Title	Fault-Resilient Cooperation of Autonomous Mobile Robots with Unreliable Compass Sensors
Author(s)	Souissi, Samia
Citation	
Issue Date	2007-09
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/3754
Rights	
Description	Supervisor:Defago Xavier, 情報科学研究科, 博士

Fault-Resilient Cooperation of Autonomous Mobile Robots with Unreliable Compass Sensors

by

Samia SOUISSI

submitted to
Japan Advanced Institute of Science and Technology
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Supervisor: Associate Professor Xavier DÉFAGO

School of Information Science
Japan Advanced Institute of Science and Technology

September 2007

Title: Fault-Resilient Cooperation of Autonomous Mobile Robots with Unreliable Compass Sensors

Author: Samia SOUISSI

Japan Advanced Institute of Science and Technology

School: Information Science

Department: Information Systems

Convocation: September 2007

Degree: PhD.

External Examiners: Prof. Masafumi YAMASHITA,

Prof. Koichi WADA,

Associate Prof. Maria GRADINARIU.

Internal Examiners: Prof. Takuya KATAYAMA,

Prof. Tetsuo ASANO,

Associate Prof. Toshiaki AOKI.

Research Supervisor: Associate Prof. Xavier DÉFAGO

Abstract

Continuous advances in technology have made possible the use of several robots in order to carry out a large variety of cooperative tasks that are dangerous or undesirable for humans to complete. These tasks include, surveillance, inspection of sites that are inaccessible to humans, e.g., tight spaces, hazardous environments or remote sites, and search and rescue tasks, such as rescuing human beings trapped under piles of debris in an earthquake disaster or searching for victims of a flood. Following this idea, we are interested in systems with no prior infrastructure (e.g., unlike Global Positioning System), where robots are deployed in adverse environments, and where they are required to cooperate and self-organize to build such an initial infrastructure. For instance, robots may need to exchange information on their states (positions, trajectories, orientation, etc.) to construct a complete configuration of the team in order to cooperate. However, robots may not initially agree on a common coordinate system. Therefore, providing a way for robots to agree on a common coordinate system is useful in exchanging geographical information, for instance. Subsequently, reaching agreement among these robots is one of the most essential issues in distributed robotic systems. Besides, as the number of robots increases in the system, the issue of resilience to failure becomes prominent.

In this dissertation, we consider a system that consists of a group of mobile robots roaming in the two-dimensional plane. Each robot occupies a point in the plane, and is equipped with sensors to observe the positions of the other robots. Each robot proceeds by repeatedly (1) observing the environment, (2) computing a destination based on the observed positions of robots, and (3) moving toward the computed destination. Also, robots are unable to communicate directly, and can only interact by observing each others' positions. Finally, all robots execute the same deterministic algorithm, and they are oblivious (i.e., stateless), meaning that they can not remember their previous states, their previous actions or the previous positions of the other robots.

In this model, we address the problem of coordination between these robots from a computational viewpoint, aiming to identify the fundamental limits of what autonomous mobile robots can do in the presence of unreliable sensors. In particular, we focus on a basic coordination problem, namely the gathering problem, where robots must self-organize, and meet at some location not determined in advance, and without the help of some global coordinate system. While being very simple to express, this problem has the advantage of retaining the inherent difficulty of agreement, namely the question of breaking symmetry between robots. Among other things, solving the gathering problem is interesting because it provides a way for robots to agree on a common origin.

Prior work has shown that gathering oblivious mobile robots in the plane cannot be achieved deterministically without additional assumptions. More specifically, if robots can detect multiplicity (i.e., count robots that share the same location), gathering is possible

for three or more robots. Alternatively, prior work has also shown that gathering can be achieved with any number of robots if they share the knowledge of a common direction (e.g., north as given by some compass). However, that result holds only if all compasses are perfect, in the sense that they all agree perfectly on a common direction.

In this dissertation, we define a model in which compasses may be unreliable, and we study the solvability of gathering oblivious mobile robots using different classes of unreliable compasses, and under different models of synchrony. More specifically, we describe two classes of unreliable compasses, namely the class of eventually consistent compasses, and the class of compasses with bounded errors. Then, we present several results of possibility and impossibility for solving the gathering problem deterministically under these classes.

This dissertation makes four major contributions:

In the first contribution, we address the problem of gathering with eventually consistent compasses, that is compasses that are unstable for some unknown periods, with the guarantee that they stabilize eventually to show the correct direction. However, the time of stabilization is unknown to robots. In particular, we address the problem in the semi-synchronous model, where robots are oblivious and they have limited visibility. Especially, we provide an algorithm that solves the problem, in finite time, in a system where compasses are unstable for some arbitrary long periods, provided that they stabilize eventually.

The algorithm can solve gathering probabilistically when the compasses are inconsistent, and deterministically after compasses have stabilized for a sufficiently long period. Our algorithm is guaranteed to recover from any arbitrary configuration when the compasses of robots eventually stabilize. We can argue that our algorithm is intrinsically self-stabilizing and offers protection against any number of transient failures in the compasses.

In the second contribution, we study the solvability of gathering in the asynchronous model under eventually consistent compasses, where robots are oblivious and have limited visibility. In particular, we propose an algorithm that gathers up to four robots, in finite time, relying on eventually stabilizing compasses. In addition, we show that our gathering algorithm developed for the semi-synchronous model solves gathering for up to three robots, when they are equipped with eventually consistent compasses.

Alternatively, we show that it is impossible to achieve the gathering of a large number of asynchronous mobile robots, when they are equipped with compasses that are unstable for some arbitrary periods, they are oblivious and have limited visibility. In particular, we show that there exists no oblivious algorithm that solves the gathering problem for nine or more robots.

In the third contribution, we focus on the solvability of the gathering of autonomous mobile robots with inaccurate compasses. In particular, we provide a self-stabilizing

algorithm to gather, in finite time, two asynchronous oblivious robots equipped with compasses that can differ by as much as 45°. In addition, we argue that our algorithm is also correct if we consider robots with volume, that is, robots are not represented by points, but they occupy some space in the plane.

In the fourth contribution, we extend our work on gathering with inaccurate compasses by proving a tight bound on the degree of divergence of robots' compasses for solving the gathering problem. More specifically, we present a self-stabilizing algorithm to gather, in a finite time, two oblivious robots equipped with compasses that can differ by an angle strictly less than 180°, and this is obviously a tight bound, since two compasses that can differ by an angle of up to 180° provide no information at all.

Finally, as a secondary contribution, we have completed a prior work on the circle formation problem by developing complete and rigorous proofs of correctness of a previous distributed circle formation algorithm. In particular, we studied the problem when robots are disoriented, i.e., share no knowledge of a common coordinate system, and they are oblivious. The algorithm allows a group of mobile robots to self-organize and move to form a circle in the semi-synchronous model. The proposed algorithm ensures that robots deterministically form a circle in a finite number of steps, and converges to a situation in which all robots are located evenly on the boundary of the circle. Among other things, the ability to form a circle means that the robots are able to agree on an origin and unit distance.

Keywords: Distributed computing, Mobile computing, Autonomous robots, Distributed algorithms, Cooperation, Control, Implicit communication, Gathering, Point formation, Rendez-vous, Circle formation, Fault-tolerance, Unreliable compasses, Unstable compasses, Eventually consistent compasses, Inaccurate compasses, Bounded error compasses, Self-stabilization.

Résumé

L'évolution continue de la technologie a fait possible l'utilisation de plusieurs robots pour effectuer une grande variété de tâches coopératives qui sont dangereux ou indésirables aux humains pour les faire. Ces tâches incluent, la surveillance, l'inspection des sites qui sont inaccessibles aux humains tels que, les espaces serrés, et des tâches de sauvetage, telles que secourir des humains piégé sous des piles de débris à cause d'une catastrophe du tremblement de terre ou chercher des victimes d'une inondation. En suivant cette idée, nous nous intéressons aux systèmes sans infrastructure initiale (contrairement au Système du Positionnement Global), où les robots sont déployés dans des environnements adversaires, et ils doivent coopérer et s'organiser pour construire une telle infrastructure initiale. Par exemple, les robots ont besoin d'échanger d'information sur leurs états (places, trajectoires, orientation, etc.) pour construire une configuration complète de l'équipe afin de coopérer. Cependant, les robots ne peuvent pas initialement se mettre d'accord sur un référentiel commun de coordonnées. Par conséquent, fournir un moyen pour que ces robots puissent se mettre d'accord sur un référentiel commun de coordonnées est utile pour échanger de l'information géographique, par exemple. Donc, aboutir à un accord entre les robots est une des questions les plus essentielles dans les systèmes de robotique distribués. De plus, la question de résilience aux pannes devient proéminente lorsque le nombre de robots dans le système augmente.

Dans cette thèse, nous considérons un système qui consiste d'un groupe de robots mobiles qui peuvent se déplacer dans le plan. Un robot est considéré comme un point dans le plan, et il est équipé par des détecteurs pour observer les positions des autres robots. Chaque robot procède en répétant les étapes suivantes: (1) observer l'environnement, (2) calculer une destination basé sur les positions observées des autres robots, et (3) se déplacer vers la destination calculée. De plus, les robots sont incapables de communiquer directement, et ils peuvent interagir seulement en observant les positions des autres robots. Finalement, tous les robots exécutent le même algorithme d'une façon déterministe, et ils sont oublieux (c-à-d., sans-états), cela veut dire qu'ils ne peuvent pas se souvenir de leurs états précédents, leurs actions antérieures ou les positions précédentes des autres robots.

Dans ce modèle, nous abordons le problème de coordination entre ces robots de point de vue calculatrice, dans le but d'identifier les limites fondamentales de coopération entre les robots mobiles autonomes en présence de détecteurs incertains. En particulier, nous focalisons sur un problème de coordination de base, qui est le problème de rassemblement, où les robots doivent s'organiser pour se réunir finalement à un emplacement arbitraire qui n'est pas déterminé à priori, sans accord sur un système de coordonnées global. Bien que ce problème aie l'avantage d'être très simple à exprimer, il retient néanmoins la difficulté inhérente aux problèmes d'accord repartis, à savoir, le problème de briser la symétrie entre les robots. De plus, résoudre le problème de rassemblement est intéressant parce

qu'il permet aux robots de se mettre d'accord sur une origine commune.

Dans les travaux précédents, il a été prouvé que le problème du rassemblement de robots ne peut pas être résolu d'une façon déterministe sans hypothèses supplémentaires. En particulier, si les robots peuvent détecter la multiplicité (c-à-d., compter le nombre de robots partageant le même endroit), le problème du rassemblement est résolu pour au moins trois robots.

De la même façon, le problème peut être résolu de façon déterministe dans un modèle asynchrone avec des robots possédant une visibilité limitée, s'ils partagent la connaissance d'une direction de référence, (par exemple, le nord fournie par une boussole). Cependant, le résultat est correct seulement si tous les compas sont parfaits, dans le sens qu'ils sont tous d'accord parfaitement sur une direction commune.

Dans cette thèse, nous définissons un modèle dans lequel les boussoles peuvent être incertaines, et nous étudions la solvabilité du problème de rassemblement des robots mobiles oublieux dans différentes classes de compas incertains, et sous des modèles différents de synchronie. Plus précisément, nous décrivons deux classes de boussoles incohérentes, la classe des boussoles instables, et la classe des boussoles avec des erreurs bornées. Alors, nous présentons plusieurs résultats de possibilité et d'impossibilité pour résoudre le problème de rassemblement d'une façon déterministe dans ces classes de boussoles.

Cette thèse a quatre contributions essentielles:

Dans une première contribution, nous adressons le problème de rassemblement avec des boussoles instables, dans le modèle semi-synchrone avec des robots oublieux et possédant une visibilité limitée. En particulier, nous donnons un algorithme qui fonctionne avec des boussoles instables, pour autant que celles-ci passent de temps en temps par des périodes de stabilité. La difficulté réside dans le fait qu'il est impossible pour les robots de déterminer si les boussoles sont dans des états stables.

L'algorithme peut résoudre le problème du rassemblement d'une manière probabiliste quand les boussoles sont inconsistantes et d'une manière déterministe après que les boussoles se stabilisent pour assez longtemps. Notre algorithme garantit le rétablissement à partir d'une configuration arbitraire quand les boussoles des robots se stabilisent après un certain temps. Nous pouvons monter que notre algorithme est intrinsèquement auto-stable et qu'il offre une protection contre les pannes transitoires des boussoles.

Dans une deuxième contribution, nous étudions la solvabilité du problème de rassemblement dans le modèle asynchrone avec des robots oublieux et possédant des boussoles instables et une visibilité limitée. En particulier, nous proposons un algorithme qui rassemble, dans un temps fini, au plus quatre robots qui sont équipés par des boussoles qui se stabilisent après un certain temps. Nous montrons aussi que notre algorithme de rassemblement développé pour le modèle semi-synchrone résout le problème de rassemblement pour trois robots au plus, dans le modèle asynchrone, quand les robots sont équipés par des boussoles instables et qui seront consistantes après un certain temps. Par contre, nous

montrons que c'est impossible de résoudre le problème de rassemblement pour neuf ou plus de robots mobiles asynchrones, quand ils sont équipés par des boussoles instables pendant certaines périodes arbitraires.

Dans une troisième contribution, nous abordons la solvabilité du problème de rassemblement de robots mobiles autonomes avec les boussoles inexactes. En particulier, nous donnons un algorithme auto-stable qui permet de rassembler, dans un temps fini, deux robots oublieux asynchrones qui sont équipés par des boussoles décalées de 45°. De plus, nous montrons que notre algorithme est aussi correct si nous considérons les volumes des robots, c.-à-d., les robots ne sont pas représentés par des points, mais ils occupent un espace dans le plan.

Dans une quatrième contribution, nous étendons notre travail concernant le problème de rassemblement avec des boussoles inexactes en déterminant une borne stricte sur le degré de divergence de boussoles des robots pour résoudre le problème de rassemblement. Plus spécifiquement, nous présentons un algorithme auto-stable pour rassembler, dans un temps fini, deux robots oublieux qui sont équipés par des boussoles décalées par un angle strictement inférieur à 180°, qui représente une borne stricte puisque deux boussoles décalées par un angle de 180° ne donnent pas d'information.

Finalement, nous avons complété un travail antérieur sur le problème de la formation d'un cercle, en développant des preuves complètes et rigoureuses d'exactitude d'un algorithme antérieur de formation de cercle comme étant une contribution secondaire. En particulier, nous avons étudié le problème quand les robots n'ont pas un référentiel commun de coordonnées, et ils sont oublieux. L'algorithme permet à un groupe de robots mobiles de se déplacer pour former un cercle dans le modèle semi-synchrone. L'algorithme proposé assure que les robots forment un cercle dans un temps fini, et converge vers une situation dans laquelle tous les robots sont distribués uniformément sur la circonférence du cercle. Un des avantages de la formation d'un cercle, c'est qu'elle permet aux robots de se mettre d'accord sur une origine et sur une unité de distance.

Les Mots Clés: Calcul distribué, Calcul mobile, Robots autonomes, Algorithmes distribués, Coopération, Contrôle, Communication implicite, Tolérance aux pannes, Problème d'assemblement, Formation d'un point, Rendez-vous, Formation d'un cercle, Boussoles non fiables, Boussoles instables, Boussoles inexactes, Algorithmes auto-stables.

To my mother, my father,
my brothers and my sisters
for their everlasting love,
support and encouragement.

Acknowledgments

First of all, I would like to express my deep gratitude and appreciation to my supervisor Prof. Xavier Défago for supervising my work with great care and enthusiasm, for his constant guidance and support, and for expressing his confidence in me. Also, I would like to thank him for giving me many opportunities to meet with other people in the same research field.

I would like also to express my sincere thanks to my principal advisor Prof. Takuya Katayama for his support, kind supervision and encouragement during my work.

I wish also to extend my great thanks to Prof. Masafumi Yamashita, for valuable scientific discussions, and his helpful collaboration and support during my research.

I would like to thank all the members of the jury, Prof. Masafumi Yamashita (from Kyushu University), Prof. Koichi Wada (from Nagoya Institute of Technology), Prof. Maria Gradinariu (from Université Paris 6), Prof. Tetsuo Asano (from JAIST), Prof. Takuya Katayama (from JAIST), and Prof. Toshiaki Aoki (from JAIST) for their efforts and the time they have spent examining this thesis.

I am very grateful to Prof. Mary Ann Mooradian from JAIST for all the time and effort that she spent editing the English of this thesis, and for her sympathy.

I am also grateful and pleased to meet with the group from Nagoya Institute of Technology, Prof. Koichi Wada, Prof. Yoshiaki Katayama, Prof. Nobuhiro Inuzuka, and Prof. Taisuke Izumi for their valuable suggestions.

My special thanks go to my colleague Dr. Rami Yared for his crucial help, and for his support during my research and for his nice friendship. Also, I am grateful to my previous colleagues Matthias Wiesmann and Péter Urbán from Google Switzerland for their insightful comments regarding my work.

I gratefully recognize all of my colleagues in the laboratory, and especially Hiroaki Tanizaki for his kind help in translation to Japanese, and the secretaries Misato Morita, Miyuki Sakurai, Miki Kondo, and Ayako Fukuoka for their crucial help especially in ad-

ministrative procedures, and for their nice friendship.

A special word of gratitude goes to my fiancé Dr. Nafaâ Jabeur from the University of Windsor Canada for his continuous support, encouragement and lovely friendship.

I wish also to devote my sincere thanks to my wonderful friend Dr. Ahlem Ben Hassine for her deep love, great support and for the unforgettable moments that we had together in Japan. Also, I would like to thank her husband Sami Ben Ismail for his support and his nice friendship.

