order regardless of reference frame, robots are easy to distinguish from one another.

A very simple solution to the gathering problem, here, is to simply instruct up to two robots to move to the position of the third robot, either gathering immediately, or forming a dense point with two robots, from which a known gathering strategy exists. Depending on movement rigidity, care must be taken not to deal with the potential formation of non-scalene configurations.

Robots form a line Depending on whether the robot that is in between the two others is centered or not, this situation can simply be considered a degenerate case of the two previous ones. However, noting that robots do not agree on orientation or chirality, such a line exhibits symmetries (such as mirror symmetry across the line) which make that attempts to move away from the line may introduce complications.

A dense point exists Should a dense point exist, most strategies will generally make the third robot join the other two, completing the gathering after that round. The algorithm cannot really do anything else, as moving the robots in the dense point can, depending on a scheduler's activation policy, always cause that dense point to be destroyed. Since, in any round where no two robots occupy the same position, the scheduler may choose to only activate 1 robot, gathering can never be achieved in one round from such a configuration. Hence, an algorithm that breaks a dense point can be driven into an infinite loop.

Obviously, it is hard to impossible to lay out and analyse the full design space for a gathering algorithm. However, barring more complex algorithms that may rely on inter-round conservation of reference frames, angle amplitude and distance thresholds, which would be more complex to analyse, the aforementioned limitations and opportunities should still paint a general picture of what an algorithm designer must face and will be able to exploit.

3.2 Example of a gathering algorithm for three robots under SSync

Following the design considerations from the previous section, and drawing inspiration from [13, 14], it is possible to construct a very simple algorithm that gathers a set of 3 robots under an SSync scheduler and rigid movement. This results in Algorithm 1.

Input: Positions c_1^*, c_2^* in the local frame of reference of robot r_0.
Output: Movement vector in the local frame of reference of robot r_0.

$c_0^* \leftarrow (0,0)$
$c_{COG}^* \leftarrow (c_0^* + c_1^* + c_2^*)/3$.
Let $c_{\min}^*, c_{\text{med}}^*, c_{\max}^*$ be c_0^*, c_1^*, c_2^* sorted by distance to c_{COG}^*.
if $|c_{\max}^*, c_{COG}^*| = |c_{\min}^*, c_{COG}^*|$ **then**
    Move to c_{COG}^*.
else
    if $|c_0^*, c_{COG}^*| > |c_{\min}^*, c_{COG}^*|$ **then**
        Move to $c_{\min}^*$.
    else
        Do not move.
    end if
end if

Algorithm 1: Gathering under SSync for $n = 3$ without errors. This algorithm is applied between the *Look* and *Move* phases of the activation cycle of each robot, with inputs being the relative positions of the other two robots, denoted r_1, r_2, in the local reference frame of the observer, denoted r_0. Note that this assignment of indices is subjective to each observer.

Simply put: if an observing robot (subjectively denoting itself r_0, and the other two as r_1, r_2) sees that all robots are equidistant from the COG, then r_0 will attempt to move to the COG. Otherwise, if r_0 is further than some robot $r_{\min}$ closest to the COG, then r_0 will move to the position of $r_{\min}$. If either of the previous two cases do not apply, r_0 does not move.

Firstly, one might notice that Algorithm 1 is expressed in terms of distances to the COG instead of equilateral/isosceles triangles. However, these notions are equivalent. For instance, for equilateral triangles:

Lemma 9. Three points $a, b, c \in \mathbb{R}^2$ are equidistant from the $COG = (a + b + c)/3$ if and only if they form an equilateral triangle.

Proof. Given that the COG is the intersection of the medians of the triangle, the COG being equidistant from points a, b, c, and one point of each median is the midpoint of one of the edges of the triangle, the medians must lie completely on the perpendicular bisector of each edge. Since a, b, c each also lie on one of the medians, this implies that each vertex is equidistant from the two others. □

Moreover, we draw an equivalence between the distance to the COG and the lengths of the sides of the triangle:

Lemma 10. Let $a, b, c \in \mathbb{R}^2$ form a triangle, and $COG = (a + b + c)/3$. If $|a, COG| < |b, COG|$, then $|a, c| < |b, c|$ and the angle at a is larger than b.

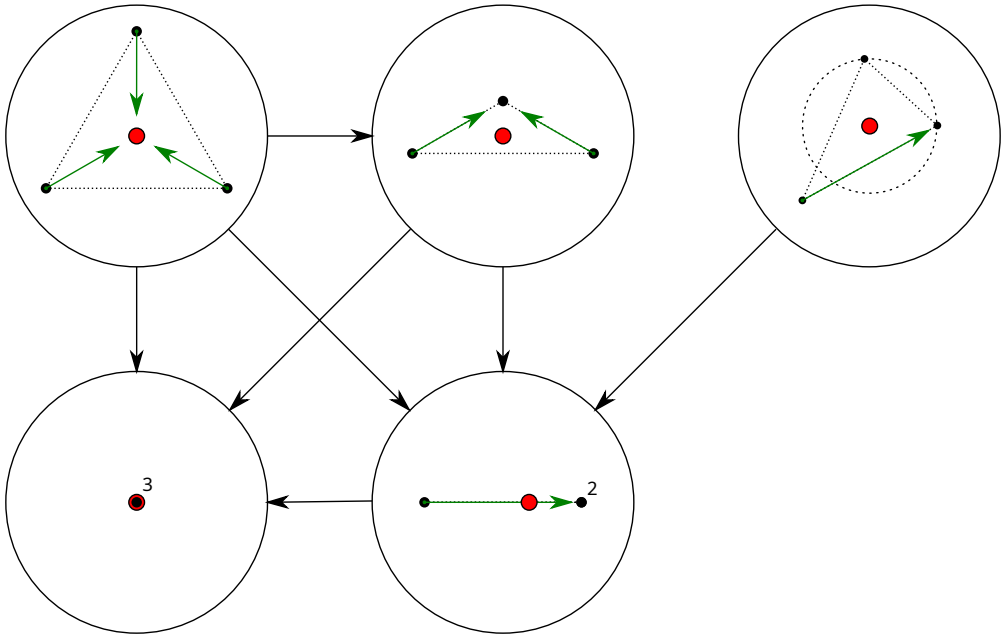


Figure 11: State machine built from configuration classes illustrating the proof of correctness of Algorithm 1, with algorithm movement instructions indicated as green arrows and COG indicated with red disk with black border. For dense points, a number indicates how many robots are at that point. Each green arrow indicates a way that the robot at the base of the arrow can move, though note that since we are under an SSync scheduler, robots will not move if they are inactive during a round. For top row: equidistant from COG (by Lemma 9, equilateral), one robot is closer with other two equidistant, other triangles. Bottom row: gathered state, dense point.

Proof. Let m be the median point between a and b. Since $|a, COG| < |b, COG|$, the amplitude of the angle formed by a, m, COG is smaller than the angle formed by b, m, COG. Since c is on the same median containing points m and the COG, we can argue that this difference in amplitudes also applies to angles a, m, c and b, m, c. Hence, it must hold that $|b, c| > |a, c|$. The property of the angle follows from the relationship between angle size and opposite edge length. □

Due to disorientation, robots do not have access to the global frame of reference, and Algorithm 1 is therefore expressed in the local reference frame of some observer r_0. However, we prefer to perform analysis in the global reference frame, where decisions by different robots are comparable. Fortunately, this is not difficult because:

Lemma 11. Algorithm 1 is equivalent to an algorithm expressed using the global positions of each robot: c_0, c_1, c_2.

Proof. Local points c_0^*, c_1^*, c_2^* are simply the points c_0, c_1, c_2 transformed to the local reference frame of the currently-active robot r_0. This transformation is a combination of uniform scaling, mirroring, rotation and translation. The resulting movement instructions are then transformed into the global reference frame.

Indeed, the algorithm would be equivalent because it makes decisions based on linear comparisons of distances between points, comparisons which are preserved through such transformations. □

Now, we can proceed with the actual proof of correctness, using c-points and c^*-points almost interchangeably:

Theorem 4. Algorithm 1 gathers 3 robots under an SSYNC scheduler and rigid movement in $O(1)$ full rounds (see Definition 3).

Proof. This proof is illustrated in Figure 11. We then distinguish different cases that can occur in the configuration.

Robots are in a gathered configuration. If the robots are gathered, the COG will coincide with their shared position. Any active robot will thus observe $|c^*_{\max}, c^*_{COG}| = |c^*_{\min}, c^*_{COG}|$ and move to the COG, which is equivalent to not moving. Hence, the robots stay gathered as they should.

A dense point exists. If a dense point already exists, the fact that two robots are in the same point will mean that the COG is closer to those than to the non-gathered robot. Hence, the latter robot moves to the other two robots (which do not move), achieving gathering in the first round in which the non-gathered robot is active.

Not all robots are equidistant to the COG and no dense point exists. There are now two kinds of robots: robots that are at the minimum distance to the COG, and robots that are further from the COG than at least one robot. Since there are three robots, there will be between one and two in each category.

Eventually, one of the robots not closest to the COG will move to one of the robots that is closest to the COG. Since robots at the minimum distance to the COG will not move, a dense point will be formed. Per the previous case, we then achieve gathering in at most one full round. If two robots are further from the COG than the minimum distance, and both are active, both will move to the closer point and gather in one round.

All robots are equidistant from the COG. If all robots are equidistant from the COG, depending on the number activated, one of the following eventually occurs: the "equidistant" condition is broken with one or two robots moving to the COG, forming a dense point in the latter case (and then gathering in one full round), or all three move to the COG, achieving gathering immediately.

In every case, the robots gather within $O(1)$ full rounds. □

3.3 Adding uncertainty

The possibility of measurement errors, however, complicates matters significantly. Re-using the labeling from Algorithm 1, a robot $r_0 \in R$ can no longer access the exact (relative) positions c^*_1, c^*_2, and must instead work with positions s^*_1, s^*_2 that include sensor noise, where for some $\epsilon_0 \geq 0$, the noise regions are such that

$$|s^*_1, c^*_1| \leq \epsilon_0 |c^*_0, c^*_1| \quad \text{and} \quad |s^*_2, c^*_2| \leq \epsilon_0 |c^*_0, c^*_2|.$$

Note: distance ratios are preserved across frames of reference, so these relations will persist if the superscript $*$ is dropped.

For ease of analysis, we denote the position of the observer as s^*_0, which is simply $(0, 0)$ in the local frame of reference. Furthermore, for each robot $r_i \in R$, we use the centers of the uncertainty region u^*_i of each robot which, as established by Lemma 8, can be computed as

$$u^*_i = (s^*_i - s^*_0)/(1 - \epsilon_0^2) + s^*_0.$$

The COG is then also translated to these types of points, with

$$s^*_{COG} = (s^*_0 + s^*_1 + s^*_2)/3 \quad \text{and} \quad u^*_{COG} = (u^*_0 + u^*_1 + u^*_2)/3.$$

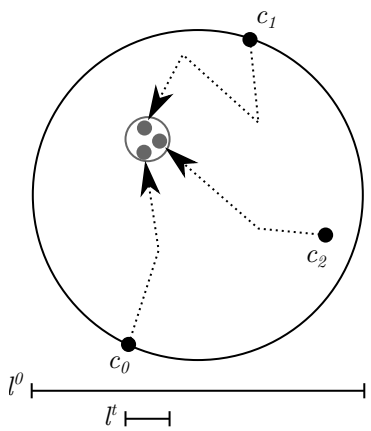


Figure 12: Illustration of Approximate Gathering, where robots eventually move to within a small disk of diameter bounded by d_t, which is a small fraction of the initial diameter of the SEC d_0.

Approximate results A critical consequence of measurement errors, unless robots are already gathered in exactly one place, is that sensor noise makes it impossible for robots to come to an agreement about any one point in space. Hence, it is impossible for two robots that have any distance between them to move to a common point, including the position of one of the robots. As already noted in [4], Gathering is generally unsolvable when measurement errors are present.

Instead, we propose the following:

Definition 11. For a set of n robots starting in a configuration C^0, we define the Approximate Gathering problem as the problem where all robots must eventually move within a disk of a diameter $d_t \leq \varepsilon d_0$, where d_t is the diameter of the smallest enclosing circle (SEC) of the positions of all robots in some future configuration C^t, and $\varepsilon \in [0, 1)$ is some value proportional to ϵ_0.

This is illustrated in Figure 12.

As may be guessed from the name, the Approximate Gathering problem is simply the Gathering problem, but where robots being located in a small disk of size relative to ϵ_0 and d_0 is considered a suitable approximation of actually gathering to a single location. Notably, Gathering is a special case of Approximate Gathering with $\epsilon_0 = \varepsilon = 0$.

Indeed, Approximate Gathering can be seen as a specific case of Convergence: due to the memoryless nature of the robots, once a single iteration of Approximate Gathering has been solved, the robots will simply solve the problem recursively again at a much smaller scale. Hence, the distance between robots will go down exponentially over time at a rate of $O\left((d_t/d_0)^m\right)$, where m is the number of iterations of Approximate Gathering.

Distance from the COG and thresholds Algorithm 1 frequently makes use of distances of the form $|c_i^*, c_{COG}^*|$, and assigns labels $\max, \mathrm{med}, \min$ to each robot based on their ordering according to that distance metric. Since the actual position of each robot is now unavailable, we will instead use the centers of uncertainty regions to compare distances, with expressions of the form $|u_i^*, u_{COG}^*|$; see Figure 13 for how these distances relate geometrically. Obviously, due to measurement errors, exact distance comparisons using such expressions will not be reliable, it is entirely possible that the distances being compared are so similar that the adversary can change the order of the distances perceived by a robot.

In this context, the following lemma will prove useful:

Lemma 12. Let $r_a, r_b \in R$ be two robots observed by some robot $r_0 \in R$ (not necessarily distinct from r_a, r_b, and let r_1, r_2 be the other two robots distinct from r_0), such that the centers of the uncertainty regions

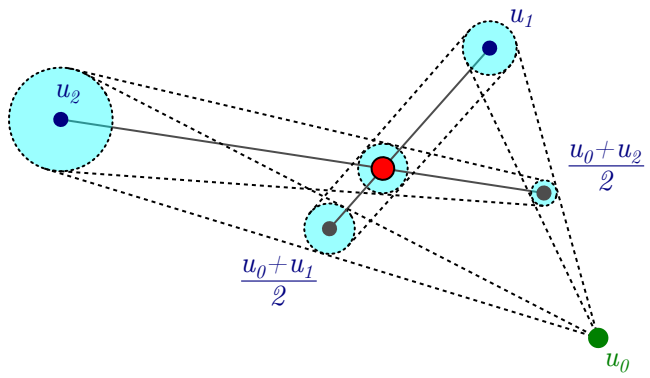


Figure 13: Illustration of how the uncertainty regions of r_1 and r_2 relate to the uncertainty region of the COG, where the COG is located in the intersection of the medians of the triangle formed by the three robots. While the interactions can be somewhat complex, note that the COG will move by a third of the movement of any of the points, hence why the threshold δ_1 is sufficient.

of r_a, r_b are u_a^*, u_b^*, respectively, and let $u_{COG}^* = (u_0^* + u_1^* + u_2^*)/3$ be the center of the uncertainty region of the COG.

Then,

$$|u_a^*, u_{COG}^*| > |u_b^*, u_{COG}^*| + \delta_1 \quad \text{where} \quad \delta_1 = \frac{4\epsilon_0}{3}(|u_0^*, u_1^*| + |u_0^*, u_2^*|)$$

implies

$$|c_a^*, c_{COG}^*| > |c_b^*, c_{COG}^*|.$$

Proof. We can break down the distances as the length of the vector between the points:

$$|u_a^*, u_{COG}^*| = |u_a^* - \frac{u_0^* + u_1^* + u_2^*}{3}| \quad \text{and} \quad |u_b^*, u_{COG}^*| = |u_b^* - \frac{u_0^* + u_1^* + u_2^*}{3}|.$$

From these equalities, we note that the displacement of the observed locations of the robot relative to their active location contributes by a third to the observed location of the COG. Hence, following the radii of the uncertainty regions according to Lemma Lemmas 5, we can state that $|u_a^*, u_{COG}^*|$ and $|u_b^*, u_{COG}^*|$ will each differ by at most $\frac{2\epsilon_0}{3}(|u_0^*, u_1^*| + |u_0^*, u_2^*|)$ from their c^*-counterparts $|c_a^*, c_{COG}^*|$ and $|c_b^*, c_{COG}^*|$. □

Intuitively, we can interpret this lemma to mean: "While we generally cannot be sure that any particular inequality will hold in the actual configuration, we can reason that if two distances differ by a sufficient threshold, we can at least know which of the distances is actually larger." Obviously, it then also follows that if the two distances differ by less than the given threshold, the given inequality may not hold in the actual configuration.

Assignment of distance order labels The first actual decision made by Algorithm 1 is where it assigns labels $\min, \text{med}, \max$ to the three robots in order of their distance to the COG. Instead, we will now order them using their u^*-points, such that:

$$|u_{\min}^*, u_{COG}^*| \leq |u_{\text{med}}^*, u_{COG}^*| \leq |u_{\max}^*, u_{COG}^*|.$$

However, as noted previously, the adversary may choose measurement errors such that this order is inconsistent between observers: one robot may assign these labels differently than another robot! While we will

still use these labels, we must be careful with them, especially when the time comes to choose which robots must move and, if so, where to.

Luckily, if a robot mis-assigns labels, these labels are still useful when it comes to reasoning about maxima and minima in the distance of a robot to the COG:

Lemma 13. Let R be a set of 3 robots, observed by a robot $r_0 \in R$ as snapshot S_0^*, under the *distance-proportional disk* error model with parameter ϵ_0, where C_0^* is unknown.

Then:

$$\max_{r_i \in R} |u_i^*, u_{COG}^*| \leq \max_{r_i \in R} |c_i^*, c_{COG}^*| + \delta_1$$
$$\max_{r_i \in R} |u_i^*, u_{COG}^*| + \delta_1 \geq \max_{r_i \in R} |c_i^*, c_{COG}^*|$$
$$\min_{r_i \in R} |u_i^*, u_{COG}^*| \leq \min_{r_i \in R} |c_i^*, c_{COG}^*| + \delta_1$$
$$\min_{r_i \in R} |u_i^*, u_{COG}^*| + \delta_1 \geq \min_{r_i \in R} |c_i^*, c_{COG}^*|$$

where δ_1 is the threshold value defined in Lemma 12.

That is, by looking at the robot where the center of the uncertainty region is furthest from (or closest to) the center of the uncertainty region of the COG, we can say something useful about the maximum distance between a robot and the COG in the actual configuration.

Proof. Let $r_{\max}$ be a robot $r_i \in R$ that maximizes $|u_i^*, u_{COG}^*|$. By Lemma 8, the radius of the uncertainty region of the COG is $\frac{\epsilon_0}{3}(|u_0^*, u_{COG}^*|)$, where $u_0^* = (0, 0)$ since we are working in the local frame of reference of the observer r_0. By Lemma 5 the uncertainty region of $u_{\max}^*$ has radius $\epsilon_0 |u_0^*, u_{\max}^*|$.

Now, suppose for a contradiction:

$$\begin{aligned}
&|u_{\max}^*, u_{COG}^*| > \max_{r_i \in R} |c_i^*, c_{COG}^*| + \delta_1 \\
\Leftrightarrow\ &|u_{\max}^*, u_{COG}^*| > \max_{r_i \in R} (|c_i^*, c_{COG}^*| + \delta_1) \\
\Leftrightarrow\ &|u_{\max}^*, u_{COG}^*| > \max_{r_i \in R} |u_i^*, u_{COG}^*| \\
\Leftrightarrow\ &\text{False}
\end{aligned}$$

Hence, we conclude that the negation of the hypothesis is true:

$$|u_{\max}^*, u_{COG}^*| \leq \max_{r_i \in R} |c_i^*, c_{COG}^*| + \epsilon_0 (|u_0^*, u_{\max}^*| + |u_0^*, u_{COG}^*|).$$

The other claims follow by symmetry. □

Equidistant from the COG The first branch of Algorithm 1, guarded by the expression $|c_{\max}^*, c_{COG}^*| = |c_{\min}^*, c_{COG}^*|$, covers the case where all robots are equidistant from the COG, also coinciding with the case where robots form an equilateral triangle (see Lemma 9). As explained in Section 3.1, it is very important to treat this case, as it presents a symmetry that must be broken by exploiting the possible choices of the adversary in which robots to activate during a round. If a *false-negative* is possible due to measurement errors, the adversary can exploit this to cause the system to end up in an infinite loop.

Hence, drawing upon Lemma 12, we must ensure that r_0 will move if there is even the *possibility* that $|c_{\max}^*, c_{COG}^*| = |c_{\min}^*, c_{COG}^*|$, formalized by the condition that

$$|u_{\max}^*, u_{COG}^*| \leq |u_{\min}^*, u_{COG}^*| + \delta_1.$$

Note: Algorithm 2 fulfills this requirement implicitly.

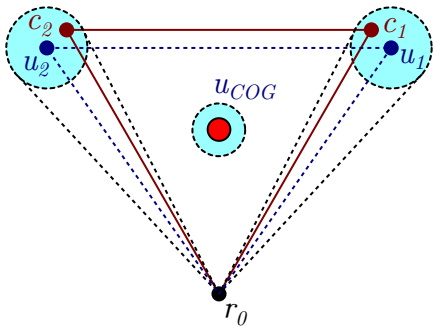


Figure 14: Example illustrating a case where two robots, while not observed as equidistant from the COG, given their uncertainty regions, could in reality actually be forming an equilateral triangle.

As a result, we know for sure that r_0 will detect the case there is no other robot either certainly further from or closer to the COG than r_0, covering the case where robots are equidistant to the COG, but it may also recognize a number of other cases that approach such a situation. Therefore, it will unavoidably be vulnerable to *false positives*, see Figure 14.

Preserving the behavior of Algorithm 1, we will still have the robot move to where it perceives the COG. More precisely, r_0 will move to s_{COG}, which Lemma 7 tells us is a point at most distance $\epsilon_0|c_0, c_{COG}|$ away from the actual COG.

So far, we have therefore distinguished two cases:

- The active robot believes that all robots may be equidistant to the COG. This corresponds to the **if** clause on line 4 in Algorithm 1.
- The active robot knows that robots cannot all be equidistant to the COG. This corresponds to the **else** clause on line 6 of the same algorithm.

Further from the COG In Algorithm 1, once we are certain that robots are not equidistant to the COG, we find another check, with the guard $|c_0^*, c_{COG}^*| > |c_{\min}^*, c_{COG}^*|$. Robots where this check passes will move to a robot that is at the minimum distance to the COG. The latter robot must therefore, obviously, not move.

As a result, rewriting the check as $|u_0^*, u_{COG}^*| > |u_{\min}^*, u_{COG}^*|$ is not good enough. Firstly because, again, it is possible that this ordering of distances does not translate to the actual configuration, something we could remedy by adding a threshold as follows: $|u_0^*, u_{COG}^*| > |u_{\min}^*, u_{COG}^*| + \delta_1$.

However, this is not quite sufficient because, whereas r_0 can know for sure when it is actually further from the COG than $r_{\min}$, the robot $r_{\min}$ may be active and may have received a different snapshot than r_0. Consequently, $r_{\min}$ may falsely believe that it is possible for the robots to be forming an equilateral triangle, and decide to move to the measured position of COG. However, this is not guaranteed. Hence, before r_0 can move to $r_{\min}$, it must ensure that $r_{\min}$ will not suffer from such a false-positive. For this purpose, the following lemma is introduced:

Lemma 14. Consider a set of three robots R positioned according to a configuration C. Let $r_0, r_1 \in R$ be two robots observing C according to snapshots S_0, S_1, respectively. Furthermore, consider two robots $r_a, r_b \in R$ (not necessarily distinct from r_0, r_1) positioned such that, from the perspective of r_0, $u_{a,0}^*$ is closer to u_{COG}^* than $u_{b,0}^*$ by a sufficiently wide margin, through the inequality

$$|u^*_{a,0}, u^*_{COG,0}| > |u^*_{b,0}, u^*_{COG,0}| + \delta_2 \quad \text{where} \quad \delta_2 = (\frac{4\epsilon_0}{3} + \frac{16\epsilon_0(1+\epsilon_0)}{3(1-\epsilon_0)})(|u^*_{0,0}, u^*_{1,0}| + |u^*_{0,0}, u^*_{2,0}|).$$

Then it follows that, from the perspective of r_1, it is certain that r_a is further away from the COG than r_b. That is, the following inequality holds

$$|u^*_{a,1}, u^*_{COG,1}| > |u^*_{b,1}, u^*_{COG,1}| + \frac{4\epsilon_0}{3}(|u^*_{1,1}, u^*_{0,1}| + |u^*_{1,1}, u^*_{2,1}|).$$

where $\frac{4\epsilon_0}{3}(|u^*_{1,1}, u^*_{0,1}| + |u^*_{1,1}, u^*_{2,1}|)$ is δ_1 from the perspective of r_1. Also, note that this will translate to the local frame of reference of r_1 since this transformation preserves distance ratios and inequalities.

Proof. The derivation is available in Appendix A.2. □

The proposal for the case guard for moving to $r_{\min}$ is thus as follows:

$$\textbf{if } |u^*_0, u^*_{COG,0}| > |u^*_{\min}, u^*_{COG,0}| + \delta_2 \textbf{ then}.$$

Ambiguity However, this now leaves a gap: what if r_0 knows it is further from the COG than $r_{\min}$, but there is still some ambiguity left as to whether $r_{\min}$ knows this, and that $r_{\min}$ has not been mislabeled. This ambiguity is illustrated in Figure 15.

More formally, the case where an active robot observes the following proposition as true:

$$|u^*_{\min}, u^*_{COG}| + \delta_1 < |u^*_0, u^*_{COG}| \leq |u^*_{\min}, u^*_{COG}| + \delta_2.$$

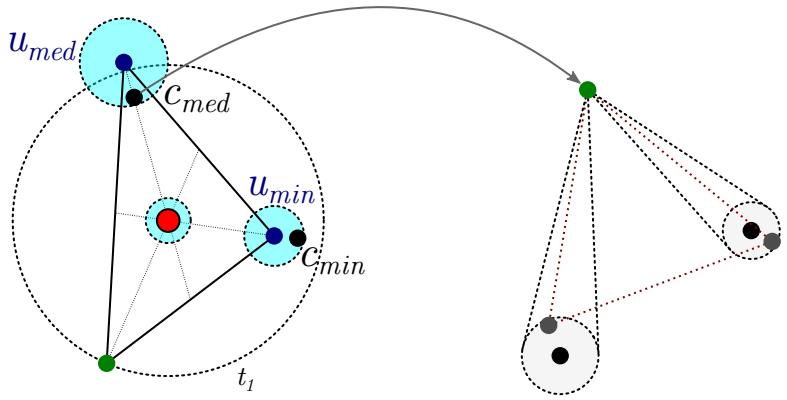


Figure 15: Illustration of how non-consensus can exist in the configuration. Given the uncertainty regions as observed by r_0, it is not possible for the robots to be arranged in an equilateral configuration. However, it *is* possible for robots to be arranged such that, when observed by the robot at the top left, the configuration looks as if it could, in fact, be equilateral.

Unfortunately, this ambiguity turned out to be highly-complex, and too difficult to resolve without changing the behavior of the robots. This was not for lack of trying: weeks were spent on a solution that would preserve the robots' behavior, but this came at the cost of a severe increase in algorithm complexity, which eventually still lead to situations involving deadlocks or other undesirable behavior which resisted attempts at resolving them. One such failed attempt is documented in Appendix B.

Instead, we shall simply accept somewhat arbitrary behavior on part of one of the robots, as long as this still eventually leads to approximate gathering. See the proof of Theorem 5 for the full detail.

All things put together, we obtain Algorithm 2.

Input: Positions s_1^*, s_2^* in local frame of reference of robot r_0.
Output: Movement vector in the local frame of reference of robot r_0.

1: $s_0^* \leftarrow (0,0)$
2: $u_0^* \leftarrow s_0^*$
3: $u_1^* \leftarrow (s_1^* - s_0^*)/(1-\epsilon_0^2) + s_0^*$
4: $u_2^* \leftarrow (s_2^* - s_0^*)/(1-\epsilon_0^2) + s_0^*$
5: $\delta_1 \leftarrow 4\epsilon_0/3(|u_0^*, u_1^*| + |u_0^*, u_2^*|)$
6: $\delta_2 \leftarrow \left(\frac{4\epsilon_0}{3} + \frac{16\epsilon_0(1+\epsilon_0)}{3(1-\epsilon_0)}\right)(|u_0^*, u_1^*| + |u_0^*, u_2^*|)$
7: $s_{COG}^* \leftarrow (s_0^* + s_1^* + s_2^*)/3$
8: $u_{COG}^* \leftarrow (u_0^* + u_1^* + u_2^*)/3$
9: Let $s_{\min}^*, s_{\text{med}}^*, s_{\max}^*$ be s_0^*, s_1^*, s_2^* sorted by distance to s_{COG}^*.
10: Let $u_{\min}^*, u_{\text{med}}^*, u_{\max}^*$ be u_0^*, u_1^*, u_2^* sorted by distance to u_{COG}^*.
11: **if** $|u_{\min}^*, u_{COG}^*| + \delta_2 < |u_0^*, u_{COG}^*|$ **then**
12: move to $s_{\min}^*$
13: **else if** $|u_{\max}^*, u_{COG}^*| > |u_0^*, u_{COG}^*| + \delta_1$ **then**
14: Stay in place.
15: **else**
16: Move to s_{COG}^*.
17: **end if**

Algorithm 2: Algorithm for Approximate Gathering under SSync for $n = 3$. Notice the structural similarities to Algorithm 1, and that points c_0^*, s_0^* and u_0^* are all simply $(0,0)$, though we will keep the separate notations to make explanations easier. Also, note that the distance order of the s^*-points and u^*-points is the same, since they are simply scalings centered on (0,0). Parameter ϵ_0 is simply the error parameter of the *distance-disk* error model.

Three cases After formulating the algorithm from the perspective of a particular robot, and explaining the role of different thresholds and guards, it is time now to analyse how all of these robots interact with one another.

Note: From this point onward, we shall use the designation r_0, r_1, r_2 as subjective each robot, in accordance to Algorithm 2, and instead use indices r_i, r_j, r_k when talking about arbitrary robots in the formation. Hence, r_0 is not a globally-consistent designation, and rather refers to the specific label used in Algorithm 2.

First, we summarize the different cases that can be observed by some robot $r_i \in R$ during a round, each corresponding to a branch of Algorithm 2:

1. **Consensus Further** Robot r_i observes that $|u_{\min}^*, u_{COG}^*| + \delta_2 < |u_0^*, u_{COG}^*|$ (line 11).

   That is, r_i is further from the COG than the robot closest to the COG, and by Lemma 14, all other robots will observe at least $|u_{\min}^*, u_{COG}^*| + \delta_2 < |u_0^*, u_{COG}^*|$. Thus, r_i will move to $s_{\min}^*$, which is the position of whichever robot it perceives as closest to the COG.

2. **Other Further** Robot r_i observes that $|u_{\max}^*, u_{COG}^*| > |u_0^*, u_{COG}^*| + \delta_1$ (line 13).

   In more intuitive terms: there is another robot $r_{\max}$ further from the COG than r_i, but other robots might not agree. Therefore, r_i chooses not to move in case $r_{\max}$ decides to move to form a dense point with r_i.

3. **Maybe Equilateral** Robot r_i observes the negation of the previous two cases: $|u_{\max}^*, u_{COG}^*| - \delta_1 \leq |u_0^*, u_{COG}^*| \leq |u_{\min}^*, u_{COG}^*| + \delta_2$ (line 15).

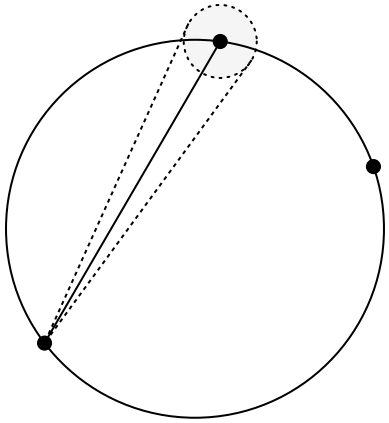


Figure 16: Illustration of Lemma 16, with three robots and the smallest enclosing circle displayed. Notice how each robot is at most the diameter of the SEC apart, and that thus the maximum radius of the noise region is thus proportional to the diameter of the SEC.