Very special thanks go also to my wonderful friend Kaouthar Samarat for the very special moments that we had together in Japan, as well as for being a great support for me during the very difficult moments, without her support, and continuous encouragement, I would probably not have been able to finish my doctorate. Also, I am grateful to her husband Mouez Lassouad for his support and kind advice. Special words of gratitude goes also to their cute daughter Balkiss who was born with this thesis.

I would like also to thank Dr. Yasser Kotb for his kind help and support during my first days at JAIST. Also, I am very grateful to Dr. Adel Cherif from the University of Qatar who recommended JAIST to me, and for his advice and kind assistance before and after my entering JAIST.

A special thanks goes also to my friend Dr. Saida Benlarbi from Alcatel, Canada for her kind encouragement and nice friendship. I am also grateful to my friends Dr. Murad Ali from Yemen and his wife Takrid, Dr. Karim from Algeria and his wife Karima, Dr. Shafique Ansari from India and his wife Dr. Zubaida Ansari, and Dr. Nebil Achour and his wife Caroline Deegean for their hospitality, and their warm welcome.

I also want to devote my thanks to all my professors at the Japan Advanced Institute of Science and Technology, and to my previous professors and friends in Tunisia.

I should also mention that my master and PhD studies in the Japan Advanced Institute of Science and Technology were supported by the Japanese government scholarship (MONBUKAGAKUSHO).

Last, but definitely not least, I am forever grateful and in dept to my parents for their loving support throughout my life. My thanks extend to my brothers and their wives, my sisters and their husbands, my nephews, especially Yamen who was born with this thesis, my nieces, and all my relatives for all their love, support and encouragement.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

A single arrow is easily broken,
but not ten in a bundle.
(Japanese proverb)

1.1 Motivation

Nowadays, the variety of robotic applications is growing at a tremendous rate, and the trend will continue in the future as progress in technology opens new possibilities in applications. Recently, robotic design has been focused on making systems involving a large number of robots working together. More specifically, interest has shifted from the design and deployment of few, rather complex and expensive robots towards the design and use of a large number of robots, which are simple, relatively inexpensive, but capable of collaborating to perform complex tasks. Several reasons motivate this shift, including reduced costs, faster computation, more potential for fault tolerance, the possibility of extendability of the system and the reusability of the robots in different applications.

A large range of applications benefit from this shift, particularly applications in dangerous environments, where human lives would be jeopardized, and applications in remote places, either inaccessible to humans or where communication delays render remote controlled robot missions unfeasible. Examples include planetary rover missions [72], Mars ground preparation [63], surveillance and inspection of remote sites [44], such as tight spaces, deep sea, hazardous and hostile environments, search and rescue missions [40], such as rescuing people trapped under piles of debris in an earthquake diaster, or searching for victims of a flood, exploration of unknown environments [11, 61], cooperating autonomous vehicles [6], etc.

Distributing these robots raises a range of issues, such as how to efficiently and robustly coordinate their activities in a distributed fashion so that they can cooperate. Consequently, part of the focus of the research community has been on how to coordinate such simple mobile robots so that they can together perform some given common task. Studies can be found in different disciplines, from artificial intelligence to engineering

(e.g., [5, 7, 43, 50]; see detailed discussion in Section 2.1).

One common feature of the majority of these studies has been the development of algorithms based on heuristics, such as from studying how a complex global behavior can emerge from the interactions of many robots exhibiting a simple local behavior [5, 9, 45, 47]. Although attractive for the average case, the proposed approaches have not proved correct in all possible situations. In contrast, we approach the problem from an algorithmic perspective, and we aim to explore the limits of provably correct algorithmic solutions in this field. For instance, being able to reach an agreement between a group of mobile robots (e.g., agreement on a common coordinate system) is a key to ensure their cohesion. To understand these issues properly, they must be studied rigorously, from the perspective of distributed algorithms. This is motivated by the needs of systems that can be justifiably trusted to correctly perform what they were intended to do (dependable systems).

Recently, this approach is gaining momentum (e.g., [32, 33, 54, 69, 82]), however many interesting questions remain. In particular, in a distributed system, as the number of robots increases, different types of faults can occur in the system. However, the issue of resilience to failure in multiple robot systems is still a largely unexplored direction, which our research aims at investigating.

For instance, a fault can occur due to the crash or a temporary misbehavior of a robot, due to some external factors, or due to the unreliability of its sensors. Also, the failure can be due to malicious intruders. Consequently, algorithms must be designed so as to deal properly with such failures.

Surprisingly, in the literature, most of the algorithmic results we are aware of rely on the assumption that robots function properly, that their sensors provide perfect information, and they are accurate. Nevertheless, in practise, sensors in general are prone to failures and are subject to instabilities and inaccuracies. For instance, a compass sensor is sensitive to magnetic interference.

Consequently, the main motivation of this work is to systematically investigate the fundamental limits of coordination between multiple mobile robots, and to provide robust distributed algorithms for reaching agreement among a group of mobile robots when their sensors exhibit faulty behavior.

1.2 Problem Statement

In this dissertation, we study the problem of coordination of a group of weak mobile robots in a totally distributed fashion, where their sensors are *unreliable*, and from an algorithmic perspective. In particular, we investigate the impact of the unreliability of sensors on the solvability of coordination problems by multiple mobile robots.

The system consists of a group of mobile robots roaming in the two-dimensional plane.

Each robot occupies a point in the plane, and it is equipped with sensors to observe the positions of the other robots. In particular, each robot proceeds by repeatedly (1) observing the environment, (2) computing a destination based on the observed positions of robots, and (3) moving toward the computed destination. Also, robots are unable to communicate directly, and can only interact by observing each others' positions. Finally, all robots execute the same deterministic algorithm, and they are oblivious (or memoryless), meaning that they cannot remember their previous states, their previous actions or the previous positions of the other robots. While this is a somewhat over-restrictive assumption, developing algorithms in this model is useful both for memory management, and because any algorithm that works correctly for oblivious robots is intrinsically self-stabilizing [69].[1] Also, robots are equipped with compass sensors that can be subject to instabilities and inaccuracies.

Unlike most work that has been done in the literature, we address the problem of coordination between robots when they are equipped with sensors that are *unreliable*. That is, robots' sensors are liable to be erroneous or misleading, and can not be trusted.

In particular, we focus on an agreement problem called gathering (also known as rendez-vous or point formation) in a system with no agreement on a global coordinate system, and in the absence of any landmarks in the environment. In short, the problem requires that the robots, initially located at random locations, move in such a way that they eventually end up at the exact same location, not determined a priori. The algorithm must ensure that the final configuration is obtained within a finite number of steps, from any initial situation and in every possible execution. Solving the gathering problem means that the robots have the inherent ability to agree on a common origin.

Prior work has shown that gathering oblivious mobile robots in the plane cannot be achieved deterministically without additional assumptions [59]. More specifically, if robots can detect multiplicity (i.e., count robots that share the same location), gathering is possible for three or more robots [17]. Alternatively, prior work has also shown that gathering can be achieved with any number of robots if the robots share the knowledge of a common direction (e.g, North as given by some compass) [34]. However, the result holds only if all compasses are perfect, in the sense that they all agree perfectly on a common direction. Yet, in practice, sensors are error-prone and sensitive to magnetic interferences. Consequently, in this thesis, we concentrate on the gathering of oblivious mobile robots when their compasses are subject to instabilities and inaccuracies. Our work is motivated by the facts that: (1) in practice, compasses are rather inaccurate sensors. For example, the accuracy of compasses typically varies from 1 degree to over 10 degrees, depending on sensor quality (cost) and environment conditions. (2) compasses are sensitive to magnetic interferences, such as metal objects, belt buckles, boats, automobiles,

[1] Self-stabilization is the property of a system which, starting in an arbitrary state, always converges toward a desired behavior [27, 62].

etc. (3) with multiplicity detection, the gathering is solvable only for more than two robots.

Particularly, we define a model in which compasses may be unreliable, and we study the solvability of gathering oblivious mobile robots with different classes of unreliable compasses, and under different models of synchrony. More specifically, we describe two classes of unreliable compasses, namely the class of eventually consistent compass, and the class of compasses with bounded errors. Then, we present several results of possibility and impossibility for solving the gathering problem deterministically under these classes.

1.3 Research Contributions

The main contribution of this thesis is to systematically investigate the fundamental limits of coordination and agreement of a group of autonomous mobile robots when their sensors can possibly be subject to failures and inaccuracies. Specifically, we define different classes of unreliable compasses, and we study the solvability of the gathering problem deterministically in the face of different variants of unreliable compasses, under different models of synchrony, and when robots are oblivious. Consequently, this work has ushered in four major contributions to the field of cooperative mobile robotics:

- In the first contribution, we provide a distributed algorithm that solves the gathering problem in the oblivious and limited visibility settings in the semi-synchronous model [69] (refereed as SYm model), when robots are equipped with compasses that are unstable for some periods, called *eventually consistent compasses*. In particular, the proposed solution guarantees that the robots gather at a single point in finite time, if their compasses provide correct output after some unknown period of instability, during which our algorithm can tolerate any number of transient failures of the compasses. The algorithm can solve gathering probabilistically when the compasses are inconsistent, and deterministically after compasses have stabilized for sufficiently long. Besides, our algorithm is intrinsically self-stabilizing (Chapter 4).

- In the second contribution, we propose a distributed algorithm that solves the gathering problem for up to four robots in the asynchronous model Corda [56], when robots are oblivious, they have limited visibility, and assuming eventually consistent compasses. The algorithm guarantees that the robots gather at a single point in finite time when the compasses of the robots eventually stabilize, and probabilistically when the compasses are inconsistent. In addition, we show that our algorithm proposed for the SYm model can solve the gathering of at most three robots in the Corda model, when robots are equipped with eventually consistent compasses, they are oblivious and have limited visibility (Chapter 5).

In contrast, we show that there exists no oblivious algorithm that solves the gathering problem in the CORDA model for a large number of robots when robots are equipped with eventually consistent compasses. In particular, we show that the gathering has no solution for nine or more robots when their compasses are unstable for some arbitrary periods, and they have limited visibility (Chapter 6).

- In the third contribution, we focus on the solvability of the gathering of autonomous mobile robots in the face of compass inaccuracies. In particular, we provide a self-stabilizing algorithm to gather, in finite time, two asynchronous oblivious robots equipped with compasses that can differ by as much as 45°. Besides, we argue that our algorithm is still correct if we consider robots with volume, that is, robots are not represented by points, but they occupy some space in the plane (Chapter 7).

- In the fourth contribution, we extend our work on the gathering with inaccurate compasses, and we present a tight bound on the degree of divergence of robots' compasses in solving the gathering of two asynchronous oblivious mobile robots. In particular, we present an algorithm that solves the problem, in finite time, when robots' compasses are inconsistent by an angle which is strictly less than 180°, and we show that it is a tight bound (Chapter 8).

In addition to the contributions mentioned above, we have also developed complete proofs of correctness for a distributed circle formation algorithm outlined in prior work [64]. The algorithm allows a group of mobile robots to self-organize and position themselves into forming a circle (circle formation problem) in the semi-synchronous model (SYm). In particular, we studied the problem when robots are disoriented, i.e., share no knowledge of a common coordinate system, and they are oblivious. The proposed algorithm ensures that the robots deterministically form a circle in a finite number of steps, and converges to a situation in which all robots are located evenly on the boundary of the circle (Chapter 10).

1.4 Thesis Organization

The thesis is organized as follows:

- Chapter 2 gives a short survey of the relevant related work in the field of distributed and cooperative robotics, and describes the system models, definitions, notations and geometric properties used throughout this thesis.

- Chapter 3 defines different classes of unreliable compasses.

- Chapter 4 presents our gathering algorithm in the SYm model, when robots are equipped with eventually consistent compasses, they are oblivious and have limited visibility.

- Chapter 5 presents an algorithm that solves the gathering problem in the CORDA model for up to four robots relying on eventually consistent compasses, oblivious computations and limited visibility.

- Chapter 6 presents an impossibility proof of gathering a large number of robots in the CORDA model, when robots are equipped with eventually consistent compasses, they are oblivious and have limited visibility.

- Chapter 7 gives an algorithm for solving the gathering of two asynchronous mobile robots when their compasses can diverge by as much 45°.

- Chapter 8 extends the work in the previous chapter by proving a tight bound on the degree of divergence of robots' compasses for solving the gathering problem. More specifically, it presents an algorithm that gathers, in a finite time, two oblivious robots equipped with compasses that can differ by an angle strictly less than 180°.

- Chapter 9 discusses how our results on the gathering with unreliable compasses can be useful in practise.

- Chapter 10 presents a complete and rigorous proof of correctness of a prior circle formation algorithm. The algorithm allows a group of oblivious mobile robots to form a circle in a finite number of steps, and converge toward a configuration, where the robots are placed at regular intervals on the boundary of the circle.

- Chapter 11 presents a discussion on the gathering of robots with volume.

- Chapter 12 summarizes the major results of this work, and outlines future research directions.

Chapter 2

Background

A theory is something nobody believes,
except the person who made it.
An experiment is something everybody
believes, except the person who made it.
Albert Einstein

This chapter presents a survey of the current state-of-the-art in the field of distributed and cooperative robotics. The chapter is divided in five sections. Section 2.1 presents a general overview of the area of multi-robot motion coordination. Section 2.2 presents in particular the computational approach, and work related to the circle formation and gathering problems. Section 2.3 describes the system models used in this thesis. Finally, Section 2.4 introduces some notations and definitions that are used in the remainder of the thesis.

2.1 Multi-robot Cooperation Approaches

There are several approaches to multi-robot motion coordination and control reported in the literature, including centralized and decentralized ones. Uny Cao et al. [70] provided a wide survey of research in this field. Decentralized approaches can be categorized to *heuristic* approaches, also called the *emergent behavior*, or the behavior-based approaches, engineering approaches, and computational approaches.

2.1.1 Emergent Behavior Approach

In *emergent behavior* approach, the goal is to predict what complex global behavior can *emerge* from the interaction of many agents exhibiting a very simple local behavior, for instance, inspired by the behavior of social insects [12, 39, 45, 47].

With this approach, some behaviors are prescribed for each agent, and the final control is derived by weighting the importance of each behavior. Algorithms are designed using mainly heuristics, and the behaviors are programmed in the robots with no explicit goal

programmed in. The cooperation and the goals *emerge* from the interactions between the robots and the environment. The main problem with this approach is the lack of formal proofs of correctness and guarantees of completeness and stability.

An *emergent behavior* approach, is also refereed in the literature as *swarm intelligence*, which is an artificial intelligence technique based on the study of collective behavior in self-organized systems.

A noteworthy study in this field is by Matarić [49, 50], who analyzes how biologically-inspired behaviors can serve as an effective substrate for control, representation, and learning in multi-robot systems, in order to generate adaptive individual and group behavior.

Beni and Wang [8] introduced the expression swarm intelligence in the context of cellular robotic systems, which consists of collection of autonomous, non intelligent robots cooperating in n-dimensional cellular space under distributed control. Limited communication exists only between adjacent robots, which operate autonomously and cooperate with others to accomplish predefined global tasks.

Bonabeau et al. [7, 9] extended the restrictive context of this early work to include any attempt to design algorithms or distributed problem solving devices inspired by the collective behavior of social insect colonies, such as ants, termites, bees, wasps, and other animal societies.

Another study in this area by Brooks [10], examined the relationship of traditional academic robotics and traditional artificial intelligence. The behavior based control strategy has been applied to collections of simple independent robots, usually for simple tasks.

Matarić [48] describes experiments with a homogenous population of robots acting under different communication constraints. The robots either act in ignorance of one another, are informed by one another, or intelligently cooperate with one another. As inter-robot communication improves, more and more complex behaviors are possible.

Dudek et al. [28] presented a swarm-robot taxonomy of the different ways in which swarm robots can be characterized. The dimensions of the taxonomic axes are swarm size, communication range, topology, bandwidth, swarm reconfigurability, unit processing ability, and composition.

Parker [53], defined a behavior-based architecture, called ALLIANCE, which allows teams of robots, each of which possesses a variety of high-level functions that it can perform during a mission, to individually select appropriate actions throughout the mission based on the requirements of the mission, the activities of other robots, the current environmental conditions, and the robot's own internal states. The feasibility of this architecture is demonstrated in an implementation on a team of mobile robots performing a laboratory version of hazardous waste cleanup.

Balch and Arkin [5] studied formation and navigation problems in multi-robot teams. In particular, they presented some formation behaviors that they integrated with other navigational behaviors to enable a robotic team to reach navigational goals, avoid hazards,

and simultaneously remain in formation. The behaviors were implemented in computer simulation, and also on Unmanned Ground Vehicles.

Chen and Luh [15] also studied the coordination of robots in formation, by imposing constraint conditions to coordinate and control the formation. In particular, they decomposed the task of coordinating and controlling a group of small mobile robots into different subtasks: formation of geometric pattern, alignment of each robot's orientation, coordination of the robots in the group, motion realization and stability of the formation in motion.

Saber and Murray [60], studied the flocking problem, where robots are asked to move in formation. In particular, they addressed the problem in the presence of multiple obstacles, and they provided formal definitions of nets and flocks as spatially induced graphs. The task of flocking is based on an automatic construction of an energy function for the groups of robots, and it is achieved via dissipation of this energy according to a protocol that only requires the use of local information.

2.1.2 Engineering Approach

Another kind of approach to the problem of studying multi-robot systems, is that of Kawauchi, Inaba, and Fukuda [43] who have studied a dynamically reconfigurable robotic system, consisting of several cells that can physically detach and combine autonomously according to the assigned task.