Intuitively: r_i is neither further out from the COG than another robot by such a margin that every other robot would agree, nor is there another robot certainly closer to the COG than r_i. Therefore, r_i must move to the COG to avoid the endless loop described in Section 3.1.

In the rest of this proof, each round will be characterized by which robots will observe which of the previous three cases at the same time. The proof of Theorem 5 will cover each of these rounds, but due to the rather complex nature of some of the cases, a few of them are separated out from the main proof as lemmas.

Local and global reference frames The next step in proving the correctness of Algorithm 2 is to make sure that the comparisons made in the algorithm are actually comparable across different robots' frames of reference:

Lemma 15. Algorithm 2 is equivalent to an algorithm on points in the global reference frame.

Proof. As with Algorithm 1 and Lemma 11, Algorithm 2 is expressed in terms of points and linear combinations thereof that do not rely on any particular orientation or origin. Hence, transformations of the input points across frames of reference do not effect the behavior of the algorithm, provided that the resulting movement vector is transformed accordingly as well. □

As a consequence, we will now reason in terms of s-points rather than their local s^*-points counterparts, since the algorithm behaves the same regardless. Of course, we must still take into account that different robots may receive different s-points, since noise is not uniform across robots.

Diameter of the smallest enclosing circle To further ease our reasoning, we adopt the usage of the diameter of the smallest enclosing circle (SEC) to express different lemmas in order to simplify the proof of correctness of the algorithm. Given a configuration C^t, we denote the diameter of the SEC for that configuration as d_t, where the time index can be omitted if implied from context. Moreover, let d_0 be the diameter of the SEC of the initial configuration C^0.

We immediately establish a simple lemma about noise regions in terms of the SEC:

Lemma 16. Expressed in terms of the diameter d of the SEC of the actual robot positions c_0, c_1, c_2, the radius of the noise range of each robot, regardless of which robot is picked as the observer, is at most $\epsilon_0 d$. See Figure 16.

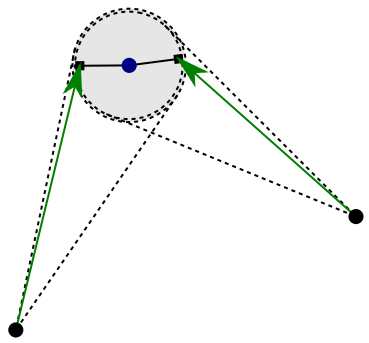


Figure 17: Two robots attempting to move to some third point constructed from the locations of the robots. Due to measurement errors, robots may end up diametrically opposed relative to the target position.

Proof. As shown in Figure 16, the maximum distance between any pair of robots is at most a chord of the SEC, and hence d is an upper bound of that. □

Approximate dense points In the proof of Theorem 4, the notion of a *dense point* is used to distinguish some cases. If we wish to use this proof as a starting point for our proof, we must provide an counterpart for it within a context that features measurement errors: an *approximate dense point* or ADP.

Such a notion would have to fulfill the following criteria:

1. The robots within the dense point must be sufficiently close together with respect to d_t such that they can reasonably be considered to be approximately in the same place, an approximation that goes to zero as ϵ_0 does.
2. Any robot within the dense point must, preferably, be immobilized, while any robot outside the dense point must eventually move to one of the robots within it.

By Lemma 16, if two robots attempt to move to some common point, be it the position of another robot or the COG, that point will have a noise region of radius $\epsilon_0 d_t$. Hence, as shown in Figure 17, both robots will end up within distance $2\epsilon_0 d_t$ from each other, which will serve as the diameter of our dense point.

Unfortunately, the second point is more difficult. Since robots are memoryless, we cannot easily distinguish between a situation where two robots have recently moved together from one where robots happen to be close together.

Instead, we shall argue that, once robots have moved to an approximate dense point, any subsequent movements they make are small in size, and will not exceed a given radius. Moreover, the third robot will either move to form something that can reasonably be called "approximately gathered", or will already be positioned to fulfill this idea.

First, we must reason about what happens if one of the robots within the dense point observes the case *consensus further*, and hence decides to move to the position of another robot. Here, the primary concern is that by over-shooting their destination, the pair of robots can effectively begin to "walk" away from their initial position by activating and over-shooting in alternation. Luckily, this movement is bounded:

Lemma 17. Given $r_i, r_j \in R$, suppose that c_i, c_j are within some distance l from each other. Now, suppose that, for an arbitrary number of rounds, r_i will move to r_j, or vice-versa. Then, r_i and r_j will not exit a

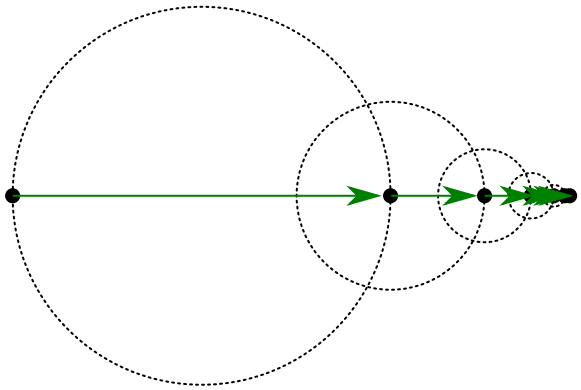


Figure 18: Illustration of the behavior described in Lemma 17 with $\epsilon_0 = 1/4$. Note how, despite the "walking" behavior caused by repeated overshooting, the movements do converge to a limit.

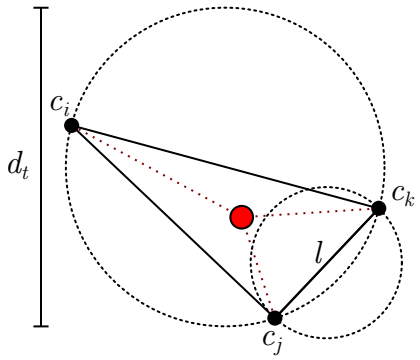


Figure 19: Illustration of Lemma 20. If r_i does not move to r_j or r_k, we can still guarantee that the diameter of the SEC is bounded to at most $|c_i, c_{COG}| + \max\{|c_j, c_{COG}|, |c_k, c_{COG}|\}$, which is in turn bounded to some function of l.

disk of diameter

$$l(1 + 2\frac{\epsilon_0(1+\epsilon_0)}{1-\epsilon_0})$$

centered on the median point between c_i, c_j.

Proof. Whenever one robot moves to the position of the other, it will arrive in a disk of radius $\epsilon_0|c_i, c_j|$. The largest movement will occur if r_i and r_j are activated in alternation, one moving to the position of the other and overshooting as much as possible, as shown in Figure 18. If we sum up the distances traveled, starting from the initial overshoot, we obtain a sum of a geometric series that converges to $\frac{\epsilon_0(1+\epsilon_0)l}{1-\epsilon_0}$. □

Next, we argue about how to describe the situation if one of the robots within the approximate dense point (ADP) decides to move to the COG, a point that is likely to be outside the bounds of the ADP. We argue that, if this happens, the diameter of the SEC is bounded.

Lemma 18. Given robots $r_i, r_j, r_k \in R$, if r_k is not furthest from the COG, then the SEC has a diameter of at most $2|c_i, c_j|$.

Proof. Without loss of generality, assume that r_i is furthest from the COG. From there, we know that the edge opposite r_i (formed by r_j and r_k) must be the shortest in the triangle. Thus, a path consisting of two chords of the SEC exists from c_i to c_k of length at most $2|c_i, c_j|$. The diameter is at most that length. □

Lemma 19. Given two robots $r_i, r_j \in R$, where r_i observes case *maybe equilateral*, the SEC will have diameter at most

$$\frac{16\epsilon_0 + 2}{1 - 8\epsilon_0}|c_i, c_j|.$$

Proof. Let r_k be the third robot, distinct from r_i, r_j. If r_k is not the furthest from the COG, the SEC already has a diameter of at most $2|c_i, c_j|$. Otherwise, if r_i observes *maybe equilateral*, then it must be further from the COG than r_j, but r_k must be close enough to the COG that r_j observes $|u^*_{\max}, u^*_{COG}| \leq |u^*_0, u^*_{COG}| + \delta_1$. From there, we can derive (see Appendix A.4) that $\frac{8\epsilon_0+1}{1-8\epsilon_0}|c_i, c_j| \geq |c_{COG}, c_k|$. By consequence, the SEC will have a diameter of at most $\frac{16\epsilon_0+2}{1-8\epsilon_0}|c_i, c_j|$. □

We apply similar reasoning in case the third robot, not explicitly part of the approximate dense point, decides to do anything other than move to one of the robots within the dense point:

Lemma 20. With robots $r_i, r_j, r_k \in R$, if r_i does not observe case *consensus further*, then the SEC will have a diameter of at most

$$\frac{6(1 - \epsilon_0)^2}{3 - 198\epsilon_0 - 61\epsilon_0^2}|c_j, c_k|.$$

Proof. Denote $l = |c_j, c_k|$. We begin by assuming that r_i is the robot furthest from the COG, since otherwise the SEC will have a diameter of at most $2l$, which is less than the proposed maximum diameter. Furthermore, without loss of generality, assume that r_j is second-furthest from the COG.

If r_k does not observe case *Consensus Further* after the ADP has been formed, we know that it observes

$$|u^*_{\min}, u^*_{COG}| + \delta_2 \geq |u^*_0, u^*_{COG}|.$$

After adjusting for the uncertainty regions defined in Lemma 5, we find that

$$|c^*_{\min}, c^*_{COG}| + \delta_1 + \delta_2 \geq |c^*_0, c^*_{COG}|,$$

which furthermore translates to, at worst,

$$|c^*_{\min}, c^*_{COG}| + t(|c^*_0, c^*_1| + |c^*_0, c^*_2|) \geq |c^*_0, c^*_{COG}|$$

where $t = \frac{8\epsilon_0}{3(1-\epsilon_0)} + \frac{16\epsilon_0(1+\epsilon_0)}{3(1-\epsilon_0)^2}$.

We then derive:

$$\begin{aligned}
|c_i, c_{COG}| &\leq |c_j, c_{COG}| + t(|c_i, c_j| + |c_i, c_k|) \\
|c_i, c_{COG}| &\leq |c_j, c_{COG}| + 2td_t \\
|c_i, c_{COG}| &\leq |c_j, c_k|/2 + |(c_j + c_k)/2, c_{COG}| + 2td_t \\
|c_i, c_{COG}| &\leq |c_j, c_k|/2 + |c_i, c_{COG}|/2 + 2td_t \\
|c_i, c_{COG}|/2 &\leq l/2 + 2td_t \\
|c_i, c_{COG}| &\leq l + 4td_t \\
d_t &\leq 2l + 8td_t \\
(1 - 8t)d_t &\leq 2l \\
d_t &\leq \frac{2}{1 - 8t}l \\
d_t &\leq \frac{6(1 - \epsilon_0)^2}{3 - 198\epsilon_0 - 61\epsilon_0^2}l.
\end{aligned}$$

□

The value of Lemmas 17, 19 and 20 is that, if any robot has any doubt about which robot is further from the COG after the approximate dense point has been formed, it is automatically implied that the diameter of the SEC is bounded relative to the distance between the two robots that form the dense point. We finally combine those lemmas to characterize the behavior of robots once two robots r_i and r_j could be guaranteed to be at most some distance $l \leq \epsilon_0 d_t$ apart, where d_t is the diameter of the smallest enclosing circle.

Lemma 21. Let robots $r_i, r_j, r_k \in R$ be positioned such that r_i, r_j form an approximate dense point (ADP), such that $|c_i, c_j| \leq l$ for some distance $l \leq 2\epsilon_0 d_t$, where d_t is the diameter of the smallest enclosing circle.

Then, if operating under Algorithm 2, we eventually obtain a SEC of diameter at most

$$\frac{6(1-\epsilon_0)^2}{3-198\epsilon_0-61\epsilon_0^2}l.$$

Proof. First, suppose that r_k might not observe *consensus further* when activated. Then, by Lemma 20,

$$d_t \leq \frac{6(1-\epsilon_0)^2}{3-198\epsilon_0-61\epsilon_0^2}l.$$

Similarly, if r_i or r_j observe *maybe equilateral*, then Lemma 19 guarantees that

$$d_t \leq \frac{16\epsilon_0+2}{1-8\epsilon_0}l.$$

Otherwise, we may assume that any active robots observe either *consensus further* (CF), and move to another robot that observes *other further*, or observe *other further* (OF) themselves, and stay in place. We distinguish a few cases based on this:

Robot r_k observes CF, others immobile Robot r_k will move to the measured position of r_j or r_k, creating a configuration with a SEC of $\epsilon_0 d_t + \epsilon_0 l$.

Robot r_k and one of r_i, r_j observe CF, other immobile The two robots will move to the position of the third, which observes OF and stays in place. They thus gather in a disk of radius $\epsilon_0 d_t + \epsilon_0 l$.

A robot in the ADP observes CF, others immobile The ADP is preserved, now with r_i, r_j even closer together, though the SEC may increase in diameter, yielding that $d_{t+1} \leq d_t + \epsilon_0 l$.

By iterating the argumentation so far, we can deduce that either we obtain one of the two maximum diameters $\frac{6(1-\epsilon_0)^2}{3-198\epsilon_0-61\epsilon_0^2}l$ and $\frac{16\epsilon_0+2}{1-8\epsilon_0}l$, or r_k finally moves under CF, and by Lemmas 16 and 17, a SEC is formed of diameter at most $\epsilon_0(1+\frac{\epsilon_0(1+\epsilon_0}{1-\epsilon_0})d_t + \epsilon_0 l$.

At most two robots may observe CF if active, since the robot closest to the COG cannot observe itself as further out than some other robot. Hence, since r_k is assumed to observe CF if active, at most one of r_i, r_k may also do so during any round.

As we will see in a later Lemma 22, we already have the constraint that $\epsilon_0 < 0.00384$, we can simply conclude that, if at any point, two robots are at most some distance l apart, we eventually obtain a SEC of diameter at most

$$\frac{6(1-\epsilon_0)^2}{3-198\epsilon_0-61\epsilon_0^2}l.$$

□

Lemma 22. Given robots $r_i, r_j, r_k \in R$, suppose that r_i observes case *maybe equilateral*, while r_j and r_k either remain in place. Then as long as $\epsilon_0 < 0.00384$, r_j and r_k can only observe case *consensus further* on subsequent rounds until one moves.

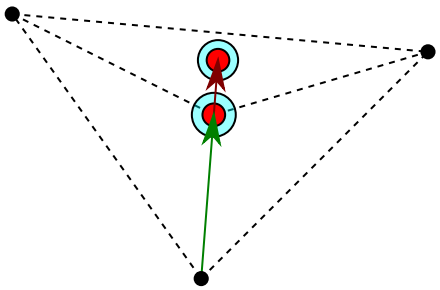


Figure 20: Illustration of Lemma 22. Notice how the movement "flattens" the triangle, guaranteeing that the non-moving robots will clearly observe case *consensus further* on the next round, if active.

Proof. First, consider the case where r_i moves alone. Since r_i observes *maybe equilateral*, it must observe that $|u_{\max}, u_{COG}| - \delta_1 \leq |u_0, u_{COG}| \leq |u_{\min}, u_{COG}| + \delta_1$. From there, we derive (see Appendix A.3) that the configuration was sufficiently close to an equilateral triangle, and the noise range of the COG sufficiently small, that the threshold of δ_2 is always obtained for r_j and r_k if $\epsilon_0 < 0.00384$ (rounded down). □

Now, after all of these lemmas, it is time to combine them to prove that Approximate Gathering has been solved for the non-ambiguous cases.

Theorem 5. From any starting configuration C^0 robots will gather within a disk of diameter

$$\epsilon_0(1+\epsilon_0)\frac{12(1-\epsilon_0)^2}{3-198\epsilon_0-61\epsilon_0^2}d_0.$$

within $O(1)$ full rounds (not to be confused with rounds from the SSync scheduler, see Definition 3) as long as $\epsilon_0 < 0.00384$.

Proof. We will denote cases *Consensus Further* as CF, *Maybe Other Further* as OF, and *Maybe Equilateral* as ME. We shall distinguish nine different cases, each characterized by which of the three cases each robot can observe if active, noting that order does not matter.

First, we treat the case where every robot is active:

CF - CF - CF This case is impossible. Indeed, by Lemmas 14 and 12, the condition $|u^*_{\min}, u^*_{COG}| + \delta_2 < |u^*_0, u^*_{COG}|$ implies that the observer is strictly further from the COG in the actual configuration than at least one robot. Hence, this cannot be true for all three robots simultaneously.

CF - CF - OF Approximate gathering is achieved after the current round. Indeed, the two robots that observe CF will move to the robot that observes OF, the latter remaining in place. As a consequence of Lemma 16, the SEC will have a diameter of at most $2d_t\epsilon_0$.

CF - CF - ME This case is impossible. Consider robots $r_i, r_j \in R$ where r_i is closest to the center of gravity, and r_j is the furthest from the COG. Then, only r_i can observe ME, since Lemmas 14 and 12 guarantee that any robot observing CF is further from the COG than at least one robot. Thus, r_j observes CF. However, r_j observing $|u^*_{\min,j}, u^*_{COG,j}| + \delta_{2,j} < |u^*_{0,j}, u^*_{COG,j}|$ implies, by Lemma 14, that r_i must observe $|u^*_{\min,i}, u^*_{COG,i}| + \delta_{1,i} < |u^*_{0,i}, u^*_{COG,i}|$, contradicting that r_i observes case ME.

CF - ME - ME Similarly to the previous case, this case is impossible.

CF - OF - OF An approximate dense point will be formed, with the diameter of the SEC of the new configuration expanding by a factor of at most $(1+\epsilon_0)$. If one robot r_i observes CF, and the other two

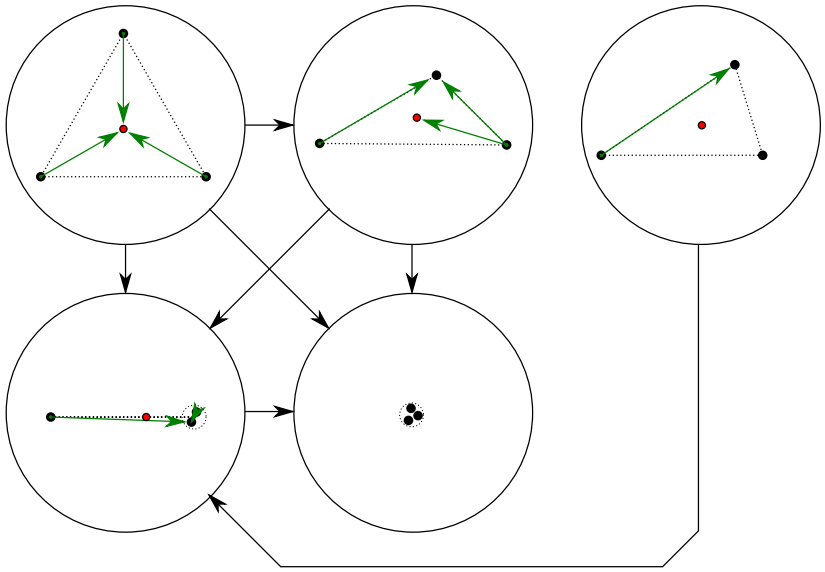


Figure 21: State machine diagram of the different possible types of configurations, split by different combinations of which of *consensus further*, *other further* and *maybe equilateral* can be observed by each robot. The diagram is more or less the same as Figure 11, but note that the dense point and gathered configuration now are small clusters rather than actual shared points. Also, note that the erratic behavior described in Lemma 21 is not fully represented.

robots r_j, r_k observe OF, then r_i will move to one of r_j, r_k, while the latter remain in place. Hence, through Lemma 16, there will be two robots within distance $\epsilon_0 d_t$ from one another. Moreover, the robot which moved will exit the original SEC by at most that same distance.

CF - ME - OF Same as before: An approximate dense point will be formed, with the diameter of the SEC of the new configuration expanding by a factor of at most $(1 + \epsilon_0)$. Indeed, this case is the same as the previous one, since the formation of the ADP does not depend on the movements of the robot that observes case ME.

OF - OF - OF This case is impossible. Indeed, a robot furthest from the COG cannot evaluate the proposition $|u^*_{\max}, u^*_{COG}| > |u^*_0, u^*_{COG}| + \delta_1$ as true, since Lemma 12 would imply that there must be some other robot even further from the COG.

OF - OF - ME One robot will move to the measured center of gravity, while the other two remain in place. By Lemma 22, both robots that remained in place can only observe CF on any subsequent rounds until a robot moves. Depending on which robots are activated on subsequent rounds, approximate gathering is eventually achieved through cases CF - CF - OF or CF - OF - OF since we treat OF as equivalent to being inactive.

OF - ME - ME An approximate dense point will be formed, with the diameter of the SEC of the new configuration expanding by a factor of at most $(1 + \epsilon_0)$. The robot observing OF will remain in place, while the other two robots will move to the measured position of the COG. By Lemmas 7 and 16, both robots will move within a disk of diameter $2\epsilon_0 d_t$ centered on the COG. In the very worst case, this disk will extend beyond the original SEC by at most its radius.

ME - ME - ME All robots will move to the measured position of the COG. By Lemmas 7 and 16, all three robots will move within a disk of diameter $2\epsilon_0 d_t$.

Finally, if not every robot is guaranteed to be active, we can simply treat each inactive robot as if it was observing case OF. We ignore cases where no robot is active, since the SSync scheduler is guaranteed to always eventually activate each robot, and case OF-OF-OF is impossible.

Once a pair of robots r_i, r_j are guaranteed to form an approximate dense point (ADP), being located within some distance $l \leq 2\epsilon_0 d_t$ from each other, we argue through Lemma 21, the robots will eventually gather within a disk of diameter

$$\epsilon_0(1+\epsilon_0)\frac{12(1-\epsilon_0)^2}{3-198\epsilon_0-61\epsilon_0^2}d_0.$$

□

3.4 Discussion and conclusion

In this section of the thesis, we explored the simple problem of Gathering of 3 robots under an SSync scheduler with rigid movement. The problem was introduced, and the available design space and pitfalls of it in the error-free setting was explored. We then presented a concrete solution to the problem in the error-free setting in the form of Algorithm 1.

Then, we introduced measurement errors under the distance-proportional error model, and went over Algorithm 1 in detail, explaining exactly how it breaks as a result of these measurement errors. We introduced the concept of thresholds, and how different thresholds can establish certainty about the actual configuration (Lemma 12) or certainty in other robots about the actual configuration (Lemma 14).

We then showed how these concepts could be applied, resulting in Algorithm 2. As proven in Theorem 5, this new algorithm indeed solves the Approximate gathering problem, moving all robots within a disk that is a fraction of the size of the original smallest-possible disk that could enclose all the robots. At this time, we do not see any reason that would prevent the algorithm from scaling to higher n, but the surprising complexity of the problem and time constraints prevented exploration of this direction of research.

Unfortunately, the behavior of the robots does deviate somewhat strongly from that of Algorithm 1, allowing one robot to move to the measured position of the COG while another moves to the position of a third robot. An attempt was made to resolve this, the result of which can be seen in Appendix B, though it unfortunately proved unsuccessful. There does not appear to be any clear and consistent way to preserve robot behavior at this time.

4 Circle formation

Within the broader scope of pattern formation, CIRCLE FORMATION, is another classic, well-known and well-studied problem.

Specifically, the CIRCLE FORMATION problem poses the requirement that a set of n robots R move to eventually occupy positions relative to each other in order to form the vertices of a regular n-gon, a formation that will increasingly approach a circle as n increases. This formation must then be maintained indefinitely.

As mentioned in Section 1.1, it has been extensively studied under different constraints and circumstances, with new discoveries surprisingly still being made in what would seem rather basic circumstances, such as the case with $n = 4$ under ASYNC by Mamino and Viglietta [16].

Unlike GATHERING and CONVERGENCE however, at the time of writing, there does not appear to be any literature available on CIRCLE FORMATION and the convergence to that pattern specifically within the context of measurement errors.

Within Section 4, we shall explore the CIRCLE FORMATION problem further, specifically within the simple context of an SSYNC scheduler and $n = 3$ robots with rigid movement: the goal is thus to form an equilateral triangle. We shall explore the available design space in depth for this particular setting in Section 4.1, and provide an example of a simple algorithm that solves the problem in an error-free setting in Section 4.2.

We will then go over this algorithm in detail, and present exactly in what ways CIRCLE FORMATION is made impossible by the presence of measurement errors under the *distance-proportional disk* error model in Section 4.3. We present the weakened problem of APPROXIMATE CIRCLE FORMATION and explain how the previously-presented algorithm can be adapted to solve this problem instead. We finish with a brief discussion in Section 4.4.

4.1 Design space analysis

As with GATHERING in Section 3.1, the nature of the problem being solved provides a rather narrow design space. As a reminder, an algorithm controlling the robots is a deterministic function that, for each robot, maps the configuration at the time of the *Look* phase of the activation cycle to a movement vector, both in the local disoriented reference frame of that robot.

Based on different symmetries that can exist, we can classify configurations in different ways:

A dense point exists
: A dense point is defined as a position in which two or more robots are located. Regardless of any design choices on the hand of an algorithm designer, an adversary can always present the exact same snapshot to the robots in the dense point, implying that they will always produce the same movement vector. Hence, in the next round, the dense point will still exist.

  Given that, therefore, the adversary can maintain the dense point indefinitely, the CIRCLE FORMATION problem is unsolvable from any configuration that features such a dense point, unless degenerate cases are accepted where a single point is considered a valid circle.

Robots form an equilateral triangle
: A requirement of the problem statement is that once the equilateral triangle is formed, it is also maintained indefinitely. Given that robots are disoriented and do not agree on a unit of measure, it is not really possible to distinguish one equilateral triangle from another based on size or orientation. Hence, the only choice an algorithm designer has here is to instruct the robot not to move.

Robots form an isosceles triangle
: As mentioned in Section 3.1, in a configuration where the robots form an isosceles triangle, mirror symmetry exists along the bisector of the apex of the triangle. The robot at the apex is uniquely distinguishable, whereas possible disagreements about chirality make the base robots indistinguishable. An algorithm designer must choose which of them to move.

If the algorithm moves the apex robot, it can:

- Move the robot along the bisector of the apex, which will change the height of the isosceles triangle while maintaining the isosceles property.

  Doing this, it can move the robot to hit special cases, such as forming a co-linear formation with one robot at the median point between between the base robots, or moving it to form an equilateral triangle, immediately completing the task.
- Move the robot away from the bisector, resulting in either the formation of a scalene triangle, a line, or the formation of a dense point. Note that, as stated earlier, circle formation becomes impossible in the latter case.

It is also possible for the algorithm to move the robots at the base of the triangle. However, given the fact that these robots are potentially indistinguishable and that the scheduler may choose to activate either one (or both) of the robots, an algorithm designer has to deal with some unpredictability here. Depending on what the adversary chooses, this may end up breaking or maintaining the fact that an isosceles triangle exists.

Robots form a scalene triangle

In configurations where the robots from the vertices of a scalene triangle, since angles are of a different size and have a unique order regardless of reference frame, robots are easy to distinguish from one another.

A straightforward solution to the circle formation problem here, is to simply pick one of the robots, for example at the largest angle, and instruct it to form an equilateral triangle with the two others. Since the scheduler is fair, this and only this robot will eventually make such a move.

Robots form a line

Unlike with the Gathering problem, the line configuration provides a symmetry that needs to be broken. Still, this can be considered a degenerate case of the scalene or isosceles triangle, depending on whether the middle robot is at the median point between the two others.

Overall, the design space of the circle-formation problem is fairly similar to the design space of the gathering problem as shown in Section 3.1 despite the goals being "inverted" in a sense.

4.2 Example of a simple circle formation algorithm

Given the information presented in the previous section, we present a fairly simple algorithm to solve CIRCLE FORMATION for three robots under an SSYNC scheduler with rigid movement guarantees: the observing robot r_0 checks whether, in the triangle formed by the three robots, it is opposite an edge of maximum length. If so, it moves to form an equilateral triangle with the other two robots. This is formalized in Aglorithm 3.

Input: Positions c_1^*, c_2^* in the local frame of reference of robot r_0.
Output: Movement vector in the local frame of reference of robot r_0.

1: $c_0^* \leftarrow (0,0)$
2: **if** $|c_1^*, c_2^*| = \max\{|c_0^* c_1^*|, |c_1^* c_2^*|, |c_2^* c_0^*|\}$ **then**
3: Move to form an equilateral triangle with c_1^* and c_2^*.
4: **else**
5: Do not move.
6: **end if**

Algorithm 3: Formation of an equilateral triangle without measurement errors under an SSYNC scheduler and rigid movement. Inputs are the relative positions of the two others, with the position of the observer $c_0^* = (0,0)$.

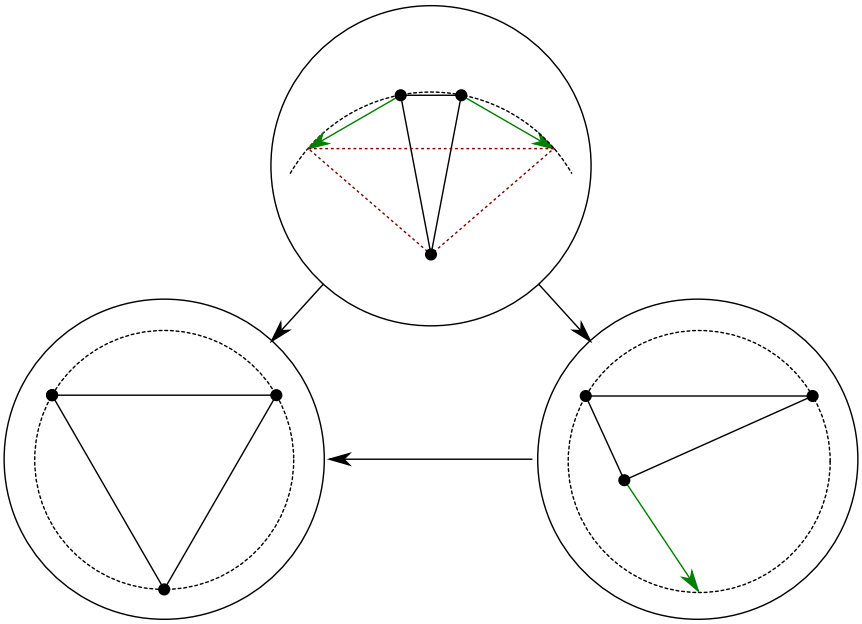


Figure 22: Illustration of Algorithm 3.

Theorem 6. Algorithm 3 solves the CIRCLE FORMATION problem under SSYNC with three robots and rigid movement, assuming that no two robots start in the same position.

Proof. We split into three different cases based on the type of triangle formed by the three robots: a perfectly equilateral triangle, a triangle with a distinct longest edge, and an isosceles triangle where the legs are longer than the base.

Robots form an equilateral triangle
If all robots are arranged in an equilateral triangle, then forming an equilateral triangle with the other robots is equivalent to staying in place.

Robots form a triangle with a unique longest edge
If the robots are arranged such that a unique longest edge exists, then the robot opposite that edge will move to form an equilateral triangle.

Robots form an isosceles triangle with a base shorter than the other two sides
If the robots are arranged in an isosceles triangle where the base is shorter than the other two edges, then both robots at the base will move if active. If the adversary activates only one, an equilateral triangle is formed.

If the adversary activates both, both will move in a way that creates a unique longest edge in the resulting triangle. We then achieve an equilateral triangle by a previous case.