In the same perspective is the work of Dumitrescu et al. [29], which examined the problem of dynamic self-reconfiguration of a modular robotic system, to a formation aimed at reaching a specified target position with one of the modules as quickly as possible. Later in their recent work [29], they addressed a number of issues related to motion planning and analysis of rectangular metamorphic robotic systems. In particular, they have shown whether a given set of motion rules maintains connectivity, and whether a goal configuration is reachable from a given initial configuration at specified locations.

Another study that explores this idea of metamorphic robot system is that of Walter et al. [78], which presented a distributed algorithm for reconfiguring a straight chain of hexagonal modules at one location to any intersecting straight chain configuration at some other location in the plane.

2.1.3 Other Approaches

The coordination between multiple robots has been also seen as an optimization problem, and addressed by metaphors such as a free market optimization [24]. Other studies addressed the problem of coordination in networked robotics (e.g., [52]), and the problem is sometimes expressed as a global control system that relies on tight real-time network guarantees. Also, in distributed systems by Yared et al. [81], the support for coordination

is provided by agreement and distributed locking protocols, but relies on the accuracy of a global positioning system.

Recently, the problem of coordination of multiple mobile robots was approached from an algorithmic or computational standpoint (e.g., [56], [69]), in which we find our interest (a detailed survey of the works in this area is provided in the next section). With this approach, the question can be as follows: given a desired *global* behavior, what *local* control behavior shall we give to robots so that they can coordinate their actions (a survey on the works in this direction is provided in the next section).

At the moment, mobile robotics is an empirical discipline. Control programs are designed through trial-and-error, and have to be refined through experimentation with the robots in the target environment.

Nehmzow [51] discusses some of the future challenges of mobile robotics research, including applying quantitative methods in mobile robotics, in order to change the discipline from an empirical one to a more precise science.

2.2 Computational Approach

With the computational approach [32, 33, 68, 69], the aim is to identify the algorithmic limitations of what distributed autonomous mobile robots can do, and to provide a formal representation of multi-robot systems and coordination problems, in which provably correct solutions can be developed and verified. In particular, two main computational models were proposed in the literature [56, 69], and some studies attempted to characterize the influence of the models on the ability of a group of robots to perform certain basic tasks under several assumptions [55, 57].

Prencipe [55, 58] has studied the minimal requirements and capabilities that the robots must have in order to achieve their goals in an asynchronous environment. In his study, the author compared his Corda model [56], with the SYm [69]. In the Corda model, the activities of the robots (wait-observe-compute-move) operate in a fully asynchronous way, however, the robots must share a common coordinate system, which is not the case in the SYm model. He also studied the relationship between the classes of problems solvable in the two models, and he has proved that the *instantaneous action* of the SYm model is at least as powerful as the *full asynchrony* of Corda.

Typical problems that have been studied in this perspective are the arbitrary pattern formation problem, where the robots are asked to form a pattern (e.g., a circle, a polygon) given in input, in finite time [32, 69]; the gathering problem, where the robots are asked to gather at some location, not determined in advance [1, 34, 59, 66]; and the flocking problem, where the robots are required to keep a formation while moving [35].

Very few studies have considered failure in multiple robot systems. An attempt by Yoshida et al. [83], considered initial crash faults of robots, and proposed an algorithm for

the *Active Robot Selection Problem* (ARSP). The ARSP creates a sub-group of non-faulty robots from a group that includes also initially crashed robots, and makes the robots in that subgroup recognize one another.

2.2.1 Gathering Problem

The gathering problem (also known as rendez-vous or point formation) is defined as follows. Given a group of robots, initially located at random locations, these robots are required to move in such a way that they eventually end up at the exact same location, not determined a priori. An algorithm that solves the gathering problem must ensure that the final configuration is obtained within a finite number of steps, from any initial situation and in every possible execution.

Despite its apparent simplicity, the problem of gathering robots at a single point is surprisingly difficult, and has been studied extensively in the literature, in different models and under various assumptions. In fact, several factors render this problem difficult to solve [16, 34, 59, 69]. In particular, in these studies, the problem has been solved only by making some additional assumptions regarding the capabilities of the robots. This problem has also interesting applications. For instance, the ability to gather at a single point means that the robots can reach an agreement on a common origin.

Earlier studies of the gathering problem include the work of Suzuki and Yamashita [69]. In their model, they proposed an algorithm to solve the gathering problem deterministically for three or more robots, in the case where robots have unlimited visibility and they are oblivious. Using the same model, Ando et al. [2] have proposed an algorithm to address the gathering problem in systems wherein robots have limited visibility. Their algorithm converges toward a solution to the problem, but it does not solve it within a finite time.

In the Corda model [56], Cieliebak et al. [17] have proposed a deterministic algorithm that gathers the robots at a point, in systems in which robots have unlimited visibility. Among other things, one feature the robots must have in order to solve this problem is the ability to detect multiple robots at a single point. Using the same model, Flocchini et al. [34] proposed a deterministic algorithm for the gathering problem, where robots have limited visibility, and are oblivious. However, their algorithm requires that robots share a common sense of direction as given by compasses.

Later on, Prencipe [59] proved that in general, it is impossible to solve the gathering problem deterministically in both SYm and Corda models when robots are oblivious, they do not share a compass and they are unable to detect multiplicity (i.e., whether there is more than one robot at one point).

Other studies of the gathering problem have been devoted to providing solutions for eventual convergence at a point [18].

The gathering problem has been also studied for faulty robots by Agmon and Peleg [1]

in synchronous and asynchronous settings. In particular, they proposed an algorithm that tolerates one crash-faulty robot in a system of three or more robots, and they show the impossibility of tolerating Byzantine[1] robots. Later on, Défago et al. [22] strengthen this impossibility by showing that it still holds in stronger models. They also show the existence of randomized solutions for systems with Byzantine-prone robots.

Recently, Cohen and Peleg [19] also addressed the issue of analyzing the effect of errors in solving gathering and convergence problems. In particular, they studied imperfections in robot measurements, calculations and movements. They showed that gathering cannot be guaranteed in environments with errors, and illustrated how certain existing geometric algorithms, including ones designed for fault-tolerance, fail to guarantee even convergence in the presence of small errors. One of their main positive results is an algorithm for convergence under bounded measurement, movement and calculation errors.

Among our recent work, we have studied the gathering of asynchronous mobile robots when they are equipped with *inaccurate* compasses [65]. In particular, we have provided an algorithm to gather, in a finite time, two oblivious robots equipped with compasses that can differ by as much as $\pi/4$. A similar result has been presented by Imasu et al. [38] at a domestic workshop in Japan.

Katayama et al. [41] has extended our work on gathering with inaccurate compasses [65] by defining two different models of compasses; fixed compass model and semi-dynamic compass model. In the fixed compass model, they proposed an algorithm that tolerates $\pi/3$ errors of compasses, and in the semi-dynamic compass model, they proposed an algorithm that solves the problem when robot's compasses can differ by as much as $\pi/4$, and the direction of the compass may vary between two different cycles, but it never changes within a cycle.

Finally, a recent study on the gathering of fat robots was done by Czyzowicz et al. [20], in which they represented robots by unit discs, and they proposed an algorithm to gather at most four robots in the plane under the Corda model.

2.2.2 Circle Formation Problem

The problem of forming a circle by a group of mobile robots has interesting applications. For instance, consider the context of space exploration and the initial preparation of a zone. A group of robots could be sent, and after landing at random locations, would self-organize to form the initial infrastructure for later expeditions. Also, pattern formation problems in general provide a first step toward flocking, i.e., allowing a group to move in formation [35]. From a conceptual standpoint, forming a circle provides a way for robots to agree on both a common origin and a common unit distance [69].

Debest [21] briefly discussed the formation of a circle by a group of mobile robots as

[1] A robot is said to be Byzantine if it executes arbitrary steps that are not in accordance with its local algorithm [46].

an illustration of self-stabilizing distributed algorithms. He discussed the problem, but did not provide an algorithm.

Sugihara and Suzuki [67] proposed several algorithms for the formation of various geometrical patterns. They proposed a simple heuristic algorithm for the formation of an approximation of a circle in the limited visibility setting. Their solution does not always reach a desirable configuration, and sometimes it brings the robots to form a Reuleaux triangle.[2]

Later on, Suzuki and Yamashita [69] proposed in their model a *non-oblivious* algorithm for the formation of a regular polygon. To achieve this, robots must be able to remember all past actions and observations. The existence of an *oblivious*, and thus self-stabilizing solution was, however, left as an open question.

In the same model, the circle formation problem was studied by Défago and Konagaya [23], who proposed an oblivious algorithm for the circle formation problem. With that algorithm, robots deterministically make a circle, albeit not uniformly, and converges asymptotically toward a solution whereby robots are uniformly distributed along the circle boundary. Unfortunately, that algorithm was unnecessarily complex, and relied on the computation of Voronoi diagrams.[3]

Recently, several studies address the same problem, relying on particular configurations of the robots [13, 26, 42]. Chatzigiannakis et al. [13] proposed a partial solution to the circle formation problem. Unfortunately, their solution relies on a simplifying assumption that completely removes the difficulty of the problem (in particular the robots must not be located on the same radius).

Later on, Katreniak [42] proposed, in the Corda model [56], an algorithm that solves a slightly simpler problem, called biangular circle.[4]

Another recent study on circle formation was by Dieudonné et al. [25], in which they proposed a deterministic solution to the problem, combined with the work of Katreniak [42]. In particular, their algorithm does not solve the problem in the case of $n = 4$, 6 and 8. Finally, Dieudonné and Petit [26] proposed an algorithm to solve the circle formation problem for a *prime* number of robots, which is based on Lyndon words.[5]

[2]A Reuleaux triangle is a curve of constant width constructed by drawing arcs from each polygon vertex of an equilateral triangle between the other two vertices [79].

[3]The *Voronoi diagram* Voronoi(P) of a set of points $P = \{p_1, p_2, \ldots, p_n\}$ is a subdivision of the plane into n cells, one for each point in P. The cells have the property that a point q belongs to the Voronoi cell of point p_i, denoted $\mathrm{Vcell}_{p_i}(P)$, if and only if, for any other point $p_j \in P$, $\mathrm{dist}(q, p_i) < \mathrm{dist}(q, p_j)$, where $\mathrm{dist}(p, q)$ is the Euclidean distance between p and q. In particular, the strict inequality means that points located on the boundary of the Voronoi diagram do not belong to any Voronoi cell. Significantly more details about Voronoi diagrams and their principal applications are surveyed by Aurenhammer [4].

[4]In a biangular circle, there is a center and two nonzero angles α and β, such that the center of the circle between each two adjacent points is either α or β, and these angles alternate.

[5]A Lyndon word is a certain type of string over an alphabet, and it has the property that, whenever it is split into two substrings, it is always lexicographically less than the right substring [77].

2.2.3 Pattern Formation Problem

Flocchini et al. [32] discussed the problem of arbitrary pattern formation, of which they give an informal definition. They show several important results about this problem, depending on what common knowledge the robots are assumed to share about the coordinate system. The authors give a more formal definition of the problem in their later work [33].

Ando et al. [3] discussed fundamental formation and location agreement problems for synchronous robots with limited visibility.

Suzuki and Yamashita [69] studied the formation of geometric patterns in the plane. They proposed a non-oblivious algorithm for the circle formation problem. They showed that asymmetric patterns cannot be formed in their model. Also, Sugihara and Suzuki [69] proposed algorithms for different geometric patterns. In particular, they proposed an algorithm for the formation of an approximation of a circle, a simple polygon and a line segment in the plane.

The flocking problem, where robots are required to move in formation like a flock of birds, has been expressed in the literature in a "leader-followers" model [35, 37, 56]. One robot (the leader) is chased by the other robots (the followers). The robot leader determines the path that the group must follow. In contrast, the followers must follow the leader in such a way that the relative positions of the robots always form a given shape. In his CORDA model, Prencipe [56] presented an oblivious algorithm that allows the robots to keep formations that are symmetric with respect to the direction of movement of the leader. Gervasi and Prencipe [36] simulate the algorithm and present interesting results.

The formation of geometrical patterns and flocking are both useful, for instance, for the self-positioning of mobile base stations in a mobile ad hoc network, e.g., as considered by Chatziagiannakis et al. [14], and for the self-deployment of sensors in a network ring [31]. In addition, solutions to the flocking problem are useful primitives for larger tasks, for instance, box pushing or cooperative manipulation, where robots can be asked to move heavy loads.

2.3 System Models

In this dissertation, we consider two system models that differ by the degree of synchrony between the robots. The first model, called SYm and proposed by Suzuki and Yamashita [69] is semi-synchronous. The second one, called CORDA, and proposed by Prencipe [56] is fully asynchronous.

2.3.1 The SYm Model [69]

The system consists of a set of autonomous mobile robots $\mathcal{R} = \{r_1, \cdots, r_n\}$ roaming on a two-dimensional plane. Each robot is modelled and viewed as a point in the plane, and is equipped with sensors to observe the positions of the other robots. In particular, each robot is able to sense its surroundings, perform computations on the sensed data, and move toward the computed destination. This behavior constitutes its cycle of *sensing*, *computing*, *moving* and being *inactive*. The sequence *Look-Compute-Move-Wait* is called the *cycle* of a robot. The SYm model assumes that activations (look, compute, move) occur instantaneously, resulting in a form of implicit synchronization. The model is called *semi-synchronous* model for this reason.

The robots are *anonymous*, in the sense that they can not be distinguished by their appearance, and they do not have any kind of identifiers that can be used during the computation. In addition, there is no direct means of communication among them. Hence, the only way for robots to acquire information is by observing each other's positions.

In the SYm model, time is represented as an infinite sequence of discrete time instants $t_0, t_1, t_2, \ldots$, during which each robot can be either *active* or *inactive*. When a robot becomes active, it observes the environment, computes a new location, and moves to it. In particular, the activities of observation, computation and movement are *instantaneous*, and thus, a robot observes other robots only when a cycle begins (i.e., when the robots are stationary).

The cycle of a robot is finite, and the activation of robots is determined by an activation schedule, which is unpredictable and unknown to the robots. At each time instant, a subset of the robots becomes active, with the guarantees that: (1) Every robot becomes active at infinitely many time instants, (2) At least one robot is active during each time instant,[6] and (3) The time between two consecutive activations is not infinite.

In every single activation, the distance that robot r can travel in one cycle is bounded by $\delta_{r_i} > 0$. Specifically, if the destination point computed at a given cycle by robot r is farther than δ_{r_i}, then the algorithm returns a point of at most δ_{r_i}. This distance may be different for different robots.

In the SYm model, each robot uses its own local x-y coordinate system, which includes an origin, a unit distance, and the directions of the two x and y axes, together with their orientations. However, the robots share neither knowledge of the coordinate systems of the other robots, nor of a global coordinate system.

2.3.2 The Corda Model [56]

The Corda model [56] is similar to the SYm model except in the way the time of the operations executed by a robot during its cycle is modelled. In particular, in the Corda

[6] As the duration of the interval between two time instants is by no means fixed, the second condition incurs no loss of generality. It is in fact only required for convenience.

model, the robots are asynchronous, in the sense that the amount of time spent in *observation*, *computation*, *movement* and *inaction* is finite, but otherwise unpredictable. That is, each robot executes its computation cycle as follows: a robot is initially in a *waiting* state (*Wait*), asynchronously and independently from the other robots, it *observes* the environment (*Look*) by taking a snapshot of the positions of the robots. Then, it *computes* a destination point based on the observed positions (*Compute*). Finally, the robot moves (*Move*) toward the computed destination, but the distance it moves is unmeasured; neither infinite, nor infinitesimally small (see Assumption 2.3.1). Hence, the robot can only go toward its goal, but the move can end anywhere before the destination.

In the CORDA model, the (global) time that passes between two successive states of the same robot is finite, but unpredictable. In addition, no time assumption within a state is made. This implies that the time that passes after the robot starts observing the positions of all others and before it starts moving is arbitrary, but finite. That is, the actual movement of a robot may be based on a situation that was observed arbitrarily far in the past, and therefore it may be totally different from the current situation.

The robots can be partitioned into sets depending on their state at some time t:

- $\mathbb{W}(t)$ is the set of all robots that are in *Wait* state at time t.
- $\mathbb{L}(t)$ is the set of all robots that are in *Look* state at time t.
- $\mathbb{C}(t)$ is the set of all robots in the state *Compute* at time t; the subset $\mathbb{C}_{\varnothing}(t)$ contains the robots whose computation results in executing a *null move*.
- $\mathbb{M}(t)$ is the set of all the robots that are executing a movement at time t; the subset $\mathbb{M}_{\varnothing}(t)$ contains the robots executing a *null move*.

In this model, the cycle of a robot is finite. In addition, there are the following two assumptions related to the distance travelled by a robot in one cycle, and the time required for the completion of a cycle.

Assumption 2.3.1 *It is assumed that the distance travelled by a robot r in a move is not infinite. Furthermore, it is not infinitesimally small: there exists a constant $\Delta_r > 0$, such that, if the target point is closer than Δ_r, r will reach it; otherwise, r will move toward it by at least Δ_r.*

Note that without this assumption, it would be impossible for any algorithm to terminate in a finite time.

Assumption 2.3.2 *The amount of time required by a robot r to complete a cycle (look-compute-move-wait) is not infinite. Furthermore, it is not infinitesimally small; there exists a constant $\tau_r > 0$, such that the cycle will require at least τ_r time.*

2.3.3 Further Assumptions

In this thesis, we further make the following assumptions, in addition to the assumptions made in the above described models:

1. We assume that the robots have *limited visibility*, in the sense that each robot can sense only up to a distance $VR > 0$ from it. In other words, each robot can see only the robots which are within its visibility radius VR.