In all of the cases above, the robots either maintain or eventually form an equilateral triangle. See Figure 22 for an illustration of how this works. □

4.3 Making the algorithm robust

Algorithm 3 can solve CIRCLE FORMATION for three robots in the absence of measurement errors. The introduction of measurement errors (according to the *distance-proportional-disk* error model), however, com-

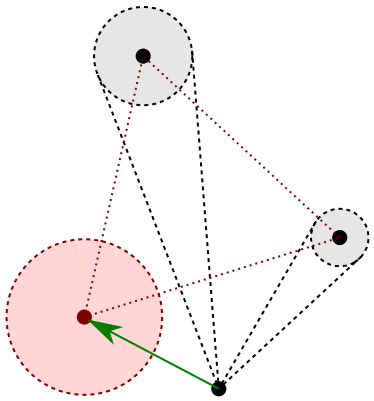


Figure 23: Illustration of Lemma 23, where the robot at the bottom constructs a third point that forms an equilateral triangle with the positions of the two others. The noise regions add up to the noise region of the third point, as depicted in red.

plicates matters considerably.

Approximate circle formation It must be noted here that for any pattern that is not simply a single point, there is always some distance between any pair of robots. As a result, under most error models, especially distance-dependant error models like the *angle-distance* error model and the *distance-proportional disk* error model, it effectively becomes impossible to reliably detect whether a desired pattern has been formed. The CIRCLE FORMATION problem is no exception, and therefore unfortunately cannot be solved exactly.

Instead, we propose the following:

Definition 12. Given a set of n robots R, the APPROXIMATE CIRCLE FORMATION problem requires that robots must eventually be positioned such that the configuration approaches a regular n-gon within some defined degree of error proportional to ϵ_0.

For $n = 3$, we shall define the degree of approximation as the ratio $t_0 \in [0, 1]$ of the length of the shortest edge of the triangle to the longest edge, where the closer t_0 is to 1, the better the approximation. Special cases are $t_0 = 1$, where the triangle is equilateral, and $t_0 = 0$, where the triangle is degenerate with two vertices coinciding.

Forming an equilateral triangle with the other robots As a result of the presence of measurement errors, a robot can no longer access the exact relative position of other robots c_1^*, c_2^* directly, and must instead move using positions affected by measurement errors: s_1^*, s_2^*. As a reminder, $c_0^* = s_0^* = (0, 0)$, but we shall use s_0^* for consistency.

We shall start making the algorithm robust by analysing the idea of forming an equilateral triangle through the movement of a single, unique robot to a point constructed from the measured positions s_1^*, s_2^* of the two other robots. We analyse the resulting location of that robot through Lemma 23:

Lemma 23. Let some robot $r_i \in R$, located at global position c_i, observe two other robots $r_j, r_k \in R$, located at global positions c_j, c_k. Then, the noise range of the point closest to c_i that forms an equilateral triangle with the two robots is a disk of radius $\epsilon_0(|c_i, c_j| + |c_i, c_k|)$ centered on a point that forms an exact equilateral triangle with c_j and c_k.

Proof. Given candidate values for points s_j and s_k, we can obtain a point p that forms an equilateral triangle with s_j and s_k by rotating s_k by $\pi/3$ radians around s_j in whichever direction yields a point closer to c_i.

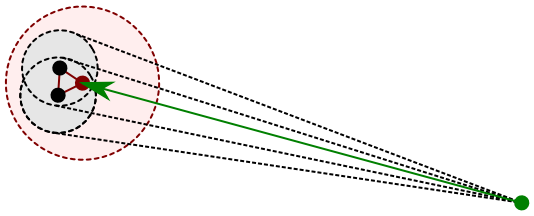


Figure 24: Example of the issue of two robots being close together: in this case, the robot further out cannot know what is actually going on: the adversary can simply make the two robots coincide in position if it wants, and the position of "third point of an equilateral triangle" becomes unclear and undefined. Moreover, even the optimal position cannot be reached, as doing so risks colliding with another robot. The two robots closest together are the only ones in a position to resolve this situation within $O(1)$ full rounds.

Thus, if we pick a value for s_j, and let s_k vary within the noise region of r_k, we obtain that p simply varies within that same noise region, rotated around s_j. Now, if we vary s_j within the noise region of r_j as well, we effectively obtain that p varies within the Minkowski sum of the two noise regions, centered around where p would be if $s_j = c_j$ and $s_k = c_k$. We essentially employ a similar argument as we did in Lemma 7. □

Lemma 23 immediately leads to something that we must be careful about: the quality of the output greatly depends on how far the robot that moves is from the other two robots, relative to the distance between those robots. The further the observer is from the two robots which are close together, the larger the noise region of the destination point will be compared to what would be a point that forms an equilateral triangle. See Figure 24 for an especially severe example of this problem.

How to deal with this issue is up to the algorithm designer. At the very least, we will want to avoid collisions between robots, and we would like to guarantee eventual termination, to the extent that this is possible under the constraints lined out in Section 2.2. However, actually reaching a bounded approximation of the target pattern within $O(1)$ full rounds is not always easy while preserving the original structure of the algorithm.

The perhaps somewhat odd design of Algorithm 3 is therefore not a coincidence: it has been designed to to avoid movements like those in Figure 24. While at this stage, an algorithm designer need not explicitly take measurement errors into account, they should try to avoid moving a robot in such a way that it ends up close to another robot, where a small error could mean a collision or other such issues.

Choosing whether to move Following the analysis of the idea of "moving to form an equilateral triangle", we also analyse the binary choice made in the algorithm:

$$|c_1^*, c_2^*| = \max\{|c_0^* c_1^*|, |c_1^*, c_2^*|, |c_2^*, c_0^*|\}.$$

That is, the check passes if the observing robot is opposite an edge that is as long as the longest edge in the triangle.

Obviously, once measurement errors are applied, the algorithm must be expressed in terms of points s_0^*, s_1^*, s_2^* instead of c_0^*, c_1^*, c_2^*. Moreover, we introduce the centers of the uncertainty regions u_0^*, u_1^*, u_2^* (see Lemma 5), which shall be used when making distance comparisons.

What is important is that, however we check whether the current robot is opposite the longest edge of the triangle, this check does not produce false-negatives. For instance, once the formation approaches an equilateral triangle where the base is the shortest angle, the adversary may get the opportunity to distort distances sufficiently to make the longest edge appear shorter than another edge, and thus make the robot facing such a longest edge unable to move.

To remedy such a potential deadlock, based on the radii of the uncertainty regions (see Lemma 5) we use the threshold $\epsilon_0(|u_0^*, u_1^*| + |u_0^*, u_2^*|)$ twice: once to prevent under-estimating the value of $|c_1^*, c_2^*|$, and once more to

avoid over-estimating the length of the longest edge. This way, the following test ensures no false-negatives occur:

$$|u_1^*, u_2^*| \geq \max\{|u_0^*, u_1^*|, |u_1^*, u_2^*|, |u_2^*, u_0^*|\} - 2\epsilon_0(|u_0^*, u_1^*| + |u_0^*, u_2^*|)$$

However, even if false-negatives can no longer occur, it is inevitable that we will have to deal with at least some false-positives. Specifically:

Lemma 24. If a robot r_0 observes that

$$|u_1^*, u_2^*| \geq \max\{|u_0^*, u_1^*|, |u_1^*, u_2^*|, |u_2^*, u_0^*|\} - 2\epsilon_0(|u_0^*, u_1^*| + |u_0^*, u_2^*|)$$

then

$$|c_1^*, c_2^*| \geq (1 - \frac{8\epsilon_0}{1-\epsilon_0}) \max\{|c_0^* c_1^*|, |c_1^* c_2^*|, |c_2^* c_0^*|\}$$

Proof. Following the uncertainty region radii from Lemma 5:

$$\begin{aligned}
&|u_1^*, u_2^*| \geq \max\{|u_0^*, u_1^*|, |u_1^*, u_2^*|, |u_2^*, u_0^*|\} - 2\epsilon_0(|u_0^*, u_1^*| + |u_0^*, u_2^*|) \\
&\Rightarrow |c_1^*, c_2^*| \geq \max\{|c_0^* c_1^*|, |c_1^* c_2^*|, |c_2^* c_0^*|\} - 4\epsilon_0(|u_0^*, u_1^*| + |u_0^*, u_2^*|) \\
&\Rightarrow |c_1^*, c_2^*| \geq \max\{|c_0^* c_1^*|, |c_1^* c_2^*|, |c_2^* c_0^*|\} - \frac{4\epsilon_0}{1-\epsilon_0}(|c_0^*, c_1^*| + |c_0^*, c_2^*|) \\
&\Rightarrow |c_1^*, c_2^*| \geq (1 - \frac{8\epsilon_0}{1-\epsilon_0}) \max\{|c_0^* c_1^*|, |c_1^* c_2^*|, |c_2^* c_0^*|\}
\end{aligned}$$

□

Near-equilateral configurations False-positives can become quite problematic if the wrong robots start moving. We start by analysing the case where three robots all see themselves as possibly being opposite a longest edge.

For instance, the argument "if all robots are arranged in an equilateral triangle, then forming an equilateral triangle with the other robots is equivalent to staying in place" no longer holds. This is because, as established in Lemma 23, the optimal position to move to when forming an equilateral triangle with two other robots is uncertain if the position of those robots is uncertain, and hence the adversary could manipulate this position to move the robots out of the equilateral triangle. While the configuration may still remain close to an equilateral triangle within a bounded degree of error, instability or diverging in size is likely to be an undesirable outcome.

A simple way to deal with this problem is to add an extra check to make sure that the edge that appears longest is actually longer than the shortest edge:

$$|u_1^*, u_2^*| > \min\{|u_0^*, u_1^*|, |u_1^*, u_2^*|, |u_2^*, u_0^*|\} + 2\epsilon_0(|u_0^*, u_1^*| + |u_0^*, u_2^*|)$$

At least one robot will be facing an edge in the triangle that is not strictly longer than the shortest edge in the triangle, and will thus not move. Three robots moving at the same time is now no longer possible.

Moving both robots at the base of the approximate isosceles triangle We are left with the possibility of two robots observing themselves as possibly being opposite a longest edge in the triangle. Since robots must now be certain that they are not facing the shortest edge in the triangle, these will be the robots facing the longest and second-longest edge in the triangle. Moreover, since there is confusion about which edge is longest, these edges must close together in length, depending on the value of ϵ_0. Therefore, this situation is actually not too different from the case where the robots form an equilateral triangle where the base is the shortest edge, and both robots at the base are active (see the top case depicted in Figure 22).

We introduce a lemma to understand what exactly happens when measurement errors are added to the mix:

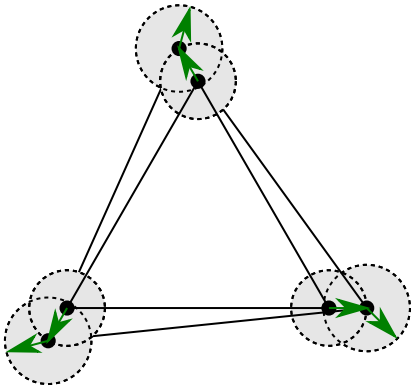


Figure 25: If it is possible for all three to move at the same time, the adversary can cause the resulting triangle to diverge in size or prevent stability, even if the overall eccentricity of the triangle is bounded.

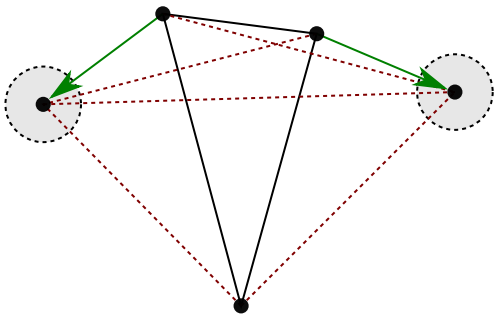


Figure 26: The two robots that form the vertices of the shortest edge will both move to somewhere within the two grey noise ranges. While the behavior is similar, the uncertainty in the resulting formation does require a few extra checks.

Lemma 25. Let r_0, r_1, r_2 be three robots, located at c_0, c_1, c_2 respectively, where

$$|c_1, c_2| < \max\{|c_0, c_1|, |c_0, c_2|\}$$

and

$$\min\{|c_0, c_1|, |c_0, c_2|\} \geq t \max\{|c_0, c_1|, |c_0, c_2|\}$$

for some $0 < t \leq 1$.

Now, suppose that r_1, r_2 both move in an attempt to form an equilateral triangle with the two other robots. Then, after the move:

$$\min\{|c_0^{i+1}, c_1^{i+1}|, |c_0^{i+1}, c_2^{i+1}|\} \geq (t - \frac{(2+2t)\epsilon_0}{1-2\epsilon_0}) \max\{|c_0^{i+1}, c_1^{i+1}|, |c_0^{i+1}, c_2^{i+1}|\}).$$

Proof. Through Lemma 23, we know that r_1, r_2 will both end up in disks of radius at most $2\epsilon_0 \max\{|c_0, c_1|, |c_2, c_0|\}$ centered on points p_1, p_2, which would be two points forming equilateral triangles with c_0, c_2 and c_0, c_1,respectively. Therefore:

$$\max\{|c_0^{i+1}, c_1^{i+1}|, |c_0^{i+1}, c_2^{i+1}|\} \geq (1 - 2\epsilon_0) \max\{|c_0, c_1|, |c_2, c_0|\}$$

We employ that inequality in the derivation:

$$\begin{aligned}
&\min\{|c_0, c_1|, |c_0, c_2|\} \geq t \max\{|c_0, c_1|, |c_0, c_2|\} \\
&\Rightarrow \min\{|c_0^{i+1}, c_1^{i+1}|, |c_0^{i+1}, c_2^{i+1}|\} + 2\epsilon_0 \max\{|c_0, c_1|, |c_0, c_2|\}) \geq t(\max\{|c_0^{i+1}, c_1^{i+1}|, |c_0^{i+1}, c_2^{i+1}|\} - 2\epsilon_0 \max\{|c_0, c_1|, |c_0, c_2|\}) \\
&\Rightarrow \min\{|c_0^{i+1}, c_1^{i+1}|, |c_0^{i+1}, c_2^{i+1}|\} \geq t \max\{|c_0^{i+1}, c_1^{i+1}|, |c_0^{i+1}, c_2^{i+1}|\} - (2t\epsilon_0 + 2\epsilon_0) \max\{|c_0, c_1|, |c_0, c_2|\} \\
&\Rightarrow \min\{|c_0^{i+1}, c_1^{i+1}|, |c_0^{i+1}, c_2^{i+1}|\} \geq (t - \frac{(2+2t)\epsilon_0}{1-2\epsilon_0}) \max\{|c_0^{i+1}, c_1^{i+1}|, |c_0^{i+1}, c_2^{i+1}|\}
\end{aligned}$$

□

After they both moved, we wish to prevent r_1, r_2 from moving again, leaving the robot that stayed in place as the only robot that can move. Hence, we must simply specify that the robot may only move if the edge that it is facing is larger than the shortest edge by such a margin that it could not have been one of the legs of an approximately isosceles triangle on the previous round:

$$\min\{|c_0^*, c_1^*|, |c_0^*, c_2^*|, |c_1^*, c_2^*|\} < (t - \frac{(2+2t)\epsilon_0}{1-2\epsilon_0})|c_1^*, c_2^*|\}).$$

When adjusting for uncertainty regions, the algorithm simply needs to check that

$$\min\{|u_0^*, u_1^*|, |u_0^*, u_2^*|, |u_1^*, u_2^*|\} < (t - \frac{(2+2t)\epsilon_0}{1-2\epsilon_0})|u_1^*, u_2^*| - \epsilon_0(1 + t - \frac{(2+2t)\epsilon_0}{1-2\epsilon_0})(|u_0^*, u_1^*| + |u_0^*, u_2^*|).$$

If we fill in $t = 1 - \frac{8\epsilon_0}{1-\epsilon_0}$ from Lemma 24, this becomes

$$\min\{|u_0^*, u_1^*|, |u_0^*, u_2^*|, |u_1^*, u_2^*|\} < \frac{1 - 15\epsilon_0 + 38\epsilon_0^2}{(\epsilon_0 - 1)(2\epsilon_0 - 1)}|u_1^*, u_2^*| - \epsilon_0 \frac{2 - 18\epsilon_0 + 40\epsilon_0^2}{1 - 3\epsilon_0 + 2\epsilon_0^2}(|u_0^*, u_1^*| + |u_0^*, u_2^*|).$$

Stability after one moving Finally, as a consequence of Lemma 23, once we succeed in ensuring that a single robot moves to form an approximate equilateral triangle, we should ideally be able to detect that such a triangle has, in fact, been formed. If we do not do this, the resulting configuration can be unstable.