2. We assume that the robots are *oblivious* or *memory-less*, which implies that they are unable to remember past actions and observations, and thus, their computations can not be based on previous observations. This last assumption is useful both for memory management and because an algorithm designed for such robots in inherently self-stabilizing.

3. We assume that the robots are unable to detect the presence of multiple robots at a single point (i.e., robots are unable to detect *multiplicity).*

2.3.4 Difficulty of Coordination

To illustrate the difficulty of gathering the models described above, consider simply two identical robots, say r_a and r_b that initially occupy distinct positions, consider the problem of having them eventually move to the same location (gathering). We can see the dilemma that r_a faces (and also r_b by symmetry) by considering the naive solution that follows.

Consider a stricter, more synchronized, model in which the robots can only be activated either simultaneously (i.e., robots observe same environment and move simultaneously) or sequentially (i.e., second observation made only after first move is completed).

Consider the viewpoint of one of the two robots, say r_a, when it is activated during some arbitrary step. Obviously, r_a can select one of two different points as its destination for the step, namely, the other robot's location (r_b), or the midpoint between r_a and r_b.

With the first choice, the robots gather at the end of the step if r_b is not activated. However, if r_b happens to be active simultaneously, it takes the same action as r_a, and hence moves to occupy the position that r_a has just left. Consequently, if the activation schedule is such that the two robots are always activated at the same time, the system remains caught in a cycle with the two robots endlessly swapping positions. As a result, the robots never gather in the worst case.

With the second choice, r_a selects the midpoint, and the two robots gather in one single step if they are activated simultaneously. Unfortunately, if the two robots end up being always activated sequentially, then they will *converge* towards each other, but it will still take an *infinite* number of steps to gather in the worst case.

2.3.5 Convergence versus Formation

Convergence to a point is the property of approaching or getting closer to a point. The point convergence problem is trivial to achieve by a set of robots modelled as points on the plane. A simple algorithm can make the robots move to the centroid forming the visible robots. However, the convergence algorithm takes an infinite number of steps to terminate. In other words, robots converge asymptotically toward the formation of a point.

The point formation (or gathering) problem requires that all robots move and end up at the exact same point in a finite number of steps. The point formation problem is difficult to achieve in the models described above, in particular, robots need to break the symmetry between each other in order to terminate in finite time. That is, robots need to agree which robot moves, and which robot remains stationary. In other words, formation can be also seen as a decision problem.

One of the main advantages of forming a point in finite time is the inherent ability to agree between robots on a common origin.

The convergence to a point can also provides a way to agree on a common origin, but with certain error. This depends in error measurements and considerations in practise.

Practically speaking, convergence and formation can be seen as the same problem. However, mathematically speaking they are different.

2.4 Notations and Geometric Properties

2.4.1 Notations

We denote by $\mathcal{U} = \{r_1, \cdots, r_n\}$ the set of all the robots in the system. Given some robot r, $r(t)$ is the position of r at time t. The circle $\mathcal{C}_r(t)$, centered at r, and with visibility radius VR, denotes the circle of visibility of r at time t. $\mathcal{R}_r(t)$ is the region enclosed by $\mathcal{C}_r(t)$.

Let A and B be two points, with $\overline{AB}$, we will indicate the segment starting at A and terminating at B, and $dist(A, B)$ is the length of such a segment. By $(\overline{AB})$, we denote the line passing through points A and B.

$\mathcal{C}(o, R)(t)$, denotes the circle centered at o, with radius R at time t. Let θ be a central angle with endpoints A and B located on the circumference of $\mathcal{C}$, then $\sphericalangle(AoB)$, denotes the circular sector at the central angle θ. Finally, we denote by $\mathcal{C}(\overline{AB})$, the circle with diameter $\overline{AB}$.

Given three distinct points A, B, and C, we denote by $\triangle(A, B, C)$, the triangle having them as corners, and by $\widehat{BAC}$, the angle formed by A, B and C, and centered at A.

Finally, given a region $\mathcal{X}(t)$ at time t, we denote by $|\mathcal{X}(t)|$, the number of robots in that region at time t. The parameter t is omitted whenever clear from the context. Let

S be a set of robots, then $|S|$ indicates the number of robots in the set S.

2.4.2 Definitions

Definition 2.4.1 (Local convex hull) *The local convex hull of robot r at time t, denoted by $LC(r,t)$ is the convex hull of robots' positions which are within visibility radius VR from robot r at time t. When all robots visible to r are collinear, the local convex hull of r represents a line.*

Definition 2.4.2 (Diameter of convex hull) *The diameter of a convex hull for $n \geq 3$ points (a convex polygon) is a local maximum for two of its vertices. That is the maximum distance between two of its vertices. When all points are collinear, we mean by the diameter of the convex hull, the distance between the two extreme points on the line.*

We introduce the following definitions due to Flocchini et al. [34].

Definition 2.4.3 (Distance graph) *Let $G(0) = (N, E(0))$ indicate the initial distance graph of the robots, whose node set N is the set of input robots, and $\forall r, s \in N$, $(r, s) \in E(0)$ if and only if r and s are at a distance no greater than the visibility radius VR.*

Definition 2.4.4 (Mutual visibility) *Two robots r and r' are* mutually visible *at time t, if both robots include each other in their computations. Formally, r and r' are mutually visible at time t, if and only if both conditions hold:*

1. $0 < dist(r, r') \leq VR$,
2. $r, r' \in \mathbb{L}(t) \cup \mathbb{C}_{\varnothing}(t) \cup \mathbb{M}_{\varnothing}(t) \cup \mathbb{W}(t)$.

Note that mutual visibility does not include robots with the same location.

We also introduce the following definition of gathering robots with volume (represented by unit discs in the plane) due to Czyzowicz et al. [20].

Definition 2.4.5 (Gathering Robots with Volume) *Gathering robots with volume means forming a configuration for which the union of all discs representing them is connected. Moreover, all robots must have full visibility to be aware that gathering is accomplished.*

2.4.3 Geometric Properties

We now introduce some geometric properties used later in this dissertation.

Lemma 2.4.6 *Every internal chord of a triangle has a length less than the longest side of the triangle.*

Lemma 2.4.7 *In an obtuse triangle, the side opposite the obtuse angle (angle greater than $\pi/2$ and less than π) is the longest side in the triangle.*

Lemma 2.4.8 *In a triangle, the side opposite to the largest angle in the triangle is the longest side in the triangle.*

Lemma 2.4.9 *Every internal chord of a circle has a length less than or equal to the diameter. That is, the distance between any two points that belong to a circle is less than or equal to the diameter.*

Chapter 3

Unreliable Compasses

Truth lies within a little and certain compass,
but error is immense.
William Blake

In this chapter, we give formal and informal definitions to different classes of unreliable compasses.

A compass is a navigational instrument for finding directions on the earth. More technically, a compass is a magnetic device using a needle to indicate the direction of the magnetic north of a planet's magnetosphere. The cardinal points are north, south, east and west [76].

However, these devices in general have limited accuracy. Besides, they are subject to instabilities because of magnetic interference, and they are error-prone devices. Therefore, it is very important to consider the unreliability of compasses when designing algorithms.

3.1 Definition of Compass

Before we proceed to the definition of a compass, we introduce the following definitions.

Definition 3.1.1 (Absolute north) *An absolute north $\overrightarrow{\mathcal{N}}$ is a vector that indicates a fixed north direction.*

Definition 3.1.2 (Relative north) *The relative north $\overrightarrow{N_A}(t)$ of some robot A is a function of time that returns a vector that indicates the north direction for robot A at some time t.*

Definition 3.1.3 (Compass) *A compass is a function of time and robots. The function outputs a relative north direction $\overrightarrow{N_r}(r,t)$ for some robot r at time t. By $\text{compass}_r(t)$, we denote the relative north of the compass of robot r at time t.*

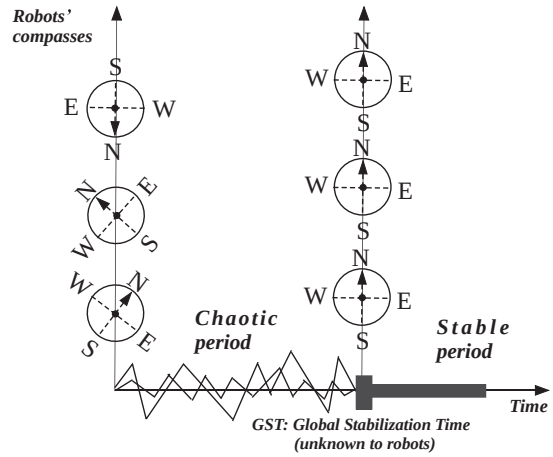


Figure 3.1: Eventually consistent compasses.

3.2 Perfect Compass

With a *perfect compass*, the robots always agree on the same north direction at any time t. In other words, the robots agree on the directions and orientations of both x and y axes at any time t. Formally, the robots in the system share a *perfect compass* if and only if the *agreement* and *invariance* properties are satisfied:

Definition 3.2.1 (Perfect compass) *A perfect compass is defined as follows:*

1. *Agreement:* $\forall r, r' \in \mathcal{U}, \forall t, compass_r(t) = compass_{r'}(t)$
2. *Invariance:* $\forall r \in \mathcal{U}, \forall t, t', compass_r(t) = compass_r(t')$

3.3 Unreliable Compasses

In this section, we define different classes of unreliable compasses for robots.

3.3.1 Eventually Consistent Compass (EVC)

With an *eventually consistent compass*, there exists a time after which all the robots agree on the same north direction. The agreement holds after some time GST (Global Stabilization Time) unknown to the robots. In other words, it is only guaranteed that the agreement on the north direction will hold, but the time for which the agreement holds is unknown to the robots. More precisely, an eventually consistent compass has the following properties: (1) The north direction of a robot's compass can change with time. (2) At a given time, the compasses of any two robots may disagree. (3) There exists some time GST after which, the compasses of all the robots agree for a sufficiently long period. Yet, the robots do not know when the time GST will occur. An eventually consistent compass is depicted in Figure 3.1.

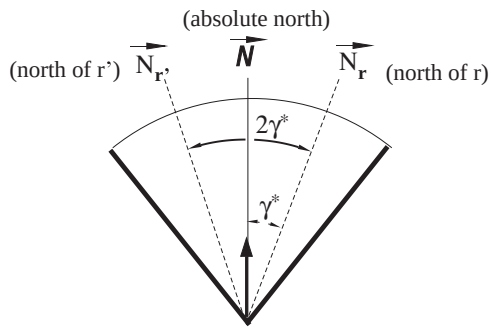


Figure 3.2: γ^*-Inaccurate compasses.

Formally, the robots in the system share an *eventually consistent compass* if and only if the *eventual agreement* and *eventual invariance* properties hold:

Definition 3.3.1 (Eventually consistent compass) *is a compass with the following properties:*

1. *Eventual agreement:* $\exists GST, \forall r, r' \in \mathcal{U}, \forall t \geq GST, compass_r(t) = compass_{r'}(t)$
2. *Eventual invariance:* $\forall r \in \mathcal{U}, \forall t, t' \geq GST, compass_r(t) = compass_r(t')$

We sometimes refer to eventually consistent compasses as unstable compasses.

3.3.2 Inaccurate Compass

Informally, we say that compasses are γ^*-*Inaccurate* if and only if, for every robot r, the absolute difference between the relative north indicated by the compass of robot r and the absolute north $\overrightarrow{\mathcal{N}}$ is at most γ^* at any time t (also referred to as *error* of the compasses). In addition, for every robot r, the error of its compass is consistent or invariant over time, that is, the relative north of each robot does not fluctuate over time.

In other words, a pair of γ^*-Inaccurate compasses can differ by as much as $2\gamma^*$ at any time t, and the difference is invariant. The special case when $\gamma^* = 0$ represents *perfect compasses.*

Definition 3.3.2 (γ^*-Inaccurate compasses) *Formally, compasses are* γ^*-Inaccurate *if, the following two properties are satisfied:*

1. γ^**-Inaccuracy:* $\forall r \in \mathcal{R}, \forall t, |\measuredangle \overrightarrow{\mathcal{N}} \overrightarrow{N}(r,t)| \leq \gamma^*$,
2. *Invariance:* $\forall r, \forall t, t', \overrightarrow{N}(r,t) = \overrightarrow{N}(r,t')$.

An inaccurate compass, also called error bounded compass, is depicted in Figure 3.2.

3.3.3 Eventually Bounded Error Compass (EBC)

The definition of eventually bounded error compass (EBC), derives from the definitions of an eventually consistent compass, and an inaccurate compass.

With an *eventually bounded error compass* (EBC), there exists a time after which the divergence error between the compasses of robots is bounded, and this happens after some time GST (Global Stabilization Time) unknown to the robots.

In other words, an eventually bounded error compass has the following two properties: (1) At a given time, the compasses of any two robots may disagree by any angle. (2) There exists some time GST, after which, the compasses of all the robots are inaccurate by a certain known value. However, the robots do not know when the time GST will occur.

Formally, the robots in the system share an *eventually consistent compass* if and only if the *Eventually γ^*-Inaccuracy* and *eventual invariance* properties hold:

Definition 3.3.3 (Eventually Bounded Error Compass (EBC)) *is a compass with the following properties:*

1. *Eventually γ^*-Inaccuracy:* $\exists GST, \forall r \in \mathcal{U}, \forall t \geq GST, |\measuredangle \overrightarrow{\mathcal{N}}\overrightarrow{N}(r,t)| \leq \gamma^*$
2. *Eventual invariance:* $\forall r \in \mathcal{U}, \forall t, t' \geq GST, \overrightarrow{N}(r,t) = \overrightarrow{N}(r,t')$.

3.3.4 Other Inaccurate Compasses

In this section, we introduce some definitions of inaccurate compasses that come from Katayama et al. [41]. The authors seem to work on these definitions in parallel to our work, and they have defined a richer model of inaccurate compasses in which they have introduced the notion of dynamism in compasses.

Definition 3.3.4 (Full-Dynamic Compass, FDC) *The full dynamic compass (FDC) is a compass whose indicated north may vary at any time during the execution.*

Definition 3.3.5 (Semi-Dynamic Compass, SDC) *The semi dynamic compass (SDC) is a compass whose indicated north may vary at the time between any two cycles, but it never changes during one cycle.*

Definition 3.3.6 (Fixed Compass, FXC) *The fixed compass (FXC) is a compass whose indicated north direction never varies during the execution of the algorithm.*

A fixed compass is equivalent to our definition of γ^*-Inaccurate compass.

Definition 3.3.7 (Eventually Fixed Compass, EFC) *The eventually fixed compass (EFC) is a compass whose indicated north direction is fixed after some point of time, but it may vary before that time.*

An eventually consistent compass (EVC) is weaker than a perfect compass. In the case of full visibility, an EVC compass is equivalent to a perfect compass.

In the inaccurate compass class, the full dynamic compass (FDC) is the weakest model, and the fixed compass class (FXC) or γ^*-Inaccurate compass is the strongest model.

In the case of full visibility, an eventually bounded error compass (EBC) is equivalent to γ^*-Inaccurate compass.

In this dissertation, we do not consider the models with dynamic compasses.

Chapter 4

Gathering with EVC Compasses in the SYm Model

Keep within compass and you may be sure,
that you will not suffer what others endure.
German proverb

Flocchini et al. [34] have proved that gathering is solvable in the asynchronous model CORDA with oblivious robots and limited visibility, as long as robots share compasses that provide perfect information on direction. However, these components are in general subject to instabilities and errors. Subsequently, the question that arises is, can we solve the same problem with weaker compasses?

In this chapter, we revise the practical significance of this assumption, and we study the solvability of the gathering problem in the face of instability of the compasses for some arbitrary periods, with the guarantee that compasses stabilize eventually. However, the time when the stabilization occurs is unknown to the robots.

Since the gathering problem is solvable with perfect compasses, one might argue that the problem would be easy, since eventually the compasses show the correct direction, and hence, the problem has almost the same complexity as in the case of a perfect compass. However, this is not true, as the robots do not know when the stabilization time will occur. Therefore, the algorithm designed for the case must guarantee that the robots do not lose sight of each other when their compasses are inconsistent (safety condition), and that when their compasses eventually become consistent, the algorithm must allow the robots to progress and gather at a single point in a finite number of steps (liveness condition). This is where the difficulty of the problem comes from, as one algorithm might be designed satisfying, for instance, the safety condition, but may not let the robots progress to gathering when their compasses eventually stabilize, and vice versa.

In this chapter, we study the solvability of the gathering problem, relying on eventually consistent compasses in the Suzuki and Yamashita model [69], referred to as a semi-synchronous model, by providing a deterministic solution to the problem. Our algorithm

Table 4.1: Solvability of the gathering problem deterministically with oblivious robots and limited visibility for $n \geq 2$ with no multiplicity detection.

	Compasses		
Model	**Perfect**	**Eventually consistent**	**None**
Asynchronous (CORDA)	Solvable[1]	Solvable for $n \leq 4$ (Chapter 5) *Impossible for some $n \geq 5$ (Chapter 6)*	Impossible[3]
Semi-synchronous (SYm)	Solvable[2]	Solvable *Chapter 4*	Impossible[4]

[1]: Proved in [34]
[2]: Deduction from [34]
[3,4]: Proved in [59] for $n \geq 2$
[4]: Proved in [69] for $n = 2$

is guaranteed to recover from any arbitrary configuration when the compasses of the robots eventually stabilize. We can argue that our algorithm is intrinsically self-stabilizing[1] [69] and offers protection against any number of transient failures in the compasses.

We also show that the gathering algorithm proposed by Flocchini et al. [34] cannot cope with instabilities in the compasses in both SYm and CORDA models.

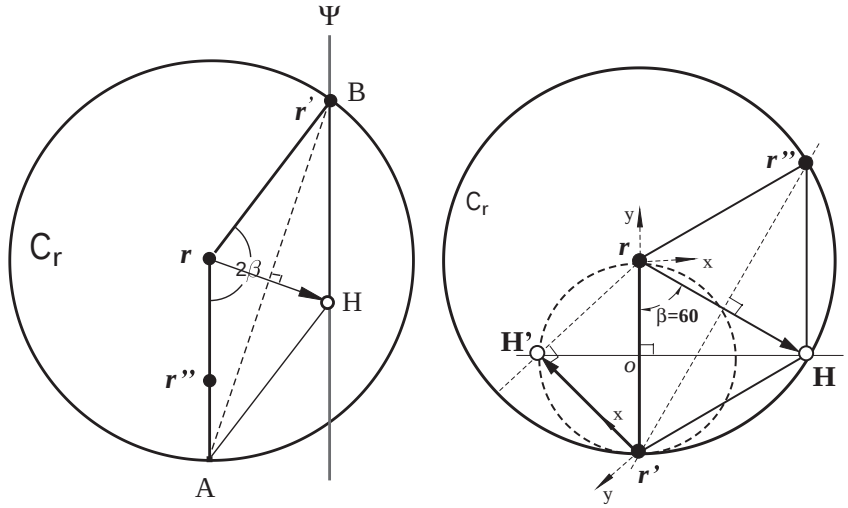
4.1 Solvability of Gathering

In this section, we discuss the solvability of the gathering problem in finite time in both SYm and CORDA models in the case of oblivious and limited visibility settings, where robots cannot detect multiplicity. Flocchini et al. [34] proved that the gathering problem is solvable deterministically when robots share perfect compasses by providing a solution to the problem. It is easy to see that the gathering problem is also solvable in the SYm model, when robots are equipped with perfect compasses, since all the possible executions in the SYm model are a subset of the possible executions in the CORDA model.

In the SYm model, Suzuki and Yamashita [69] showed that there is no oblivious algorithm for solving the gathering problem for the case of two robots. At a more general level, Prencipe [59] showed that without a compass and without multiplicity detection, there exists no deterministic oblivious algorithm that solves the gathering problem in finite time for $n \geq 2$ robots in both SYm and CORDA models. Table 4.1 summarizes these results.

In Section 4.3, we show that gathering is solvable deterministically in the SYm model relying on eventually consistent compasses, by providing a solution to the problem.

[1]Self-stabilization is the property of a system which, starting in an arbitrary state, always converges toward a desired behavior [27, 62].



(a) Diagonal move in Flocchini et al. [34] algorithm.

(b) The compasses of r and r' are inconsistent. Then, r executes a diagonal move to H, while r' executes a horizontal move to H'. As a result, $dist(H, H') > VR$: disconnection of the distance graph $G(0)$.

Figure 4.1: Flocchini et al. [34] algorithm in the face of eventually consistent compasses.

In Chapter 5, we show that gathering is also solvable deterministically in the CORDA model, when robots are equipped with eventually consistent compasses, for up to four robots.

Finally, in Chapter 6, we show that gathering has no solution in the CORDA model for a certain number of robots when compasses are unstable for some arbitrary periods.

4.2 Algorithm of Flocchini et al. [34] in the Face of EVC Compasses

In this section, we show that the algorithm proposed by Flocchini et al. [34] does not solve the gathering problem in either SYm or CORDA if we assume eventually consistent compasses. The proof is simply based on a counter example, combined with some activation schedule of the robots which leads to a configuration where the vision graph between robots will be partitioned, thus making the problem impossible, because robots are oblivious, and they are unable to remember their past observations and computations.

The Flocchini et al. [34] algorithm is described briefly as follows: depending on the positions of the robots in its circle of visibility, robot r computes its destination point as follows:

- *Null move*: if r sees robots to its left or above on its vertical axis, it does not move.
- *Vertical move*: if r sees robots only below it on its vertical axis, it moves down

toward the nearest robot.

- *Horizontal move*: if r sees robots only to its right, it moves horizontally toward the vertical axis of the nearest robot.

- *Diagonal move*: if r sees robots both below on its vertical axis and on its right, then it considers the nearest one below it, and the nearest one to its right, and performs a diagonal move as illustrated in Figure 4.1(a). In this move, the destination point of r is always computed in such a way so as to remain inside its circle of visibility. That is, r forms an angle 2β with its two mutually visible robots. Then, this move ensures that the destination point is always computed based on an angle $60 \leq \beta \leq 90$, so that robot r remains visible to the robots in its visibility range.

Recall that the initial distance graph $G(0)$ is connected. Assume now that the robots will execute the algorithm of Flocchini et al. [34], and that they are equipped with eventually consistent compasses. As a result, we will show that the *distance graph* will be disconnected. The proof is by a counter example. We will assume the SYm model. Then, the same proof is valid in the CORDA model.

Lemma 4.2.1 *In a system in which robots have eventually consistent compasses, the algorithm of Flocchini et al. [34] results in a disconnection of the initial distance graph $G(0)$ in the* SY*m model.*

PROOF. Let us consider the example illustrated in Figure 4.1(b). Assume that robot r performs a look operation at time t. Then, according to its compass, r sees robot r' below it on its vertical axis and r'' to its right[2] . By the algorithm, r will perform a *diagonal move*. Let H be its destination, such that $H \in \mathcal{C}_r(t)$, i.e., $\beta = 60°$. Assume also that r' performs a look operation at the same time t. According to its compass, robot r' sees robot r to its right. Consequently, r' will perform a *horizontal move*. Let H' be the destination of r' such that, $dist(r, H') = dist(r', H')$, as illustrated in Figure 4.1(b). By considering this simple example, we will show that the movement of r to H, and the movement of r' to H', will lead to a situation wherein r and r' are at a distance greater than the visibility radius VR; i.e, $dist(H, H') > VR$.

Let us denote by $\mathcal{C}(\overline{rr'})$, the circle with diameter $\overline{rr'}$, and by o its center. $H' \in \mathcal{C}(\overline{rr'})$ because $\overline{rH'} \perp \overline{r'H'}$ by the algorithm. We also have $H \in \mathcal{C}_r(t)$ because $\beta = 60°$ by construction. Hence, the triangle $\triangle(r, r', H)$ is equilateral. As a result, the perpendicular on $\overline{rr'}$ passing through H passes by o. In addition, the perpendicular to $\overline{rr'}$ passing through H', passes by o because the triangle $\triangle(r, r', H')$ is isosceles by construction ($dist(r, H') =$

[2]The right direction for a robot is the same as the East direction indicated by its compass, which is represented by the x-axis.

$dist(r', H'))$. We thus, compute the distance $dist(H, H')$ as follows:

$$\begin{aligned} dist(H, H') &= dist(o, H) + dist(o, H') \\ &= \sqrt{dist(r, H)^2 - dist(o, r)^2} + dist(o, H') \\ &= \sqrt{VR^2 - (VR/2)^2} + VR/2 \\ &= \frac{\sqrt{3}}{2} VR + VR/2 > VR \end{aligned}$$

We conclude that $dist(H, H') > VR$. Consequently, r and r' are unable to see each other any more. This results in a disconnection of the distance graph. $\square_{\text{Lemma 4.2.1}}$

Theorem 4.2.2 *In a system in which robots have eventually consistent compasses, the algorithm of Flocchini et al. [34] does not solve the robots's gathering problem in any of the models;* SY*m or* CORDA.

PROOF. From Lemma 4.3.1, we have gathering is unsolvable if the distance graph is disconnected. Also, from Lemma 4.2.1, we conclude that the algorithm of Flocchini et al. [34] leads to a disconnection of the distance graph under eventually consistent compasses. Thus, the theorem holds. $\square_{\text{Theorem 4.2.2}}$

4.3 Gathering with EVC in the SYm Model

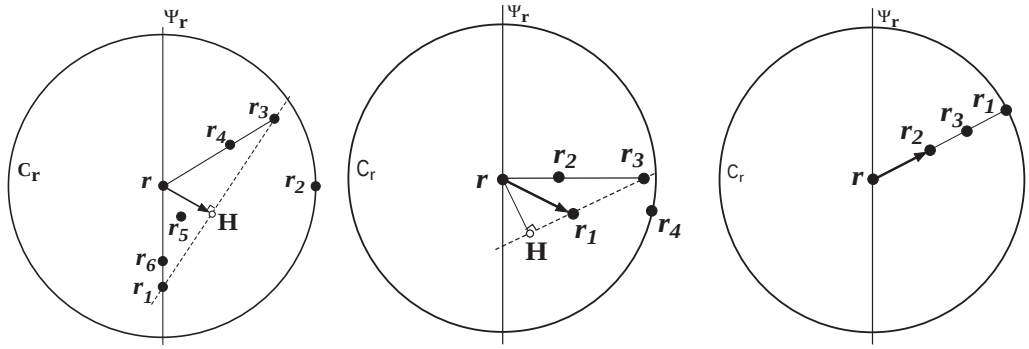
In this section, we present a deterministic algorithm for solving the gathering problem in the SYm model, when robots are oblivious, have limited visibility, and are equipped with eventually consistent compasses. Before we proceed to the description of the algorithm, we recall an important lemma due to Flocchini et al. [34].

Lemma 4.3.1 *If the initial distance graph $G(0)$ is disconnected, the gathering problem is unsolvable.*

The idea of the algorithm is to solve the problem by achieving the following two subgoals at every time instant t:

1. Robots that are visible at time t must remain visible at time $t + 1$, in spite of the inconsistencies in their compasses;

2. Robots located on the leftmost side at time t move toward the visible robots on their right side at time $t + 1$, and eventually gather at the rightmost and bottommost robot in the system after GST.



(a) r determines the positions of the two outermost robots r_1 and r_3. H is inside $\triangle(r, r_1, r_3)$, then r moves to H.

(b) r determines the positions of the two outermost robots r_1 and r_3. Since H is outside $\triangle(r, r_1, r_3)$, then r moves to r_1 instead of H.

(c) r is collinear with all robots in $\mathcal{R}_r$ and is the leftmost robot, then r moves to r_2.

Figure 4.2: Principle of the algorithm.

The gathering algorithm in the SYm model is depicted in Algorithm 1, where the functions Activation_Step($\mathcal{R}_r$, $compass_r$), Do_nothing(), and Move_to($Goal$) are as follows: the function Activation_Step($\mathcal{R}_r$, $compass_r$) is executed by robot r when it becomes active, and it takes as input the parameters of the visibility region $\mathcal{R}_r$ and $compass_r$ of robot r. The function Do_nothing() is executed by r when it stays still. Finally, the function Move_to($Goal$) terminates the computation of robot r and moves it toward $Goal$.

Before we proceed to the description of the algorithm in more detail, we further introduce the following notations. Let Ψ_r be the vertical axis passing through robot r according to its compass at time t. Ψ_r is collocated with the north direction indicated by the compass of r at time t. We denote by $Left_r(t)$ and $Right_r(t)$, the regions, respectively, to the left and to the right of Ψ_r, excluding Ψ_r. Let also Φ_r, be the perpendicular axis to Ψ_r passing by r. Then, we denote $Top_r(t)$ and $Bottom_r(t)$ as the regions respectively, above and below Φ_r excluding Φ_r. When no ambiguity arises, we shall omit the temporal indication. Finally, Ψ_r^+ and Ψ_r^- denote the intersections of Top_r and Ψ_r, and of $Bottom_r$ and Ψ_r, respectively.

Algorithm 1 is described informally as follows. First, at every time instant t where some robot r becomes active, r queries its compass, considers all the robots in its visibility region $\mathcal{R}_r(t)$, and then decides its movement as follows:

- If r sees robots on its left side $Left_r$, or on Ψ_r^+ (above it on Ψ_r), then, r does not move (line 13).

- If r is collinear with all robots in $\mathcal{R}_r(t)$, then r moves linearly to the nearest robot to it (robot r_2 in Figure 4.2(c)). In this case, r must be the topmost or leftmost robot in the line (line 16).

- If r sees some robots on $Right_r$ or on $Right_r$ and some robots on Ψ_r^-, then r computes

Algorithm 1 Gathering with EVC Compasses in the SYm Model

```
Activation_Step(R_r, compass_r)
if (|R_r| = 1) then {Gathering terminated; r sees only itself.}
   Do_nothing();
else
   Ψ_r := vertical axis passing through robot r according to compass_r;
   Φ_r := perpendicular to Ψ_r passing by r;
   Left_r := any robot in R_r to the left of Ψ_r, but not on Ψ_r;
   Top_r := any robot in R_r above Φ_r;
   Bottom_r := any robot in R_r below Φ_r;
   Ψ_r^+ := Top_r ∩ Ψ_r;
   Ψ_r^- := Bottom_r ∩ Ψ_r;
   if (|Left_r| > 0 ∨ |Ψ_r^+| > 0) then {r sees robots on its left side or on Ψ_r^+.}
      Do_nothing();
   else
      if (r is collinear with all robots in R_r) then
         Goal := nearest robot to r;
      else {r computes two outermost robots s_1 and s_2.}
         ÂrB: biggest central angle of C_r with endpoints A and B;
         s_1 := farthest robot from r on the segment rA;
         s_2 := farthest robot from r on the segment rB;
         H := foot of the perpendicular dropped from r to the side s_1s_2 in △(r, s_1, s_2);
      end if
      if (H ∈ △(r, s_1, s_2) ) then
         Goal := H;
      else {H is outside the triangle △(r, s_1, s_2).}
         s := nearest robot to r among s_1 and s_2;
         Goal := s;
      end if
      Move_to(Goal);
   end if
end if
```

the positions of the *two outermost robots*. The two outermost robots are computed as follows: First, r selects each pair of robots that form the biggest central angle with r in its circle of visibility $\mathcal{C}_r$, and that contains all robots visible to r. Second, if there are more than one pair of such robots, the robots at the maximum distance from r are selected (e.g., r_1 and r_3 in Figure 4.2(a)).

In other words, the two outermost robots are first, the two robots that form the biggest central angle with r and second, they are at the greatest distance from r.

Afterwards, r computes the height of the triangle that it forms with the two outermost robots, say s_1 and s_2, and having a base segment $\overline{s_1 s_2}$. Let H be the foot of a perpendicular starting at r. Then, r moves to H if H is inside the triangle $\triangle(r, s_1, s_2)$ (see Figure 4.2(a)). Otherwise, if H is outside $\triangle(r, s_1, s_2)$, then r moves to the closest robot to it among s_1 and s_2 (see Figure 4.2(b)).

4.4 Correctness of the Algorithm

In this section, we prove the correctness of our algorithm in two steps. In the first step, we show that the connectivity of the distance graph is preserved before and after GST. That is, the robots that are initially visible remain always visible during the entire execution of the algorithm. In a second step, we show that all the robots will gather at one point in a finite number of steps. Before proceeding, let us recall an important lemma proved by Flocchini et al. [34]; if the initial *distance graph* $G(0)$ is disconnected, the gathering problem is unsolvable. Hence, throughout we will always assume that the initial distance graph is connected.

4.4.1 Safety: Preserved Connectivity

We now prove that the connectivity of the distance graph is preserved during the entire execution of the algorithm. Recall that the compasses of the robots may be inconsistent, including those of the robots that are located at the same location. From the algorithm, trivially, we derive the following lemma:

Lemma 4.4.1 *Let r_1 and r_2 be the two outermost robots for some robot r, and $\widehat{ArB}$, the central angle whose sides pass by r_1 and r_2, and with endpoints A and B located on the circumference of $\mathcal{C}_r$. Let G be the destination of r. Then, $G \in \sphericalangle(ArB)$.*

Lemma 4.4.2 *Let robot r be active at time t, and $Left_r = \emptyset$ and $\Psi_r^+ = \emptyset$. Let r_1 and r_2 be its two outermost robots, and $\widehat{ArB}$, the central angle whose sides pass by r_1 and r_2, and with endpoints A and B located on the circumference of $\mathcal{C}_r$. Let also G be the destination of r. Then, for any point p in $\sphericalangle(ArB)$, we have $dist(p, G) < VR$.*

PROOF. Let H be the foot of the height of the triangle $\triangle(r, r_1, r_2)$ starting at r, and G be the destination of r.

By the algorithm, $G = H$ if $H \in \triangle(r, r_1, r_2)$; otherwise $G = r_1$ or $G = r_2$. Then, two cases follow, depending on whether H is inside or outside the triangle $\triangle(r, r_1, r_2)$:

1. *H belongs to $\triangle(r, r_1, r_2)$* (see Figure 4.3(a)).
   Consider the triangle $\triangle(r, r_1, H)$, by Lemma 2.4.6, for every point p in triangle $\triangle(r, r_1, H)$, we have $dist(p, H) < dist(r, r_1) \leq VR$. Similarly, $\forall p \in \triangle(r, r_2, H)$, $dist(p, H) < dist(r, r_2) \leq VR$. Consider now the subregion $\mathbf{P}(r_1, r_2, A, B)$ of the circular sector $\sphericalangle(ArB)$ delimited by r_1, r_2, A, and B, then $\forall p \in \mathbf{P}(r_1, r_2, A, B)$, the triangle that p forms with r and H is an obtuse triangle at H since $\widehat{rHr_1} = 90°$. Consequently, by Lemma 2.4.7, $\forall p \in \sphericalangle(ArB)$, $dist(p, H) < dist(p, r) \leq VR$. Hence, $\forall p \in \sphericalangle(ArB)$, $dist(p, G) < VR$.