We derive a short lemma to help with this:

Lemma 26. Let robot $r_0 \in R$ located at c_0, be observing robots $r_1, r_2 \in R$, at c_1, c_2 such that $|c_1, c_2| = t \max\{|c_0, c_1|, |c_1, c_2|, |c_2, c_0|\}$, where $0 < t \leq 1$. Then, on the next round:

$$\min\{|c_0^{i+1}, c_1^{i+1}|, |c_1^{i+1}, c_2^{i+1}|, |c_2^{i+1}, c_0^{i+1}|\} \leq \frac{1 - 2\epsilon_0 t^{-1}}{1 + 2\epsilon_0 t^{-1}} \max\{|c_0^{i+1}, c_1^{i+1}|, |c_1^{i+1}, c_2^{i+1}|, |c_2^{i+1}, c_0^{i+1}|\}$$

Proof. Through Lemma 23, r_0 will move to within distance $\epsilon_0 \max\{|c_0, c_1|, |c_1, c_2|, |c_2, c_0|\}$ from a point exactly equidistant from c_1 and c_2, whereas r_1, r_2 to not move. Hence

$$\max\{|c_0^{i+1}, c_1^{i+1}|, |c_1^{i+1}, c_2^{i+1}|, |c_2^{i+1}, c_0^{i+1}|\} \leq (1 + 2\epsilon_0 t^{-1})|c_1^{i+1}, c_2^{i+1}|$$

and

$$\min\{|c_0^{i+1}, c_1^{i+1}|, |c_1^{i+1}, c_2^{i+1}|, |c_2^{i+1}, c_0^{i+1}|\} \geq (1 - 2\epsilon_0 t^{-1})|c_1^{i+1}, c_2^{i+1}|$$

This directly leads us to the given length ratio. □

Intuitively, what this lemma means is that, if the robot that ends up moving is not actually opposite the longest edge in the triangle (false-positive), but instead one at least t times as long, the degree of error in the resulting formation is still bounded, avoiding a situation as depicted in Figure 24. What we will want to do, at this point, is avoid moving if the triangle could be such an approximately equilateral triangle.

For this, we require that the following passes before the robot is allowed to move:

$$\min\{|u_0^*, u_1^*|, |u_1^*, u_2^*|, |u_2^*, u_0^*|\} < \frac{1 - 2\epsilon_0 t^{-1}}{1 + 2\epsilon_0 t^{-1}}|u_1^*, u_2^*| - \epsilon_0(1 + \frac{1 - 2\epsilon_0 t^{-1}}{1 + 2\epsilon_0 t^{-1}})(|u_0^*, u_1^*| + |u_1^*, u_2^*|).$$

After filling in $t = 1 - \frac{8}{1-\epsilon_0}$ from Lemma 24, this becomes:

$$\min\{|u_0^*, u_1^*|, |u_1^*, u_2^*|, |u_2^*, u_0^*|\} < \frac{7 + 3\epsilon_0 - 2\epsilon_0^2}{7 - \epsilon_0 + 2\epsilon_0^2} \max\{|u_0^*, u_1^*|, |u_1^*, u_2^*|, |u_2^*, u_0^*|\} - \epsilon_0 \frac{2(\epsilon_0 + 7)}{\epsilon_0(2\epsilon_0 - 1) + 7}(|u_0^*, u_1^*| + |u_1^*, u_2^*|).$$

Basically: "Do not move if the difference between the edge you are facing and the smallest edge is small enough that the current formation could be the result of a single robot moving to form an equilateral triangle."

Bringing it together Summarizing, the robot should move if and only if all of the following hold:

1. The opposite edge could be the longest edge in the triangle:

$$|u_1^*, u_2^*| \geq \max\{|u_0^*, u_1^*|, |u_1^*, u_2^*|, |u_2^*, u_0^*|\} - 2\epsilon_0(|u_0^*, u_1^*| + |u_0^*, u_2^*|)$$

2. The opposite edge is actually longer than the shortest edge in the triangle:

$$|u_1^*, u_2^*| > \min\{|u_0^*, u_1^*|, |u_1^*, u_2^*|, |u_2^*, u_0^*|\} + 2\epsilon_0(|u_0^*, u_1^*| + |u_0^*, u_2^*|)$$

3. The robot was not one of two robots active on the previous round:

$$\min\{|u_0^*, u_1^*|, |u_0^*, u_2^*|, |u_1^*, u_2^*|\} < \frac{1 - 15\epsilon_0 + 38\epsilon_0^2}{(\epsilon_0 - 1)(2\epsilon_0 - 1)}|u_1^*, u_2^*| - \epsilon_0 \frac{2 - 18\epsilon_0 + 40\epsilon_0^2}{1 - 3\epsilon_0 + 2\epsilon_0^2}(|u_0^*, u_1^*| + |u_0^*, u_2^*|).$$

4. The current configuration is not the result of a single robot moving during some previous round:

$$\min\{|u_0^*, u_1^*|, |u_1^*, u_2^*|, |u_2^*, u_0^*|\} < \frac{7 + 3\epsilon_0 - 2\epsilon_0^2}{7 - \epsilon_0 + 2\epsilon_0^2}|u_1^*, u_2^*| - \epsilon_0 \frac{2(\epsilon_0 + 7)}{\epsilon_0(2\epsilon_0 - 1) + 7}(|u_0^*, u_1^*| + |u_1^*, u_2^*|).$$

Up to $\epsilon_0 \leq 0.335$ (rounded down), we can simply drop points 2 and 3, since number 4 implies those.

We then combine these checks into the final, resulting Algorithm:

Input: Positions s_1^*, s_2^* in the local frame of reference of robot r_0.
Output: Movement vector in the local frame of reference of robot r_0.

$s_0^* \leftarrow (0,0)$
$u_0^* \leftarrow (0,0)$
$u_1^* \leftarrow \frac{s_1^* - s_0^*}{1-\epsilon_0^2} + s_0^*$
$u_2^* \leftarrow \frac{s_2^* - s_0^*}{1-\epsilon_0^2} + s_2^*$
$d_{\min} \leftarrow \min\{|u_0^*, u_1^*|, |u_1^*, u_2^*|, |u_2^*, u_0^*|\}$
$d_{\max} \leftarrow \max\{|u_0^*, u_1^*|, |u_1^*, u_2^*|, |u_2^*, u_0^*|\}$
$\Delta \leftarrow |u_0^*, u_1^*| + |u_0^*, u_2^*|$
if $d_{\max} - 2\epsilon_0\Delta \leq |u_1^*, u_2^*|$ **and** $d_{\min} < \frac{7+3\epsilon_0-2\epsilon_0^2}{7-\epsilon_0+2\epsilon_0^2}|u_1^*, u_2^*| - \epsilon_0\frac{2(\epsilon_0+7)}{\epsilon_0(2\epsilon_0-1)+7}\Delta$ **then**
    Move to the closest point that forms an equilateral triangle with s_1^* and s_2^*.
else
    Do not move.
end if

Algorithm 4: Formation an approximate equilateral triangle with measurement errors under SSync with three robots. Since we are working in the local reference frame of r_0, the positions c_0^*, s_0^* and u_0^* are simply $(0,0)$, though we treat them separately for semantic reasons.

Theorem 7. Algorithm 4 solves approximate circle formation under SSync with 3 robots and rigid movement, with a shortest-to-longest edge ratio in the triangle formed by the robots of

$$t_0 = \frac{60\epsilon_0 + 13\epsilon_0^2 - 2\epsilon_0^3 - 7}{(\epsilon_0 - 1)(-\epsilon_0 + 2\epsilon_0^2 + 7)}$$

with $\epsilon_0 < 0.335$ (rounded down).

Proof. Moreover, by Lemma 24, the edge that appears longest is at least $1 - \epsilon_0$ as long as the edge that is actually longest.

Case: 1 robot moves The thresholds are designed such that no robot will move on subsequent rounds if a single robot moves.

Case: 2 robots move By Lemma 25, one of two things will happen: only the non-moving robot will see itself as opposite the longest edge on subsequent rounds, reducing this to the case where one robot moves, or no robot moves again, which is the last case in this analysis.

Case: 3 robots move This case cannot occur since no three robots can observe that the side of the triangle formed by the two opposite robots is certainly strictly longer than the shortest edge of the triangle.

Case: No active robots move Robots will move if and only if both of the following are true:

$$\begin{cases} \max\{|u_0^*, u_1^*|, |u_1^*, u_2^*|, |u_2^*, u_0^*|\} - 2\epsilon_0(|u_0^*, u_1^*| + |u_0^*, u_2^*|) \leq |u_1^*, u_2^*| \\ \min\{|u_0^*, u_1^*|, |u_1^*, u_2^*|, |u_2^*, u_0^*|\} < \frac{7+3\epsilon_0-2\epsilon_0^2}{7-\epsilon_0+2\epsilon_0^2}|u_1^*, u_2^*| - \epsilon_0\frac{2(\epsilon_0+7)}{\epsilon_0(2\epsilon_0-1)+7}(|u_0^*, u_1^*| + |u_0^*, u_2^*|) \end{cases}$$

Either a robot does eventually move (see other cases), or the scheduler, being fair, will eventually activate a robot which is facing an edge of maximum length in the triangle. For that robot, the first check always passes, leaving us with the second one. If the robot does not move, we know that:

$$\min\{|u_0^*, u_1^*|, |u_1^*, u_2^*|, |u_2^*, u_0^*|\} + \frac{2(\epsilon_0+7)}{\epsilon_0(2\epsilon_0-1)+7}\epsilon_0(|c_0^*, c_1^*| + |c_0^*, c_2^*|) \geq \frac{7+3\epsilon_0-2\epsilon_0^2}{7-\epsilon_0+2\epsilon_0^2}|u_1^*, u_2^*|$$

Which translates to:

$$l_{\min} + \frac{4\epsilon_0(\epsilon_0+7)}{(1-\epsilon_0)(\epsilon_0(2\epsilon_0-1)+7)}(|u_0^*,u_1^*|+|u_0^*,u_2^*|) \geq \frac{7+3\epsilon_0-2\epsilon_0^2}{7-\epsilon_0+2\epsilon_0^2} l_{\max}$$

where $l_{\min} = \min\{|c_0,c_1|,|c_1,c_2|,|c_2,c_0|\}$ and $l_{\max} = \max\{|c_0,c_1|,|c_1,c_2|,|c_2,c_0|\}$. We then further condense this to a shortest-to-longest edge ratio of

$$\frac{60\epsilon_0+13\epsilon_0^2-2\epsilon_0^3-7}{(\epsilon_0-1)(-\epsilon_0+2\epsilon_0^2+7)}.$$

In either case, we eventually achieve a longest-to-shortest edge ratio of at most $\frac{60\epsilon_0+13\epsilon_0^2-2\epsilon_0^3-7}{(\epsilon_0-1)(-\epsilon_0+2\epsilon_0^2+7)}$. If robots do move, they can only move a finite number of times until they hit the case where only one robot moves, and stability is guaranteed. Otherwise, stability is achieved by default. □

4.4 Discussion and conclusion

In this section of the thesis, we explored the Circle Formation problem for three robots under an SSync scheduler with rigid movement, and how this problem is affected by measurement errors.

We explored different possible kinds of configurations that could occur when dealing with three robots, such as different kinds of triangles, lines or dense points, concluding that the latter makes Circle Formation impossible in an adversarial context. We then presented a very simple Circle Formation algorithm that moves the robots into an equilateral triangle in a constant number of full rounds.

Then, we added measurement errors to the mix. Immediately, we concluded that Circle Formation is impossible in this context because it requires exact distance measurements: robots cannot move to an exact relative position if the data it uses to compute this position is inaccurate.

Subsequently, we defined the Approximate Circle Formation problem as an approximation of the Circle Formation problem, defining a metric of error between the exact and approximate formation for $n = 3$, leaving the definition for higher n undefined.

An important result here, was that certain kinds of algorithms, even if they work fine without measurement errors, cannot easily be made meaningfully robust without preventing collisions or guaranteeing any bounded inaccuracy in the output.

Finally, we took the Circle Formation algorithm that we established previously, and explored in detail on which points it succeeds and fails when confronted with measurement errors. We established additional thresholds to prevent it from deadlocking, and to ensure stability after an approximate equilateral triangle was created to within a bounded degree of error.

The process of making the algorithm robust was comparatively simple in comparison to Approximate Gathering in Section 3, despite the apparent similarities between Gathering and Circle Formation. However, it should be noted that Algorithm 3 is a bit simpler than Algorithm 1: it does not contain the second check where, once it has been established that the triangle is not equilateral, the role within the formation must be determined, nor are any disambiguation movements necessary.

A likely cause for this simplicity is that Algorithm 3 does not explicit require coordination between robots, where one robot must stay in place such that another may move to that position, instead simply allowing robots to move, and reasoning about the consequences. While this reasoning breaks own a bit as we saw with the issue where three robots activate at once in the presence of measurement errors, where the formation can be unstable or grow and shrink, it apparently only requires adding a check to make sure that the triangle is sufficiently eccentric to get a bounded approximation.

5 Towards a general framework

Over the course of this document, we have presented some general information about the OBLOT model, and how to extend it to feature measurement errors.

We then explored the problems of Gathering and Circle Formation for three robots under an SSync scheduler and, for each, explained how an example algorithm could be made robust against measurement errors in such a way that an approximation of the original result is obtained.

Obviously, the choice of just using three robots is somewhat limited. Nevertheless, the problems encountered even in such a simple setting can be surprisingly numerous, and unfortunately often hard to reason about and somewhat counter-intuitive.

For instance, consensus is difficult to reliably obtain, and there will always be a certain degree of "asymmetry" when reasoning about which robot knows what about the situation; if some robot $r_1 \in R$ knows a proposition to be true about a configuration, then this does not mean that another robot $r_2 \in R$ necessarily agrees and that, even if $r_2 \in R$ does agree, that $r_1 \in R$ knows this. This is made even more difficult by the fact that robots are oblivious, and hence cannot even remember what happened in the past and have to guess each time based on noisy data.

As an example, take the problem outlined in Section 4 where two robots located at the base of an approximately isosceles triangle must ensure that they do not move again on the next round. In this case, while this task is not rendered impossible, the additional threshold directly comes at the cost of the quality of the resulting formation, due to the false-positives that it creates. If these robots each had even a single bit of memory about whether they had moved in the previous round, the solution could have been greatly simplified.

So far, with the two given examples, a clear path towards generalization to patterns other than points or circles remains elusive, as does any particular method for the preservation of robot behavior. However, a few points do appear to emerge that could prove useful for future research:

- Analysis is simplified if any discrete decisions in the algorithm are made using the centers of uncertainty regions rather than the actual measured positions of the robots.

  This type of analysis goes a long way to understanding which robots will move under what circumstances, and is generation of thresholds is relatively straightforward.

- Continuous decisions such as computing the movement vectors is better done using the actual measured relative positions of the other robots. Doing so takes advantage of the fact that these positions are much easier to reason about when considering the robots collectively, as they are simply dependant on the noise regions, which can be computed without considering each robot's individual perspective.

  Moreover, using these measured positions takes advantage of the fact that, should ϵ_0 prove to be an over-estimate for the actual size of the measurement errors, the algorithms will naturally converge towards a more accurate result.

- Using thresholds and offsets relative to the error parameter (such as ϵ_0 with the *distance-disk* error model) allows for the creation of algorithms that can be adjusted for the expected size of the error, and that converge to exact algorithms as ϵ_0 approaches 0.

- Reasoning about the collective behavior of robots is best done by considering the discrete decisions made by each robot as a mostly non-deterministic choice between any of the discrete branches of the algorithm.

  An earlier attempt to reason about the collective behavior of these robots from the perspective of an individual robot, intuitively understood as asking questions like "A robot decides to do X, what choices might other robots take, and what is the resulting configuration like?" yielded a somewhat complex and redundant analysis of possibilities of the future. Analysis from the position of a neutral observer that knows the actual configuration, but then faces an adversarial choice of the snapshot seen by each robot, proved to be advantageous.

- In situation where robots must rely on each other to perform a coordinated movement, robots should only take action if they are certain that consensus exists about the current situation.

  In Section 3 for instance, a robot $r_0 \in R$ must avoid moving if another robot $r_1 \in R$ is certainly further from the center of gravity than itself.

  It does not matter if r_1 actually knows this: it might, therefore r_0 must not move or risk interfering with r_1's attempt to move to r_0 and form a(n approximate) dense point. It is then up to r_1 to ensure that it does, in fact, eventually resolve the situation.

  We take advantage here of a property of the non-robust original algorithm: it functions under an SSync scheduler, and therefore can already deal with arbitrary robots not moving during any specific round. If careful to ensure that some robot eventually moves, the robust version of the algorithm can safely choose to not move certain robots in certain situations.

- The outcome of a certain movements may lead to uncertain situations

  In Section 3 for instance, we see that the proof of Theorem 5 counter-intuitively contains an ambiguous situation after a robot moved to the position of another robot, where a robot may observe *disagreement further*, a situation where we'd expect any ambiguities to have been resolved.

  For another example, in Section4, the proof of Theorem 5, a two-robot movement could directly result in the formation of an approximate equilateral triangle, rather than ending in a situation where one could guarantee that the robot that was stationary on the last round would move. In effect, the robots "skip" a step.

  In either case, though, the problem was manageable: if the robots did not quite make the same choices as they would do in the error-free setting, one can reason that the confusion between cases indicates that the formation is already very close to the target.

- Some groups of configurations may require "thickening".

  For instance, one may propose that the formation of an isosceles triangle as an intermediate stage towards some other formation. Once measurement errors are added, this effectively becomes impossible since such a shape requires exact measurements. You would essentially be trying to hit an infinitely small or thin target (a space with effectively no volume in the overall configuration space) without knowing its' exact location.

  Instead, one will want to loosen the definition of such an isosceles triangle, such that even if a robot that tries to form such a triangle actually misses the mark by some margin, robots observing the resulting configuration will still conclude that such a triangle has been formed.

  Moreover, depending on how important it is that this triangle is detected, one may have to loosen the margins such that the shape is never missed, at the cost of having to deal with false-positives. This was done with the "equilateral triangle" cases in both Sections 3 and 4.

To see to what extent these approaches generalize, it may prove useful to explore either a larger number of robots, or different starting algorithms or assumptions. Time constraints, combined with the surprising complexity of the problem and counter-intuitive interactions between robots, unfortunately did not permit such explorations for this thesis, but could be the subject of future work. As shown by the failed attempt in Appendix B, we must be very careful about the information asymmetry between different robots: the ambiguous situations were highly counter-intuitive, and presented many apparent solutions that turned out to lead to a deadlock or more erratic behavior. We do not rule out that modifications could be made to the algorithm in that attempt to make it work properly, but these are apparently hard to find.

As algorithms grow more complex, the error handling may grow more complex as well, possibly in a non-linear fashion. Especially if a large number of robots must interact, in a way where each has a unique role, it can be expected that an equally large number of potentially overlapping thresholds and complex analysis will be necessary. However, specifically for the case with *Approximate Gathering*, we do not currently see any obvious barriers that would prevent the algorithm from scaling to higher n.

As a concluding remark, the author of this thesis hopes that this work helps the reader understand and gain an intuition for measurement errors. We furthermore learn that we should be somewhat careful when reasoning about the applicability of solutions formulated in the OBLOT model: while we understand that their purpose is primarily to explore theoretical properties of this computational model, it may nonetheless be tempting to implement the algorithms in a simulator, or on real-life robots. Depending on the situation and environment, these will indeed exhibit varying amounts of sensor inaccuracy, and one can therefore indeed expect to run into the problems described in this thesis. Caution is, therefore, advised.

References

[1] Noa Agmon and David Peleg. "Fault-tolerant gathering algorithms for autonomous mobile robots". In: *SIAM Journal on Computing* 36.1 (2006), pp. 56–82.

[2] Zohir Bouzid, Maria Gradinariu Potop-Butucaru, and Sébastien Tixeuil. "Byzantine-Resilient Convergence in Oblivious Robot Networks". In: *Distributed Computing and Networking*. Ed. by Vijay Garg, Roger Wattenhofer, and Kishore Kothapalli. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 275–280. ISBN: 978-3-540-92295-7.

[3] Mark Cieliebak et al. "Distributed computing by mobile robots: Gathering". In: *SIAM Journal on Computing* 41.4 (2012), pp. 829–879.

[4] Reuven Cohen and David Peleg. "Convergence of autonomous mobile robots with inaccurate sensors and movements". In: *SIAM Journal on Computing* 38.1 (2008), pp. 276–302.

[5] Shantanu Das et al. "Autonomous mobile robots with lights". In: *Theoretical Computer Science* 609 (2016), pp. 171–184.

[6] Yoann Dieudonné, Ouiddad Labbani-Igbida, and Franck Petit. "Circle formation of weak mobile robots". In: *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 3.4 (2008), pp. 1–20.

[7] Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro. "Moving and Computing Models: Robots". In: *Distributed Computing by Mobile Entities: Current Research in Moving and Computing*. Ed. by Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro. Cham: Springer International Publishing, 2019, pp. 3–14. ISBN: 978-3-030-11072-7. DOI: 10.1007/978-3-030-11072-7_1. URL: https://doi.org/10.1007/978-3-030-11072-7_1.

[8] Paola Flocchini et al. "Distributed computing by mobile robots: Solving the uniform circle formation problem". In: *International Conference on Principles of Distributed Systems*. Springer. 2014, pp. 217–232.

[9] Paola Flocchini et al. "Gathering of asynchronous oblivious robots with limited visibility". In: *Annual Symposium on Theoretical Aspects of Computer Science*. Springer. 2001, pp. 247–258.

[10] Adam Heriban and Sébastien Tixeuil. "Mobile Robots with Uncertain Visibility Sensors". In: *Structural Information and Communication Complexity*. Ed. by Keren Censor-Hillel and Michele Flammini. Cham: Springer International Publishing, 2019, pp. 349–352. ISBN: 978-3-030-24922-9.

[11] Marcello Mamino and Giovanni Viglietta. "Square Formation by Asynchronous Oblivious Robots". In: *CCCG*. 2016.

[12] Kazuo Sugihara and Ichiro Suzuki. "Distributed motion coordination of multiple mobile robots". In: *Proceedings. 5th IEEE International Symposium on Intelligent Control 1990*. IEEE. 1990, pp. 138–143.

[13] Ichiro Suzuki and Masafumi Yamashita. "Distributed anonymous mobile robots—formation and agreement problems". In: *Proc. 3rd International Colloquium on Structural Information and Communication Complexity (SIROCCO'96)*. 1994, pp. 313–330.

[14] Ichiro Suzuki and Masafumi Yamashita. "Distributed anonymous mobile robots: Formation of geometric patterns". In: *SIAM Journal on Computing* 28.4 (1999), pp. 1347–1363.

[15] O Tanaka. "Forming a circle by distributed anonymous mobile robots". In: *Bachelor thesis, Department of Electrical Engineering, Hiroshima University, Hiroshima, Japan* (1992).

[16] Giovanni Viglietta. “Uniform Circle Formation”. In: *Distributed Computing by Mobile Entities*. Springer, 2019, pp. 83–108.

[17] Kenta Yamamoto et al. “Convergence of mobile robots with uniformly-inaccurate sensors”. In: *International Colloquium on Structural Information and Communication Complexity*. Springer. 2009, pp. 309–322.

A Derivations

The following appendix contains derivations that are part of the proofs of some of the lemmas in the thesis, but were too long to fit well into the main text. Variables and notation are declared in the lemma referenced in the title of each derivation.

A.1 Derivation of Lemma 5

Recall that, in the local frame of reference of an observer r_i, that c_i^* and s_i^* are both simply the origin point $(0,0)$, and can be used interchangeably. We start from the noise region in the local frame of reference, which is a disk of radius $\epsilon_0|c_i^*, c_j^*|$ around c_j^*. However remember that c_j^* is generally unknown to the observer. Given this noise region, the boundary is simply the curve corresponding to the equation $|s_j^*, c_j^*| = \epsilon_0|c_i^*, c_j^*|$.

$$
\begin{aligned}
&|s_j^*, c_j^*| = \epsilon_0|c_i^*, c_j^*| \\
\Leftrightarrow &|s_j^*, c_j^*|^2 = \epsilon_0^2|c_i^*, c_j^*|^2 \quad \text{where } \epsilon_0 \geq 0 \\
\Leftrightarrow &(s_x - c_x)^2 + (s_y - c_y)^2 = \epsilon_0^2(c_x^2 + c_y^2) \quad \text{where } (c_x, c_y) = c_j^* \text{ and } (s_x, s_y) = s_j^* \\
\Leftrightarrow &s_x^2 + c_x^2 - 2s_xc_x + s_y^2 + c_y^2 - 2s_yc_y = \epsilon_0^2(c_x^2 + c_y^2) \\
\Leftrightarrow &s_x^2 + (1-\epsilon_0^2)c_x^2 - 2s_xc_x + s_y^2 + (1-\epsilon_0^2)c_y^2 - 2s_yc_y = 0 \\
\Leftrightarrow &(1-\epsilon_0^2)s_x^2 + (1-\epsilon_0^2)^2c_x^2 - 2s_xc_x(1-\epsilon_0^2) + (1-\epsilon_0^2)s_y^2 + (1-\epsilon_0^2)^2c_y^2 - 2s_yc_y(1-\epsilon_0^2) = 0 \\
\Leftrightarrow &s_x^2 + (1-\epsilon_0^2)^2c_x^2 - 2s_xc_x(1-\epsilon_0^2) + s_y^2 + (1-\epsilon_0^2)^2c_y^2 - 2s_yc_y(1-\epsilon_0^2) = \epsilon_0^2(s_x^2 + s_y^2) \\
\Leftrightarrow &s_x^2/(1-\epsilon_0^2)^2 + c_x^2 - 2s_xc_x/(1-\epsilon_0^2) + s_y^2/(1-\epsilon_0^2)^2 + c_y^2 - 2s_yc_y/(1-\epsilon_0^2) = \epsilon_0^2(s_x^2 + s_y^2)/(1-\epsilon_0)^2 \\
\Leftrightarrow &(c_x - s_x/(1-\epsilon_0^2))^2 + (c_y - s_y/(1-\epsilon_0^2))^2 = \epsilon_0^2(s_x^2 + s_y^2)/(1-\epsilon_0)^2 \\
\Leftrightarrow &|c_i^*, s_j^*/(1-\epsilon_0^2)|^2 = \epsilon_0^2|s_i^*, s_j^*|^2/(1-\epsilon_0)^2 \\
\Leftrightarrow &|c_j^*, s_i^*/(1-\epsilon_0^2)| = |s_i^*, s_j^*|\epsilon_0/(1-\epsilon_0^2)
\end{aligned}
$$

The locus of that equation is simply the boundary of a disk centered on $s_i^*/(1-\epsilon_0^2)$ with radius $|s_i^*, s_i^*|\epsilon_0/(1-\epsilon_0^2)$, both of which are entirely known to the observer r_i.

A.2 Derivation of Lemma 14

Here, we start from the assumption made in Lemma 14 that robot r_0 observes that

$$|u_{a,0}^*, u_{COG,0}^*| > |u_{b,0}^*, u_{COG,0}^*| + (\frac{4\epsilon_0}{3} + \frac{16\epsilon_0(1+\epsilon_0)}{3(1-\epsilon_0)})(|u_{0,0}^*, u_{1,0}^*| + |u_{0,0}^*, u_{2,0}^*|).$$

After compensating for the uncertainty regions (see Lemmas 5, 8), substituting that $\forall r_i \in R : |c_0, c_0| \leq |u_0^*, u_i^*|(1+\epsilon_0)$, and noting that distance ratios and angles are preserved across reference frames, we get that

$$|c_a, c_{COG}| > |c_b, c_{COG}| + \frac{16\epsilon_0}{3(1-\epsilon_0)}(|c_0, c_1| + |c_0, c_2|).$$

Then, thanks to the triangle inequality, we know $|c_0, c_1| + |c_0, c_2| \geq (|c_1, c_0| + |c_1, c_2|)/2$. So we can rewrite

$$|c_a, c_{COG}| > |c_b, c_{COG}| + 8\frac{\epsilon_0}{3(1-\epsilon_0)}(|c_1, c_0| + |c_1, c_2|).$$

After re-applying the uncertainty radii in the opposite direction, now in the reference frame of r_1, we obtain that

$$|u^*_{a,1}, u^*_{COG,1}| > |u^*_{b,1}, u^*_{COG,1}| + \frac{4\epsilon_0}{3}(|u^*_{1,1}, u^*_{0,1}| + |u^*_{1,1}, u^*_{2,1}|.$$

Hence: r_1 can know for sure that r_a is indeed further from the COG than r_b. Also, note that the transition from local u^*-points to global c-points is valid given the nature of the transformation.

A.3 Derivation of Lemma 22

Given robots r_i, r_j, r_k, where r_i observes that

$$|u^*_{r\max}, u^*_{COG}| - \delta_1 \leq |u^*_i, u^*_{COG}| \leq |u^*_{r\min}, u^*_{COG}| + \delta_1$$

we know that the difference in distances from any robot to the COG is bounded as follows:

$$|c^t_{\max}, c^t_{COG}| - \frac{8\epsilon_0}{3(1-\epsilon_0)}(|c^t_i, c^t_1| + |c^t_i, c^t_2|) \leq |c^t_i, c^t_{COG}| \leq |c^t_{\min}, c^t_{COG}| + \frac{8\epsilon_0}{3(1-\epsilon_0)}(|c^t_i, c^t_1| + |c^t_i, c^t_2|).$$

Then, we compare the distances of r_i and r_j to the COG, treating r_k by symmetry.

$$\begin{aligned}
&|c^{t+1}_j, c^{t+1}_{COG}| \\
&\geq \text{(Robot } r_1 \text{ immobile, COG moves by third of } r_0\text{)} \\
&|c^t_1, c^t_{COG}| - \frac{1}{3}|c_i, c_{COG}| - \frac{\epsilon_0}{3}(|c_i, c_j| + |c_i, c_k|) \\
&\geq \text{(Minimum is always smaller.)} \\
&\min_{r_j \in R} |c^t_j, c^t_{COG}| - \frac{1}{3}|c^t_i, c^t_{COG}| - \frac{\epsilon_0}{3}(|c^t_i, c_j| + |c^t_i, c^t_2|) \\
&\geq \text{(Applying the known bound on distances to the COG.)} \\
&(2/3)|c^t_i, c^t_{COG}| - (\frac{\epsilon_0}{3} + \frac{8\epsilon_0}{3(1-\epsilon_0)})(|c^t_i, c^t_1| + |c^t_i, c^t_2|) \\
&\geq \text{(Split } |c^t_i, c^t_{COG}| \text{ into } (1/6)(|c^t_i, c^t_1| + |c^t_i, c^t_2|)\text{.)} \\
&(1/3)|c^t_i, c^t_{COG}| + (1/18 - \frac{\epsilon_0}{3} + \frac{8\epsilon_0}{3(1-\epsilon_0)})(|c^t_i, c^t_1| + |c^t_i, c^t_2|) \\
&\geq (m_{\vec{COG}} = \vec{m_0}/3, \text{ therefore } (|c^{t+1}_i, c^{t+1}_{COG}| - \frac{2\epsilon_0}{3}(|c^t_i, c^t_1| + |c^t_i, c^t_2|)) \leq |c^t_i, c^t_{COG}|/3) \\
&|c^{t+1}_i, c^{t+1}_{COG}|(1/18 - \epsilon_0 - \frac{8\epsilon_0}{3(1-\epsilon_0)})(|c^t_i, c^t_1| + |c^t_i, c^t_2|) \\
&\geq (r_0 \text{ approaches the COG, hence } |c^t_i, c^t_1| + |c^t_i, c^t_2| \leq |c^{t+1}_i, c^{t+1}_j| + |c^{t+1}_i, c^{t+1}_k|\text{.)} \\
&|c^{t+1}_i, c^{t+1}_{COG}| + (1/18 - \epsilon_0 + \frac{8\epsilon_0}{3(1-\epsilon_0)})(|c^{t+1}_i, c^{t+1}_j| + |c^{t+1}_i, c^{t+1}_k|) \\
&\geq \text{(Re-center the uncertainty on } r_1 \text{ (or } r_2 \text{ by symmetry).)} \\
&|c^{t+1}_i, c^{t+1}_{COG}| + (1/2)(1/18 - \epsilon_0 + \frac{8\epsilon_0}{3(1-\epsilon_0)})(|c^{t+1}_j, c^{t+1}_i| + |c^{t+1}_j, c^{t+1}_k|)
\end{aligned}$$

Hence, until any robot moves, r_j and r_k are guaranteed to observe themselves as further out by threshold δ_2 as long as

$$(1/2)(1/18 - \epsilon_0 + \frac{8\epsilon_0}{3(1-\epsilon_0)}) \geq \frac{\frac{4\epsilon_0}{3} + \frac{16\epsilon_0(1+\epsilon_0)}{3(1-\epsilon_0)}}{1-\epsilon_0} + \frac{4\epsilon_0}{3(1-\epsilon_0)},$$

which works out to a bound of $\epsilon_0 < 0.00384$ (rounded down).