2. *H does not belong to $\triangle(r, r_1, r_2)$* (see Figure 4.3(b)).
   In this case, $G = r_1$ (nearest to r among r_1 and r_2). Since H is outside the triangle $\triangle(r, r_1, r_2)$, then $\triangle(r, r_1, r_2)$ is an obtuse triangle at r_1. By Lemma 2.4.7, the segment $\overline{rr_2}$ is the longest side of the triangle. Thus, $\forall p \in \triangle(r, r_1, r_2)$, $dist(p, r_1) < dist(r, r_2) \leq VR$.

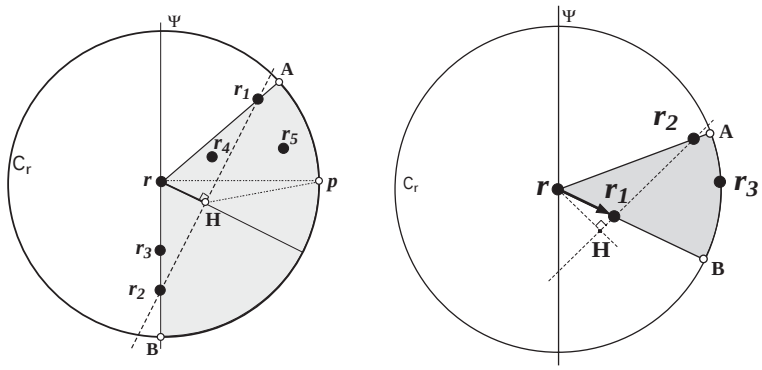
   Consider now the subregion $\mathbf{P}(r_1, r_2, A, B)$ of the circular sector $\sphericalangle(ArB)$ excluding the point B, then $\forall p \in \mathbf{P}(r_1, r_2, A, B)$, the triangle that forms p with r and r_1 is an obtuse triangle at r_1, since $\widehat{rr_1r_2} > 90°$. Consequently, by Lemma 2.4.7, $\forall p \in \mathbf{P}(r_1, r_2, A, B)$, $dist(r_1, p) < dist(r, p) \leq VR$. Let us now consider the point B, by hypothesis, $dist(r, B) = VR$ and $r_1 \in \overline{rB}$, then $dist(r_1, B) < VR$. Consequently, $\forall p \in \sphericalangle(ArB)$, $dist(r_1, p) < VR$. Hence, for all points p in $\sphericalangle(ArB)$, we have $dist(p, G) < VR$.

In both cases, $\forall p \in \sphericalangle(ArB)$, $dist(p, G) < VR$. This completes the proof. $\square_{\text{Lemma 4.4.2}}$

Lemma 4.4.3 *Let S be the set of robots visible to r at some time t. Then, at any time $t' > t$, r is at a distance of at most VR from all robots in S.*

PROOF. The proof consists of showing that the distance between the destination of r, and the destination of any robot in S at time $t + 1$ is less than or equal to VR. Trivially, the case of two robots holds, since one robot must move toward the other one. Thus, in the following, we assume that the number of robots in S is greater than one. Let robot r be active at time t. We distinguish the following cases, depending on the movement of r and whether the robots in S are active or not:

1. *Robot r is active at time t; all robots in S are inactive at time t.*
   We distinguish the following cases depending on the movement of robot r.



(a) Case where H is inside the triangle $\triangle(r, r_1, r_2)$ that r forms with the two outermost robots r_1 and r_2.

(b) Case where H is outside the triangle $\triangle(r, r_1, r_2)$ that r forms with the two outermost robots r_1 and r_2.

Figure 4.3: The destination of r is within distance VR from all robots in $\sphericalangle(ArB)$.

(a) *Robot r executes a null move.*
If r has robots on $Left_r$ or Ψ_r^+ then it does not move. In addition, by hypothesis, $\forall s \in S$, s is inactive at time t. This means that at time $t+1$, r remains at the original distance from all robots in S, and such distance is less than or equal to VR by hypothesis.

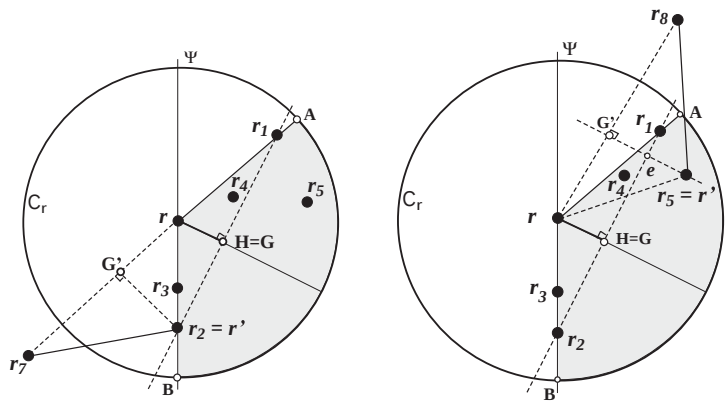
(b) *Robot r is collinear with all robots in S.*
Two possibilities follow: (1) Robot r can be in case (a) above (i.e., it executes a null move), so the lemma holds for case (1). (2) robot r can be the leftmost or topmost robot, then it performs a real move. Let r_1 be the robot farthest away from r on the line. By hypothesis, $dist(r, r_1) \leq VR$ and by the algorithm, r moves on the segment $\overline{rr_1}$. Thus, at time $t+1$, r gets closer in distance to all robots in S.

(c) *Robot r computes the positions of the two outermost robots.*
Let r_1 and r_2 denote the two outermost robots of r at time t, and let $\sphericalangle(ArB)$ be the circular sector enclosing all the robots in S such that $dist(r, A) = dist(r, B) = VR$, and $r_1 \in \overline{rA}$ and $r_2 \in \overline{rB}$ (see Figure 4.3(a)).

We denote by G the destination of r. Then, by Lemma 4.4.2, for every point p in $\sphericalangle(ArB)$, we have $dist(p, G) < VR$. Thus, $\forall r_i \in S$, $dist(r_i, G) < VR$.

2. *Robot r is active at time t; some or all robots in S are also* active *at time t.*
We consider $r' \in S$ to be active at time t. Let G' be its destination, and S' be the set of robots visible to r' at time t. Let also G be the destination of r. In the following, we will show that at time $t+1$, $dist(G, G') \leq VR$. We only prove the case for r', and the same argument can be applied to the other robots in S.

(a) *Robot r executes a* null move*; Robot r' executes a* null move.



(a) r' has the robots r and r_7 as outermost robots, and computes G' as destination. Then, $dist(G, G') < dist(r, r') \leq VR$.

(b) r' has the robots r and r_8 as outermost robots, and computes G' as destination. Then, $dist(G, G') \leq dist(r, e) \leq VR$.

Figure 4.4: The destination of r and r' are within distance VR from each other.

By hypothesis, $dist(r, r') \leq VR$, thus the case holds trivially.

(b) *Robot r executes a null move; Robot r' is collinear with all robots in S'.*
This case is similar to case 1(b). above, since r stays still at time t.

(c) *Robot r executes a null move; Robot r' computes the positions of its two outermost robots.*
This case holds by case 1(c). above, since r stays still at time t.

(d) *Robot r is collinear with all robots in S; Robot r' computes the positions of its two outermost robots.*
Let $\sphericalangle(A'r'B')$ be the circular sector of robot r' at time t. By Lemma 4.4.1, $G' \in \sphericalangle(A'r'B')$. In addition, $r \in \sphericalangle(A'r'B')$, and by the algorithm, $G \in \overline{rr'}$ (r moves on $\overline{rr'}$). Consequently, $G \in \sphericalangle(A'r'B')$, and by Lemma 4.4.2, the proof holds for this case.

(e) *Robot r computes the positions of its two outermost robots; Robot r' computes the positions of its two outermost robots.*
Depending on where robot r' is located in the circular sector $\sphericalangle(ArB)$, and where its visible robots (other than robot r) are located, its destination G' can either be within $\sphericalangle(ArB)$ or outside it. We thus, distinguish the following cases:

- The destination G' of r' belongs to $\sphericalangle(ArB)$ (r can be an outermost robot of r' or not): In all cases where $G' \in \sphericalangle(ArB)$, by Lemma 4.4.2, $\forall p \in \sphericalangle(ArB)$, $dist(p, G) < VR$. Therefore, $dist(G', G) < VR$. This completes the proof for this case.
- The destination G' of r' does not belong to $\sphericalangle(ArB)$ (r can be an outermost robot of r' or not):

Let r_1 and r_2 denote the two outermost robots of r at time t. In this case, we assume that the destination G of r is the foot of the perpendicular to the segment $\overline{r_1 r_2}$ dropped from r. The case when the destination of r is the location of one of its two outermost robots can be adapted easily. We assume the same for the destination G' of r' (i.e., G' is the foot of the perpendicular to the segment defined by its two outermost robots).

(a) $\exists s \in S'$ such that s does not belong to $\sphericalangle(ArB)$; and r' is one of the outermost robots of r (see Figure 4.4(a)).
In this case, assume that r' is the robot r_2, and its two outermost robots are r and r_7. We will show that $dist(G, G') \leq VR$. Consider the circle $\mathcal{C}(\overline{rr'})$ with diameter $\overline{rr'}$, we have $G \in \mathcal{C}(\overline{rr'})$ because $\widehat{rGr'}$ is a right angle by construction (see Algorithm 1). Similarly, $G' \in \mathcal{C}(\overline{rr'})$. Consequently, by Lemma 2.4.9, $dist(G, G') \leq dist(r, r') \leq VR$.

(b) $\exists s \in S'$ such that s does not belong to $\sphericalangle(ArB)$; and r' is not an outermost robot for r (see Figure 4.4(b)).
In this case, assume that r' is the robot r_5, and its two outermost robots are r and r_8. We will show that $dist(G, G') \leq VR$.
Let $e = (\overline{r'G'}) \cap \overline{r_1 r_2}$. Consider the circle $\mathcal{C}(\overline{re})$ with diameter $\overline{re}$, we have $G \in \mathcal{C}(\overline{re})$ because $\widehat{rGr_1}$ is a right angle by construction. Similarly, $G' \in \mathcal{C}(\overline{re})$. Consequently, by Lemma 2.4.9, $dist(G, G') \leq dist(r, e) \leq VR$. This completes the proof.

In all cases, r remains within distance VR from all robots in S at time $t+1$, and the rest follows by induction. $\square_{\text{Lemma 4.4.3}}$

From Lemma 4.4.3, we conclude that:

Theorem 4.4.4 *Algorithm 1 preserves the connectivity of the distance graph.*

4.4.2 Liveness: Termination of the Algorithm

In this section, we show that Algorithm 1 solves the gathering problem deterministically. Thus, in the following, we consider the system after time GST has been reached. Thus, all robots agree on the direction of their compasses.

Lemma 4.4.5 *In any collinear configuration of robots, all robots will gather in a finite time at the rightmost or bottommost robot.*

PROOF. In a configuration where robots are collinear, there exist two cases; either all the robots are located on the same vertical axis Ψ, or they are collinear, but not on the same Ψ. Consider the first case, where all robots are located on the same Ψ. By assumption,

the activation schedule is fair. Then, whenever the topmost robot becomes active, it will move to the nearest one below it. Since, the cycle of a robot is finite, and the number of robots is finite, then recursively, all robots in Ψ will gather at the bottommost robot in finite time. Similarly, in the second case, the leftmost robot will reach the nearest one to its right in a finite time. Thus, by using the same arguments, all robots will gather at the rightmost robot in a finite time, and the lemma holds. $\square_{\text{Lemma 4.4.5}}$

Lemma 4.4.6 *In any configuration with three or more robots, all robots will gather in a finite time at the rightmost and bottommost robot.*

PROOF. We recall that the robots reach the time GST. The proof is a simple adaptation of the proof of the Flocchini et al [34] algorithm.

Let Ψ_{left} be the leftmost vertical axis that passes by the leftmost robot (one or many robots) at time t. Let also Ψ_{right} be the rightmost vertical axis that passes by the rightmost robot at time t. Let D be the horizontal distance between Ψ_{left} and Ψ_{right}. If $D = 0$, this means that all the robots in the system are located on the same vertical axis. Then, by Lemma 4.4.5, they will gather at the bottommost robot in a finite time.

We now consider the case when $D \neq 0$. Assume by contradiction that some robots never reach Ψ_{right}. This means that there are some axes that will not be passed by all the robots that were to the left of the robots at the beginning of the algorithm: we call them *limit axes*. Let Ψ be the leftmost such axis. Let $\mathbb{A}$ be the sets of robots, initially to the left of Ψ, that will become arbitrarily close to Ψ but never reach it. Let $\mathbb{B}$ be the sets of robots, initially to the left of Ψ, that will pass Ψ within finite time. Finally, let $\mathbb{C}$ be the sets of robots, initially to the left of Ψ, that will reach Ψ without ever moving to the right of Ψ.

First observe that since the robots reach the time GST, then they will move toward the right. Second, if some robot r leaves its vertical axis Ψ_r, then by Assumption 2.3.1, it will progress toward Ψ_{right} by some distance $d > 0$, with $d = \delta_r \sin \beta_r$, where $\delta_r \neq 0$ is the distance between r and its target on the right, and $0 < \beta_r \leq 90°$ is a non-null angle that r forms with Ψ_r and its destination. Let $\beta > 0$ be the minimal angle that some robot can form with its vertical axis and its destination to the right, and δ be the minimal distance travelled by any robot toward its target.

Let t' be a time when all robots in $\mathbb{B}$ have passed Ψ, and those in $\mathbb{C}$ have reached Ψ. That is at time t', the only robots to the left of Ψ are those in $\mathbb{A}$.

Consider first the case when $\mathbb{A} = \emptyset$. In this case, by Lemma 4.4.5, after a finite number of moves, one of the robots in $\mathbb{C}$ will leave Ψ. A contradiction.

Now we assume that $\mathbb{A} \neq \emptyset$. Consider a vertical axis Ψ' to the left of Ψ, at distance $d' < \delta \sin \beta$ from Ψ. Since Ψ is the leftmost limit axis, each $r \in \mathbb{A}$ arrives at the right of Ψ' within finite time. Observe that, once to the right of Ψ', r must stop at least once,

since by definition, it does not reach Ψ. Let $t'' > t'$ be a time when all robots in $\mathbb{A}$ have stopped at least once to the right of Ψ'. Let also Ψ'' be an axis between Ψ' and Ψ, such that at time t'' no robot in $\mathbb{A}$ is to its right. Since Ψ'' is not a limit axis, the robots of $\mathbb{A}$ will pass Ψ'' within finite time. Since at time t'' there are no robots between Ψ'' and Ψ, the first robot $r \in \mathbb{A}$ that passes Ψ'' must have as destination a point to the right of Ψ or on Ψ. According to the Algorithm, r will move on a straight line at an angle β', with $\beta \leq \beta' \leq 90°$; such a line intersects Ψ at a point H. Since this move by r is started from a point S to the right of Ψ', then $dist(S, H) < \frac{d'}{\sin \beta'} < \frac{\sin \beta}{\sin \beta'}.\delta \leq \delta$. Thus, in this move r will reach Ψ. A contradiction. Consequently, no limit axis Ψ exists, and all robots reach the rightmost axis Ψ_{right} in finite time. $\square_{\text{Lemma 4.4.6}}$

Lemma 4.4.7 *Under Algorithm 1, all the configurations in which all the robots gather at one point are stable.*

PROOF. Assume that at some time t, all the robots gather at one point. In such a configuration, none of the robots see other robots in their visibility regions. Thus, by the algorithm, none of the robots will ever move. Consequently, such a configuration is stable by the algorithm. $\square_{\text{Lemma 4.4.7}}$

Theorem 4.4.8 *Under Algorithm 1, all robots gather at one point in finite time.*

PROOF. By Lemma 4.4.5 and Lemma 4.4.6, any configuration of robots is transformed to gathering in a finite time. Moreover, by Lemma 4.4.7, the gathering configuration is stable. This completes the proof. $\square_{\text{Theorem 4.4.8}}$

From Theorem 4.4.4 and Theorem 4.4.8, it follows that:

Theorem 4.4.9 *In a system, with n anonymous, oblivious mobile robots, with limited visibility, and eventually consistent compasses, the gathering problem is solvable deterministically in the* SYm *model.*

4.5 Complexity Analysis

In this section, we give an analytic analysis of the complexity of the algorithm, and the number of steps of its termination.

Complexity of the algorithm.

- The verification of robots that are on $Left_r$ or on Ψ_r^+ (line 12 in Algorithm 1) takes $O(n)$.

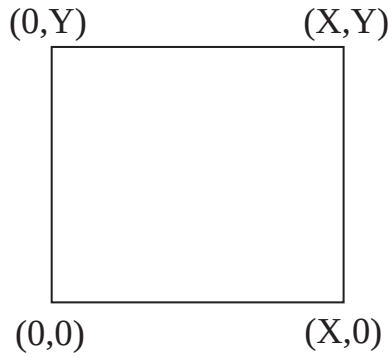


Figure 4.5: The coordinates of the portion of the plane occupied by the robots.

- The verification of robots that are collinear (line 15 in Algorithm 1) takes $O(n)$.
- For the computation of the two outermost robots, there are three steps:
  1. The algorithm needs to compute the different angles that each pair of robots makes with robot r. In the worst case, robot r is visible to all the robots in the system. Suppose that the coordinates of these robots that are visible to robot r are stored in an array. Then, the algorithm takes $O(n^2)$ to browse the array, and compute all the different angles that robot r forms with each different pair of robots.
  2. The algorithm needs to sort these angles in order to find the biggest angle. This will take $O(n \log n)$ using the QuickSort algorithm.
  3. Finally, in the case where there are several pairs of robots with the biggest central angle with robot r, the algorithm takes $O(n)$ in order to find the pair of distant robots among them.
- In conclusion, the complexity of the algorithm in one step is: $n^2 + n \log n + 3n$, which is $O(n^2)$.

Number of steps of termination of the algorithm. First, assume that robots occupy the portion of the plane as depicted in Figure 4.5. The maximum x coordinates of is X, and the maximum y coordinates is Y.

By assumption in the SYm model, the distance that a robot r can travel in one cycle is bounded by $\delta_{r_i} > 0$, and this distance is different between robots. We assume that the average distance travelled by a robot in one cycle is bounded by δ_{avg}.

To make the analysis of the algorithm, we assume that the compasses of robots are in the stabilization period. That is, robots has reached the time GST.

Note that, the extreme case is when all robots are collinear and placed on the diagonal of the portion of the plane occupied by the robots.

For the sake of analysis, we also consider a stricter model where robots are fully synchronized. That is, at each time instant, all robots are activated. Then, the number of steps of termination of the algorithm is given by the following equation:

$$S = \lceil \frac{\sqrt{X^2 + Y^2}}{\delta_{avg}} \rceil$$

Now if we consider a strict activation schedule, where robots are activated in mutual exclusion, i.e., at each time instant exactly one robot is activated at each time instant, then the maximum number of steps of termination of the algorithm is:

$$MaxS = \lceil \frac{n\sqrt{X^2 + Y^2}}{\delta_{avg}} \rceil$$

with n is the number of robots in the system.

4.6 Summary

In this chapter, we took a new look at the gathering of a group of oblivious mobile robots with limited visibility and no multiplicity detection. In particular, we have studied the solvability of gathering when robots are equipped with unstable compasses, and found that gathering can nevertheless be solved with such compasses in the semi-synchronous model SYm by providing a solution to the problem. Our algorithm can solve the gathering probabilistically when the compasses are unstable, and in finite time when compasses stabilize eventually for sufficiently long periods.

We have also shown that the algorithm of Flocchini et al. [34] based on perfect compasses cannot cope with instabilities of compasses in either SYm and Corda models.

The main benefit of our approach is its practical value. In particular, eventually consistent compasses allow the algorithm to tolerate transient faults, and also gives the algorithm the nice property of self-stabilization.

In the next chapter, we will investigate the solvability of the gathering problem in the asynchronous model Corda, relying on compasses that are eventually consistent, in a system where robots have limited visibility, and they are oblivious.

Chapter 5

Gathering with EVC Compasses in CORDA Model for Four Robots

In this chapter, we investigate the solvability of the gathering problem in the CORDA model, relying on eventually consistent compasses when robots are oblivious, and they have limited visibility. We first show that our gathering algorithm proposed for the SYm model with eventually consistent compasses (Chapter 4, Algorithm 1) solves the problem in the CORDA model for up to three robots in finite time. Then, we present a deterministic algorithm for solving the gathering problem in the CORDA model for a maximum of four robots.

5.1 Gathering with EVC Compasses in the CORDA Model for Three Robots

In this section, we show that Algorithm 1 (presented in Chapter 4) solves the gathering problem in the CORDA model for at most three robots. We first show that the connectivity of the distance graph is preserved during the entire execution of the algorithm. Second, we show that all robots gather in a finite time.

5.1.1 Safety: Preserved Connectivity

Lemma 5.1.1 *Let $\mathcal{U}$ be composed of two robots r and r' that are mutually visible at time t, and execute Algorithm 1. Then, at any time $\bar{t} > t$, r and r' are mutually visible or gather at the same point.*

PROOF. Recall that the compasses of the robots may disagree. Then, we consider three cases depending on the robots' movements.

In the first case, r and r' do not move if each of them sees the other robot to its left or above it. Thus, r and r' remain at the initial distance from each other, which is less than or equal to VR by the hypothesis.

In the second case, only one robot decides to move, say r. Then, by the algorithm, r moves on the segment $\overline{rr'}$ toward r'. Hence, $\forall p \in \overline{rr'}$, $dist(p, r') \leq VR$.

In the third case, both r and r' decide to move. Then, each robot will move toward the position of the other one. As a result, r and r' may swap positions, or stop anywhere before reaching their respective destinations, hence they remain within distance VR from each other.

In all three cases, both r and r' remain at a distance of at most VR from each other at any time $\bar{t} > t$, and the lemma holds. $\square_{\text{Lemma 5.1.1}}$

Lemma 5.1.2 *Let each pair of robots, (r, r') and (r, r'') be mutually visible at time t, and execute Algorithm 1. Then, at any time $\bar{t} > t$, r and r', and r and r'', are mutually visible or gather at the same point.*

PROOF.

The proof is straightforward. First, if all robots r, r' and r'' are collinear, then the proof holds by Lemma 5.1.1.

Second, if r, r' and r'' are non collinear, the proof derives from Lemma 4.4.3. $\square_{\text{Lemma 5.1.2}}$

The above two lemmas show that if two robots are mutually visible, they will continue to be so at future times. Consequently, we conclude that:

Theorem 5.1.3 *Under Algorithm 1, the distance graph is connected during the entire execution of the algorithm.*

5.1.2 Liveness: Termination of the Algorithm

In the following, we assume that the robots reach the time GST.

Lemma 5.1.4 *Let $\mathcal{U}$ be composed of two robots r and r' that are mutually visible at time t, and execute Algorithm 1. Then, r and r' will gather at the rightmost or bottommost robot in finite time.*

PROOF.

Assume without loss of generality that r is located on $Left_{r'}$ or on $\Psi^+_{r'}$. Then, by the algorithm, r' is prevented from moving because of r, and r will move toward r'.

First, assume r is located on $Left_{r'}$. Let r perform a look operation at time t, and let t' be the time when r finishes its move. Between time t and t', r' is prevented from moving because it sees r to its left. Hence, r' will stay still.

By Assumption 2.3.1, in one cycle, r moves by at least Δ_r, so the number of steps performed by r to reach r' is at most $dist(r, r')/\Delta_r$, which is finite. Thus, r reaches r' in a finite number of steps.

Similarly, if r lies on $\Psi^+_{r'}$, then by the algorithm, r' can not move because it sees r above it. Then, by using the same arguments, r will reach r' in a finite number of steps, and the lemma holds. $\square_{\text{Lemma 5.1.4}}$

Lemma 5.1.5 *Let the pairs of robots (r, r') and (r, r'') be mutually visible at time t, and execute Algorithm 1. Then, all three robots will gather at the rightmost and bottommost robot in finite time.*

The proof is a direct consequence from Lemma 4.4.5 and Lemma 4.4.6.

Theorem 5.1.6 *Algorithm 1 solves the gathering problem deterministically for at most three robots in the* CORDA *model, assuming eventually consistent compasses in the oblivious and limited visibility settings.*

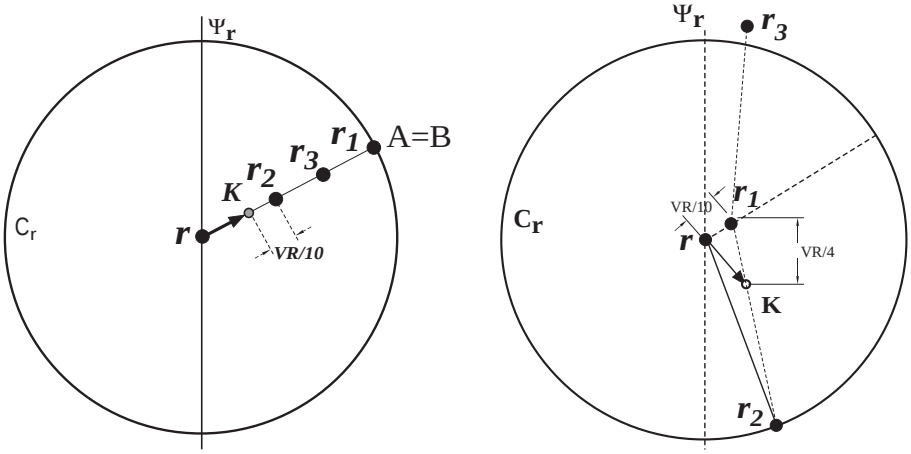
PROOF. From Theorem 5.1.3, the distance graph is connected during the entire execution of the algorithm. Moreover, by Lemma 5.1.4 and Lemma 5.1.5, all robots will gather in a finite time. This terminates the proof. $\square_{\text{Theorem 5.1.6}}$

5.2 Gathering with EVC Compasses in the CORDA Model for Four Robots

In this section, we present a deterministic algorithm for solving the gathering problem for a maximum of four robots in the asynchronous model CORDA, where robots are oblivious, they have limited visibility, and they are equipped with eventually consistent compasses. The algorithm is a simple adaptation of our algorithm proposed for the SYm model, which was presented in the previous chapter.

5.2.1 Algorithm Description

The intuition behind the algorithm is as follows: First, we let all the robots move in such a way as to achieve the following subgoals: (1) Robots that are mutually visible at time t remain mutually visible at time $t' > t$. (2) The global convex hull including all of the robots shrinks, until all of them become mutually visible. (3) When all robots in the system become mutually visible, only robots on the left side move toward the ones on the right side until they gather at the leftmost and bottommost robot when their compasses stabilize eventually. This algorithm is an adaptation of our algorithm proposed for the SYm model in Chapter 4. Note that our algorithm presented in Chapter 4 does not solve the gathering for four robots in the CORDA model assuming eventually consistent compasses. That algorithm can satisfy the safety property, by preserving the connectivity



(a) $\widehat{ArB} = 0$, then r moves to the point K at distance $VR/10$ from r_2.

(b) r computes r_1 and r_2 as outermost robots. $dist(r, r_3) = VR/10$ and $dist(r, r_2) > 9VR/10$. Then, r moves to the point K at distance $VR/4$ from r_1 on $\overline{r_1 r_2}$. K is within VR from all robots in $\mathcal{R}_r$.

Figure 5.1: The diameter of the convex hull of robot r is greater than $VR/10$.

of the distance graph between the four robots. However, it only converges toward the gathering, but it does not guarantee termination in a finite number of steps.

The gathering algorithm is depicted in Algorithm 2, where the functions Activation_Step($\mathcal{R}_r$, $compass_r$), Do_nothing(), and Move_to($Goal$) are as described in the previous chapter.

Algorithm 2 is described informally as follows. First, at every time instant t, where some robot r becomes active, it queries its compass and then, it considers all the robots in its visibility region $\mathcal{R}_r(t)$. As long as the diameter of the local convex hull of robot r, denoted by $LC(r, t)$ is greater than $VR/10$,[1] then r decides its movement as follows:

1. If r is not a vertex of $LC(r, t)$, then r does nothing. i.e., r stays still.

2. If r is collinear with all robots in $\mathcal{R}_r(t)$ and r is in the middle of the line, then r does nothing. Otherwise, if r is at an end of the line, then it moves toward the nearest robot to it, and stops at a distance of $VR/10$ from it (see Figure 5.1(a)).

3. If r is a vertex of $LC(r, t)$ and r is not collinear with all robots on $\mathcal{R}_r(t)$ then, r computes the positions of the *two outermost robots* as described in the previous chapter.

   Recall that, the two outermost robots are first the two robots that form the biggest central angle with r and then, they are at the largest distance from r.

   After computing the two outermost robots, robot r finds itself in one of the following two cases:

[1] The choice of this value is arbitrary, however it should be any value that is less than or equal to $VR/4$.

(a) If r is at distance less than or equal to $VR/10$ from one of the two outermost robots, say r_1, and r is at distance greater than $9VR/10$ from the other robot r_2, and the angle $\widehat{r_1 r r_2}$ is right or obtuse, then r moves to the point K on the segment $\overline{r_1 r_2}$, which is at distance $VR/4$ from r_1 if the distance from K to every robot in $\mathcal{R}_r$ is less than or equal to VR (refer to Figure 5.1(b)).

Else, r moves to the point H on the segment $\overline{r_1 K}$, such that H is at a distance less than or equal to VR from every robot in $\mathcal{R}_r$.

(b) Otherwise, r computes the height of the triangle that it forms with the two outermost robots, say r_1 and r_2, and having a base segment $\overline{r_1 r_2}$. Let H be the foot of a perpendicular starting at r. Then, r moves to H, if H is inside the triangle $\triangle(r, r_1, r_2)$. Otherwise, if H is outside $\triangle(r, r_1, r_2)$, then r moves to the closest robot to it, among r_1 and r_2.

When the diameter of the local convex hull of robots becomes less than $VR/10$, we allow only robots which are at the leftmost and topmost locations with respect to their local compasses to move. More specifically, if the diameter of $LC(r, t)$ is less than or equal to $VR/10$, then robot r decides its movement as follows:

1. *No move*: if r sees robots on its left side $Left_r$, or on Ψ_r^+, then r does not move (Algorithm 2, line 29).

2. *Vertical move*: if r sees robots below it (Ψ_r^-), then r moves toward the nearest robot below it (Algorithm 2, line 31).

3. *Right move*: If r sees robots on its right side $Right_r$, then, it moves to the nearest robot on $Right_r$ (Algorithm 2, line 33).

5.2.2 Correctness of the Algorithm

In this section, we prove the correctness of our algorithm in two steps. In the first step, we show that the connectivity of the distance graph is preserved during the entire execution of the algorithm. In the second step, we show that all robots will gather at one point in a finite number of steps. Before proceeding, let us recall an important lemma proved by Flocchini et al. [34]; if the initial *distance graph* is disconnected, the gathering problem is unsolvable. Hence, we assume that the initial distance graph is connected.

5.2.2.1 Safety: Preserved Connectivity

Trivially, the following lemma holds:

Lemma 5.2.1 *If the diameter of the local convex hull of each robot in the system is less than or equal to $VR/10$. Then, the diameter of the global convex hull forming the four robots is less than VR, and thus all robots are visible.*

Algorithm 2 Gathering with EVC Compasses in the CORDA Model for $n \leq 4$

Algorithm:

```
Activation_Step(R_r, compass_r)
if (|R_r| = 1) then {Gathering terminated; r sees only itself.}
  Do_nothing();
else
  if ((∃s ∈ R_r, dist(r,s) > VR/10) ∨ (∃s,s' ∈ R_r, dist(s,s') > VR/10)) then {The
  local convex hull of r is greater than VR/10.}
    ÂrB̂: biggest central angle of C_r with endpoints A and B;
    if (ÂrB̂ = π) then {robots are collinear and r is in the middle of the line.}
      Do_nothing();
    else if (ÂrB̂ = 0) then {r is on the extreme of the line.}
      s := nearest robot to r;
      Goal := the point on the segment rs at distance VR/10 from s (Figure 5.1(a));
    else {r computes two outermost robots s_1 and s_2.}
      s_1 := farthest robot from r on the segment rA;
      s_2 := farthest robot from r on the segment rB;
      if (dist(r,s_1) ≤ VR/10 ∧ dist(r,s_2) > 9VR/10 ∧ ŝ_1rs_2 ≥ 90°) then
        K := point on the segment s_1s_2 at distance VR/4 from s_1;
        Goal := farthest point H from s_1 on the segment s_1K, such that ∀s ∈ R_r,
        dist(s,H) ≤ VR (Figure 5.1(b));
      else
        H := foot of the height of the triangle △(r,s_1,s_2) starting from r;
        if (H ∈ △(r,s_1,s_2) ) then
          Goal := H;
        else {H is outside the triangle △(r,s_1,s_2)}
          Goal := nearest robot to r among s_1 and s_2;
        end if
      end if
    end if
  else {the local convex hull of r is less than or equal to VR/10}
    if (|Left_r| > 0 ∨ |Ψ_r^+| > 0) then {r sees robots on its left side or on Ψ_r^+}
      Do_nothing();
    else if |Ψ_r^-| > 0 then {r sees robots on its bottom side}
      Goal := nearest robot to r on Ψ_r^-;
    else {r sees robots on its right side }
      Goal := nearest robot to r on Right_r;
    end if
  end if
  Move_to(Goal);
end if
```

When the diameter of the global convex hull including all robots is less than or equal to VR. Obviously, the connectivity of the distance graph is preserved during the execution of Algorithm 2, because each robot tries to move toward another robot in the system, thus shrinking the distance between them. We thus, establish the following lemma:

Lemma 5.2.2 *The connectivity of the distance graph is preserved during the execution of Algorithm 2 when the global convex hull including all robots is less than or equal to VR.*

Lemma 5.2.3 *Let a pair of robots r_1 and r_2 be mutually visible at time t, such that $dist(r_1, r_2) \leq VR/10$. Then, by Algorithm 2, r_1 and r_2 can get farther away, up to a distance of less than $VR/2$ from each other.*

PROOF. The proof consists of showing the following cases:

- Neither r_1 nor r_2 sees an other robot in the system. Then, by the algorithm, either r_1 or r_2 moves to the other one, and thus they remain within the initial distance from each other.

- Both r_1 and r_2 see an other robot in the system (as depicted in Figure 5.2).

  We have r_1 and r_2 mutually visible at time t. Let also the pair of robots r_1 and r_3 be mutually visible, and r_2 and r_4 be mutually visible at time t.