A.4 Derivation of Lemma 19

Robot r_i observes robots r_j, r_k, under the premise that

$$|u_i^*, u_{COG}^*| + \delta_1 \geq |u_{\max}^*, u_{COG}^*|$$

which, after compensating for uncertainty regions, means that

$$|c_i, c_{COG}| + \frac{8\epsilon_0}{3}(|c_i, c_j| + |c_i, c_k|) \geq |c_k, c_{COG}|.$$

From there, employing the fact that the COG is exactly two thirds of the way down each of the medians of the triangle and the triangle inequality, we reason

$$\begin{aligned}
&|c_i, c_{COG}| + \frac{8\epsilon_0}{3}(|c_i, c_j| + |c_i, c_k|) \geq |c_k, c_{COG}| \\
&\Rightarrow |c_i, (c_i + c_j)/2| + |(c_i + c_j)/2, c_{COG}| + \frac{8\epsilon_i}{3}(|c_i, c_j| + |c_i, (c_i + c_j)/2| + |(c_i + c_j)/2, c_k|) \geq |c_k, c_{COG}| \\
&\Rightarrow |c_i, c_j|/2 + |c_k, c_{COG}|/2 + \frac{8\epsilon_0}{3}((3/2)|c_i, c_j| + (3/2)|c_{COG}, c_k|) \geq |c_k, c_{COG}| \\
&\Rightarrow (1/2 + 4\epsilon_0)|c_i, c_j| \geq (1/2 - 4\epsilon_0)|c_{COG}, c_k| \\
&\Rightarrow \frac{8\epsilon_0 + 1}{1 - 8\epsilon_0}|c_i, c_j| \geq |c_{COG}, c_k|
\end{aligned}$$

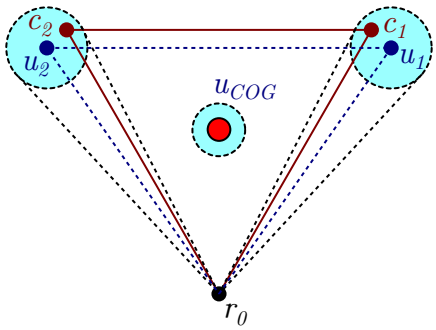


Figure 27: Example illustrating a case where two robots, while not observed as equidistant from the COG, given their uncertainty regions, could in reality actually be forming an equilateral triangle.

B Previous attempt at Section 3.3 (algorithm deadlocks)

This appendix documents a previous attempt at Section 3.3, where we attempted to preserve robot behavior from Algorithm 1. However, this turned out to be quite difficult, and with robots displaying erratic behavior and deadlocks, despite attempts to regulate this type of behavior. Instead, we chose to allow for robot behavior to diverge from the original algorithm, and analysed the resulting situation.

Equidistant from the COG We first attempted to directly adapt the check for the robots forming an equilateral triangle. The first branch of Algorithm 1, guarded by the expression $|c^*_{\max}, c^*_{COG}| = |c^*_{\min}, c^*_{COG}|$, covers the case where all robots are equidistant from the COG, also coinciding with the case where robots form an equilateral triangle (see Lemma 9). As explained in Section 3.1, it is very important to treat this case, as it presents a symmetry that must be broken by exploiting the possible choices of the adversary in which robots to activate during a round. If a *false-negative* is possible due to measurement errors, the adversary can exploit this to cause the system to end up in an infinite loop.

Hence, drawing upon Lemma 12, we must rewrite the case guard to cover the *possibility* that $|c^*_{\max}, c^*_{COG}| = |c^*_{\min}, c^*_{COG}|$ as

$$|u^*_{\max}, u^*_{COG}| \leq |u^*_{\min}, c^*_{COG}| + \delta_1$$

where $delta_1$ is defined in Lemma 12.

Rewritten as such, even under the worst measurement errors, the observer r_0 will never fail to detect whether an equilateral triangle has been formed by the robots.

Preserving the behavior of Algorithm 1, we decided to still have the robot move to where it perceives the COG. More precisely, r_0 will move to s_{COG}, which Lemma 7 tells us is a point at most distance $\epsilon_0|c_0, c_{COG}|$ away from the actual COG.

A natural consequence of adding such thresholds is that the check will unavoidably be vulnerable to *false positives*, see Figure 27.

So far, we therefore had distinguished two cases:

- The active robot believes that all robots may be equidistant to the COG. This corresponds to the **if** clause on line 4 in Algorithm 1.

- The active robot knows that robots cannot all be equidistant to the COG. This corresponds to the **else** clause on line 6 of the same algorithm.

Further from the COG In Algorithm 1, once we are certain that robots are not equidistant to the COG, we find another check, with the guard: $|c_0^*, c_{COG}^*| > |c_{\min}^*, c_{COG}^*|$. Robots where this check passes will move to a robot that is at the minimum distance to the COG. The latter robot must therefore, obviously, not move.

As a result, rewriting the check as $|u_0^*, u_{COG}^*| > |u_{\min}^*, u_{COG}^*|$ is not good enough. Firstly because, again, it is possible that this ordering of distances does not translate to the actual configuration, something we could remedy by adding a threshold as follows: $|u_0^*, u_{COG}^*| > |u_{\min}^*, u_{COG}^*| + \delta_1$.

However, this is not quite sufficient because, whereas r_0 can know for sure when it is actually further from the COG than $r_{\min}$, the robot $r_{\min}$ may be active and may have received a different snapshot than r_0. Consequently, $r_{\min}$ may falsely believe that it is possible for the robots to be forming an equilateral triangle, and decide to move to the measured position of COG. However, this is not guaranteed. Hence, before r_0 can move to $r_{\min}$, it must ensure that $r_{\min}$ will not suffer from such a false-positive. For this purpose, the following lemma is introduced:

The proposal for the case guard for moving to $r_{\min}$ is thus as follows:

$$\textbf{if } |u_0^*, u_{COG,0}^*| > |u_{\min}^*, u_{COG,0}^*| + \delta_2 \textbf{ then}.$$

Ambiguity However, this now leaves a gap: what if r_0 knows it is further from the COG than $r_{\min}$, but there is still some ambiguity left as to whether $r_{\min}$ knows this, and that $r_{\min}$ has not been mislabeled. This ambiguity is illustrated in Figure 28.

More formally, the case where an active robot observes the following proposition as true:

$$|u_{\min}^*, u_{COG}^*| + \delta_1 < |u_0^*, u_{COG}^*| \leq |u_{\min}^*, u_{COG}^*| + \delta_2.$$

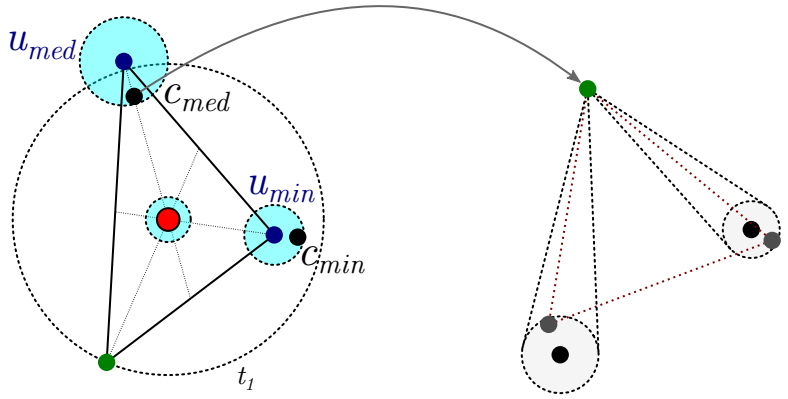


Figure 28: Illustration of how non-consensus can exist in the configuration. Given the uncertainty regions as observed by r_0, it is not possible for the robots to be arranged in an equilateral configuration. However, it *is* possible for robots to be arranged such that, when observed by the robot at the top left, the configuration looks as if it could, in fact, be equilateral.

An initial proposal would be that r_0 does one of two things:

- Robot r_0 chooses not to move. However, in this case, $r_{\min}$ may be waiting for r_0 to move towards it. Hence, a deadlock could occur.

- Robot r_0 moves to s_{COG}. However, this would make it possible that one robot approximately moves to the COG, while another moves to the position of the third robot. This significantly diverges from Algorithm 1.

Instead, we wanted to preserve the behavior of the robots under Algorithm 1 as much as possible, since this allows us to use the original analysis as a starting point for the analysis of the adapted version. We attempted to accomplish this by introducing the concepts of *main* and *ambiguous* cases, as well as a heuristic for dealing with them.

Definition 13. A *main* case is a case observed by an individual robot that maps to one of the branches of the original algorithm.

Definition 14. An *ambiguous* case is a case observed by an individual robot that maps to a set of snapshots that mark an ambiguous boundary between two or more main cases caused by measurement errors. When observing such a case, a robot may be certain about its own situation, but is uncertain about the state of consensus among robots.

For an example of these, in the context of making Algorithm 1 robust, a *main* case would be where a robot r_i observes itself as further away from the COG than another robot by such a margin that, no matter what the adversary does, every other robot must agree that r_i is indeed further out (but not by how much). Conversely, an ambiguous case would be where such an observer knows that it is further from the COG than some other robot, but the margin is small enough as to be lost after considering the different snapshots that other robots may receive about the situation.

In the analysis of the algorithm that has been made robust, these *main* cases can be used as a starting point to adapt the logic from one proof to the next, as we shall see in the proof of Theorem 4.

We then formulated a general heuristic to deal with these: a robot that observes an *ambiguous* case shall not make large movements to avoid interfering with other robots that do observe a *main* case in the current round. It must, however, ensure that at least one robot eventually observes a *main* case to avoid deadlocks. If necessary, the robot may perform a *disambiguation movement* to exit the ambiguous situation; such a movement will typically be designed to be easily dealt with in the analysis of the *main* cases.

Since r_0 not moving would induce a deadlock, we were left with the option of a disambiguation movement. We have two more-or-less equivalent possible choices here:

- Move towards the COG to induce a situation where robots are equidistant from the COG. Unfortunately, hitting this target is difficult without using an approximate definition of equidistance, complicating analysis, so this approach was used here.
- Move away from the COG in order to, should no other robot move, achieve the δ_2 threshold on the next round. This is the approach we will use here.

We defined such a "step away" movement as a movement by vector $a(s_0^* - s_{COG}^*)$, where $a > 0$ would be some scalar function of ϵ_0, large enough to ensure that whichever robot made such a movement would observe itself as further away from the COG by threshold δ_2.

Furthermore, to simplify future analysis and prevent the current robot from moving if another robot may move to the current robot, we added the condition $|u_0^*, u_{COG}^*| + \delta_1 \geq |u_{\max}^*, u_{COG}^*|$. Using threshold δ_1, as established in Lemma 12, this condition would prevent the *step away* movement if another robot was clearly further away from the COG than the observer. Since this condition cannot be true for the robot furthest from the COG, a deadlock is avoided.

All things put together, we obtain Algorithm 5.

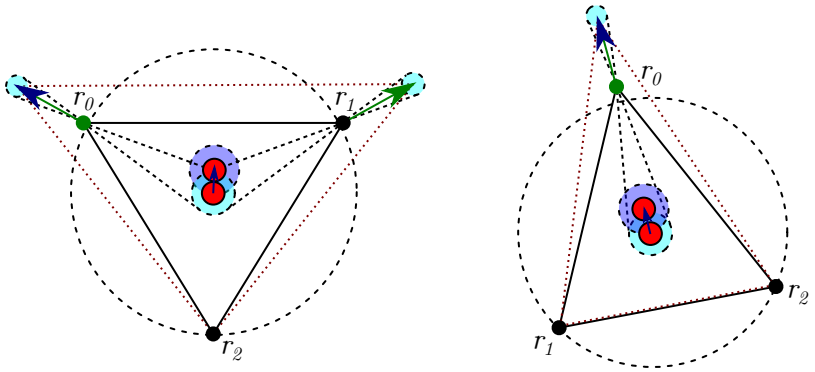


Figure 29: Illustration of a *step away* movement, where one or both robots move away from the COG by a small step. Notice how the triangle becomes more "flat" as robots move away from the COG, and how movement by one robot complements, rather than interferes with the other.

Input: Positions s_1^*, s_2^* in local frame of reference of robot r_0.
Output: Movement vector in the local frame of reference of robot r_0.

$s_0^* \leftarrow (0,0)$
$u_0^* \leftarrow s_0^*$
$u_1^* \leftarrow (s_1^* - s_0^*)/(1-\epsilon_0^2) + s_0^*$
$u_2^* \leftarrow (s_2^* - s_0^*)/(1-\epsilon_0^2) + s_0^*$
$\delta_1 \leftarrow 4\epsilon_0/3(|u_0,u_1| + |u_0,u_2|)$
$\delta_2 \leftarrow \left(\frac{4\epsilon_0}{3} + \frac{16\epsilon_0(1+\epsilon_0)}{3(1-\epsilon_0)}\right)(|u_0,u_1| + |u_0,u_2|)$
$a \leftarrow \frac{12\epsilon_0^2+16\epsilon_0}{1-12\epsilon_0^3-52\epsilon_0^2-55\epsilon_0}$
$s_{COG}^* \leftarrow (s_0^* + s_1^* + s_2^*)/3$
$u_{COG}^* \leftarrow (u_0^* + u_1^* + u_2^*)/3$
Let $s_{\min}^*, s_{\text{med}}^*, s_{\max}^*$ be s_0^*, s_1^*, s_2^* sorted by distance to s_{COG}^*.
Let $u_{\min}^*, u_{\text{med}}^*, u_{\max}^*$ be u_0^*, u_1^*, u_2^* sorted by distance to u_{COG}^*.
if $|u_{\max}^*, u_{COG}^*| \leq |u_{\min}^*, u_{COG}^*| + \delta_1$ **then**
    move to s_{COG}^*
else
    if $|u_{\min}^*, u_{COG}^*| + \delta_2 < |u_0^*, u_{COG}^*|$ **then**
        move to $s_{\min}^*$
    else if $|u_{\min}^*, u_{COG}^*| + \delta_1 < |u_0^*, u_{COG}^*|$ **and** $|u_0^*, u_{COG}^*| + \delta_1 \geq |u_{\max}^*, u_{COG}^*|$ **then**
        move by vector $a(s_0^* - s_{COG}^*)$
    else
        stay in place
    end if
end if

Algorithm 5: Algorithm for Approximate Gathering under SSync for $n = 3$. Notice the structural similarities to Algorithm 1, and that points c_0^*, s_0^* and u_0^* are all simply $(0,0)$, though we will keep the separate notations to make explanations easier. Also, note that the distance order of the s^*-points and u^*-points is the same, since they are simply scalings centered on (0,0).

Unfortunately, in this version, Algorithm 5 suffers from a deadlocking condition. Indeed, note that $|u_{\max}^*, u_{COG}^*| \leq |u_{\min}^*, u_{COG}^*| + \delta_1$ only requires that *some* robot appears further from the COG than some other robot, but does not specify which.

As a result, suppose that robots r_i, r_j, r_k are positioned such that the formation approaches an equilateral triangle, with r_i and r_j approximately equidistant from the COG and r_k closest to the COG. Now, suppose that the adversary provides a snapshot to r_i showing that r_j is certainly further from the COG, but suggests that r_i may be closest to the COG. It provides a similar snapshot to r_j, though showing r_i as further out instead of r_j. Both r_i and r_j will thus observe *maybe equilateral* and stay in place, inducing a deadlock.