  Observe that, the extreme case is when $dist(r_1, r_3) = VR$, and $\widehat{r_2 r_1 r_3} \geq 90°$. In addition, $dist(r_2, r_4) = VR$, and $\widehat{r_1 r_2 r_4} \geq 90°$. Then, by the algorithm, r_1 moves to the point K on the segment $\overline{r_2 r_3}$, which is at distance $VR/4$ from r_2. Similarly, r_2 moves to the point K' on the segment $\overline{r_1 r_4}$, which is at distance $VR/4$ from r_1. Since, r_1, r_2, r_3 and r_4 are not collinear by hypothesis, then r_1 and r_2 get away from each other by at most a distance less than two $VR/4$, that is by a distance up to less than $VR/2$.

  Trivially, the case also holds, when one robot, say r_1, gets farther away from r_2 by $VR/4$, and r_2 moves to the height of the triangle $\triangle(r_2, r_1, r_4)$ because r_2 will get farther away from its location by a distance less than $dist(r_1, r_2)$, which is equal to $VR/10$.

- Only robot r_1 sees other robots in the system. By the algorithm, r_2 moves toward r_1 or it does not move. Besides, r_1 can get farther away from r_2 by a maximum of $VR/4$, by the algorithm. Then, r_1 and r_2 remain within distance less than $VR/2$ from each other.

$\square_{\text{Lemma 5.2.3}}$

Lemma 5.2.4 *Let $S(t_0)$ be the set of robots visible to r at initial time t_0. Then, by Algorithm 2, at any time $t > t_0$, r is within distance VR from all robots in $S(t_0)$.*

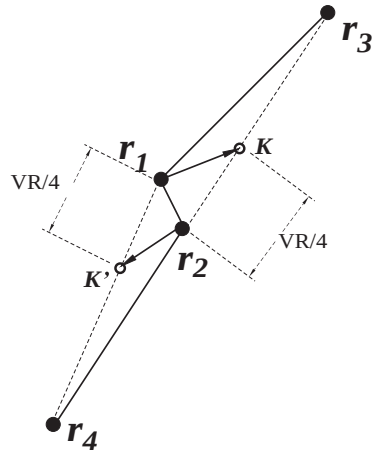


Figure 5.2: r_1 and r_2 can get farther away from each other up to a distance less than $VR/2$.

PROOF. Let $t \geq t_0$ be the time when r performs a look operation, and $\bar{t}$ be the time when the last robot in $S(t_0)$ finishes its move. Then, the proof consists of showing that $S(t_0) \subseteq S(\bar{t})$.

Trivially, the case of two robots holds, since one robot must move toward the other one. Then, both robots will remain at distance VR from each other.

In the following, we assume that the number of robots in $S(t_0)$ is greater than one. Let robot r be active at time t, and let t' be the time when r finishes its move. We distinguish the following cases depending on the movement of r and the movements of robots in $S(t_0)$.

1. *Robot r performs a look operation at time t; none of robots in $S(t_0)$ perform a look operation between t and t'.*
   The following cases arise:

   (a) *Robot r executes a null move.*
   If the destination computed by r at time t results in a null move, then r stays still. In addition, by hypothesis, $\forall s \in S(t_0)$, s is idle between t and t'. This means that at time $t' = \bar{t}$, r remains at the original distance from all robots in $S(t_0)$, which is by hypothesis less than or equal to VR.

   (b) *Robot r is collinear with all robots in $S(t_0)$ (i.e., $\widehat{ArB} = \{0, \pi\}$).*
   Two possibilities follow: (1) Robot r can be in case (a) above (i.e., it executes a null move), so the lemma holds for this case. (2) robot r performs a real move. Let r_1 be the robot farthest away from r on the line. By hypothesis, $dist(r, r_1) \leq VR$, and by the algorithm, r moves on the segment $\overline{rr_1}$. Thus, at time $t' = \bar{t}$, r gets closer in distance to all robots in $S(t_0)$.

   (c) *The diameter of $LC(r, t)$ is less than or equal to $VR/10$; r moves to the nearest robot to it.*

The diameter of $LC(r,t)$ is less than or equal to $VR/10$, this means that all robots in $S(t_0)$ are visible. By the algorithm, r moves to the nearest robot to it, and thus it remains visible to all robots in $S(t_0)$.

(d) *Robot r computes the positions of the two outermost robots.*
Let r_1 and r_2 denote the positions of the two outermost robots of r at time t, and let $\sphericalangle(ArB)$ be the circular sector enclosing all the robots in $S(t_0)$ such that $dist(r,A) = dist(r,B) = VR$, and $r_1 \in \overline{rA}$ and $r_2 \in \overline{rB}$.

Let I be the destination of r. If I is the foot of triangle $\triangle(r, r_1, r_2)$ or the position of r_1 or r_2. Then, the case holds by Lemma 4.4.2, where for every point p in $\sphericalangle(ArB)$, $dist(p,I) < VR$.

In the case where I is a point on the segment $\overline{r_1 r_2}$, such that, $\forall r_i \in \mathcal{R}_r)$, $dist(r_i, I) < VR$, then the lemma holds for this case by definition of the algorithm, and $\forall r_i \in S(t_0))$, $dist(r_i, I) < VR$. This terminates the proof.

2. *Robot r performs a look operation at time t; some or all robots in $S(t_0)$ perform a look operation between t and t'.*

Assume that $r' \in S(t_0)$ performs a look operation at time $t \leq t'' \leq t'$. Let I' be its destination, and $S'(t'')$ be the set of robots visible to r' at time t''. Assume without loss of generality that r' is the last robot to finish its move at time $\bar{t}$. Let also I be the destination of r. Thus, in the following, we will show that at any time $t \leq t''' \leq \bar{t}$, $\forall p \in \overline{rI}$, $\forall p' \in \overline{r'I'}$ $dist(p,p') \leq VR$.

We will only show the proof for the case of r', however the same arguments hold for the other robots in $S(t_0)$.

(a) *Robot r executes a* null move*; Robot r' executes any kind of move allowed by the algorithm*
This case is similar to case 1. above, thus omitted here.

(b) *Robot r is collinear with all robots in $S(t)$, and executes real move; Robot r' computes the positions of the two outermost robots s_1 and s_2.*

Let $\sphericalangle(A'r'B')$ be the circular sector of robot r' at time t''.

If the destination I' of r' is the height of the triangle $\triangle(r', s_1, s_2)$ or s_1 or s_2. Then, by Lemma 4.4.2, $I' \in \sphericalangle(A'r'B')$, and for every point $q \in \sphericalangle(A'r'B')$, we have, $dist(q, I') < VR$.

Since, $r \in \sphericalangle(A'r'B')$, and by the algorithm, $I \in \overline{rr'}$ (r moves on $\overline{rr'}$). Consequently, $\overline{rI} \subset \sphericalangle(A'r'B')$. Hence, by Lemma 4.4.2, $\forall p \in \overline{rI}$, $dist(p, I') \leq VR$, and $\forall p' \in \overline{r'I'}$, $dist(p,p') \leq VR$.

In the case where I' is a point on the segment $\overline{s_1 s_2}$ that satisfies $\forall s' \in S'(t'')$, $dist(s', I') \leq VR$, we have robot r is getting closer to r'. By hypothesis, we have $dist(r, r') \leq VR$, and by the algorithm $dist(r, I') \leq VR$, and $dist(r', I') \leq VR$. Thus, by Lemma 2.4.6 in triangle $\triangle(r, r', I')$, $\forall p \in \overline{rr'}$, $\forall p' \in \overline{r'I'}$, $dist(p, p') \leq VR$.

(c) *Both robot r and r' compute the positions of the two outermost robots.*
We distinguish the following two cases:

First, each robot move to the height of triangle that it forms with its outermost robots, or to one of the two outermost robots. Let r and r' be at the maximum distance VR from each other. Then, by the algorithm, I and I' belong to the circle with diameter $\overline{rr'}$. Consequently, $\forall p \in \overline{rI}$, $\forall p' \in \overline{r'I'}$, $dist(p, p') \leq VR$

Second, both robots r and r' are at distance less than or equal to $VR/10$. By Lemma 5.2.3, r and r' get farther away from each other by up to a distance less than $VR/2$. Thus, $\forall p \in \overline{rI}$, $\forall p' \in \overline{r'I'}$, $dist(p, p') \leq VR$.

(d) *The local convex hull of r and r' is less than $VR/10$*
In this case, obviously, the robots that are visible to r are also visible to r', and vise versa. By the algorithm, robot r (respectively, r') tries to go toward the nearest robot visible to it. Thus, they remain visible to each other during and after finishing their move. Hence, $\forall p \in \overline{rI}$, $\forall p' \in \overline{r'I'}$, $dist(p, p') \leq VR$.

In all cases, at time $\bar{t}$, $\forall p \in \overline{rI}$, $\forall p' \in \overline{r'I'}$, $dist(p, p') \leq VR$, and the rest follows by induction. This terminates the proof. $\square_{\text{Lemma 5.2.4}}$

From Lemma 5.2.4, we deduce the following theorem:

Theorem 5.2.5 *Algorithm 2 guarantees that the distance graph remains connected during the entire execution of the algorithm.*

5.2.2.2 Liveness: Termination of the Algorithm

We assume that the robots reach the global stabilization time GST.

Lemma 5.2.6 *Let $CH(t_0)$ be the global convex hull of all the robots at time t_0. Then, by Algorithm 2, at any time $t \geq t_0$, $CH(t) \subseteq CH(t_0)$.*

PROOF. From Theorem 5.2.5, robots that are mutually visible at time t_0 remain visible at any time $t \geq t_0$. Moreover, each robot moves inside its local convex hull, thus shrinking it. Also, robots that are vertices of $CH(t_0)$ will shrink the global convex hull by reducing their local convex hull. Consequently, the global convex hull shrinks, and then $CH(t) \subseteq CH(t_0)$. $\square_{\text{Lemma 5.2.6}}$

Lemma 5.2.7 *By Algorithm 2, there is a finite time after which all robots become mutually visible. In other words, there is a finite time, after which the diameter of the global convex hull including all robots is less than or equal to VR.*

PROOF. In the case, when the robots form a line, the proof is straightforward, because always the robot located at the extreme of the line, will move toward the one located in the middle of the line, thus shrinking the total distance between the two robots located at the end of the line. Since by Assumption 2.3.1, each robot moves by at least $\Delta_r > 0$. Then, the distance between the two robots at the end of the line can be reduced within VR in a finite number of steps.

In the case when robots form a convex polygon. By Lemma 5.2.6, the global convex hull including all robots shrinks over time. However, this is not enough to prove that the diameter of the global convex hull can become less than or equal to VR in finite number of steps.

Assume by contradiction that the diameter of the global convex hull including all robots cannot become less than or equal to VR in a finite number of steps. This means that robots that are vertices cannot shorten the diameter of the convex hull.

There are three cases to consider.

- Robot r is a vertex robot, and is visible to only one robot in the system, say r'. By the algorithm, if r is at distance greater than $VR/10$, then r can shorten this distance up to $VR/10$. Thus, obviously r reduces the diameter of the global convex hull including all robots. A contradiction.

- Robots r and r' are both vertices, and are mutually visible. If r and r' are at distance greater than $9\,VR/10$, and are visible to other robots in the system, then the proof consists of showing that the distance between r and r' will be shortened to at least $4\,VR/5$ in a finite number of steps. A contradiction.

□Lemma 5.2.7

Lemma 5.2.8 *Any configuration of four robots is transformed to the gathering, in finite time, by Algorithm 2.*

PROOF. By Lemma 5.2.7, there is a finite time after which all robots become mutually visible. By assumption, the robots have reached the time GST, and then their compasses become perfect. By the algorithm, robots on the left move to the robots on their right, and robots on top move toward the robots below. By assumption, the cycle of a robot is finite, then all robots gather at the rightmost and bottommost robot in finite time, and the lemma holds. □Lemma 5.2.8

We thus conclude the following theorem:

Theorem 5.2.9 *In a system, with four anonymous, oblivious mobile robots, with limited visibility, and equipped with eventually consistent compasses, the gathering problem is solvable deterministically in the* CORDA *model.*

5.3 Summary

In this chapter, we have shown that the gathering problem is solvable in the CORDA model for four robots when robots are equipped with eventually consistent compasses. In particular, we give an algorithm that gathers up to four robots at a single point if their compasses provide correct output after some unknown period of instability. The algorithm can solve gathering probabilistically when the compasses are inconsistent, and deterministically after compasses have stabilized for sufficiently long.

In addition, we have shown that our gathering algorithm proposed for the SYm model solves the gathering for up to three robots in the CORDA model, when robots are equipped with eventually consistent compasses. Thus, we can argue that *eventually consistent compasses* have the same computational power as *perfect compasses* for solving the gathering problem of up to three robots.

In the next chapter, we show that the problem is nevertheless impossible for a large number of robots, when they are equipped with unstable compasses.

Chapter 6

Impossibility of Gathering with EVC Compasses in CORDA Model for Large Number of Robots

The difficult is done at once;
the impossible takes a little longer.
Charles Alexandre de Calonne

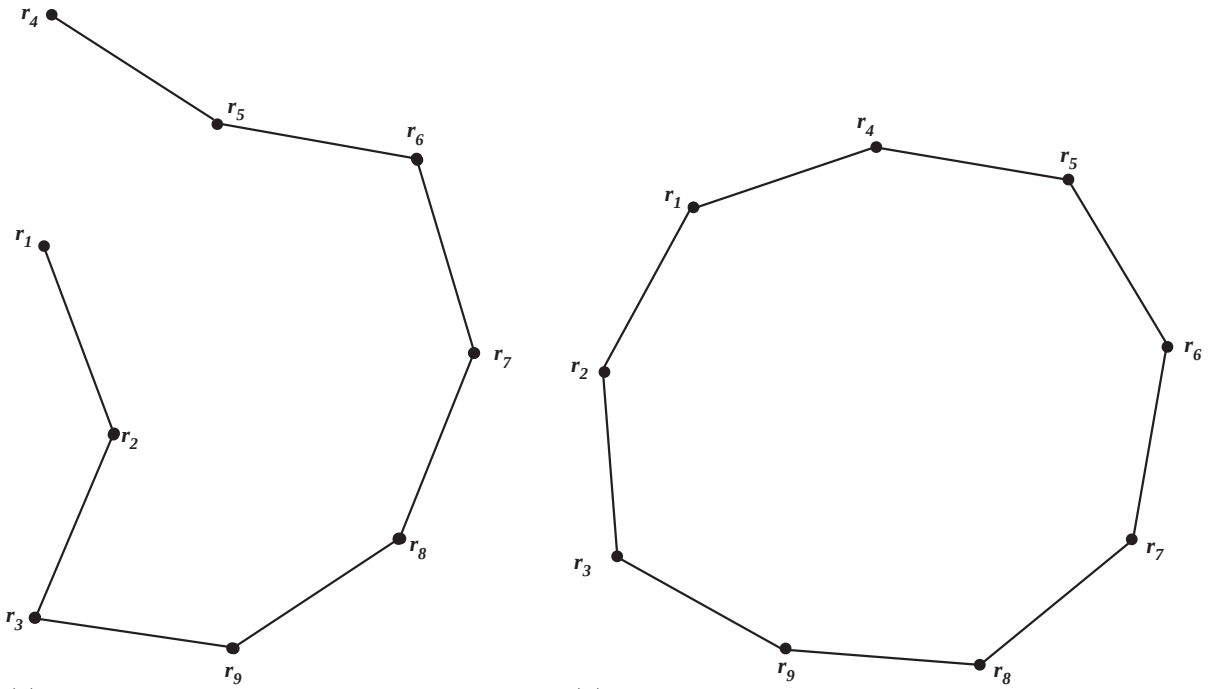
In the previous chapter, we presented an algorithm that gathers up to four robots in finite time in the CORDA model when robots share eventually consistent compasses, they are oblivious, and they have limited visibility. However, in this chapter, we show that the gathering problem has no solution in the CORDA model for nine or more robots when they are equipped with compasses that are unstable for some arbitrary periods.

Theorem 6.0.1 *In the* CORDA *model, there exists no oblivious algorithm that solves the gathering problem for any value n of robots ($n \geq 9$), when robots are equipped with eventually consistent compasses, have limited visibility and are unable to detect multiplicity.*

6.1 The Proof: General Idea

Recall that robots' compasses can be inconsistent before the Global Stabilization Time GST, and that the time GST is unknown to the robots. First, we want to stress that preserving the connectivity of the distance graph is a necessary condition to solve the problem. We recall Lemma 4.3.1, which states that if the initial distance graph $G(0)$ is disconnected, the gathering problem is unsolvable. This result holds also if initially the graph $G(0)$ is connected, and then it becomes disconnected at some time t; since the robots are oblivious, then $G(t)$ can be regarded as an initial distance graph.

To show that the problem is impossible to solve, we need to show that: (1) any algorithm satisfying the *safety* condition by preserving the connectivity of the distance graph, implies *no liveness*, that is no achievement of the gathering. (2) any algorithm



(a) Configuration $\mathbb{E}_1$: the robots do not form a cycle; each pair of visible robots are at distance VR from each other.

(b) Configuration $\mathbb{E}_2$: all robots form a cycle with side's length VR.

Figure 6.1: Impossibility of gathering in the CORDA model for $n \geq 9$ with EVC compasses.

satisfying the *liveness* condition by achieving the gathering, implies *no safety*, that is the disconnection of the distance graph between the robots. In other words, *liveness* and *safety* conditions are mutually exclusive; if the algorithm is *live*, then there exists a situation in which the safety is violated, and if the algorithm is *safe*, there exists a situation in which liveness is violated.

In particular, we will show that before GST, robots are unable to keep the distance graph $G(0)$ connected because of the inconsistencies of their compasses. Specifically, we define two configurations, $\mathbb{E}_1$ and $\mathbb{E}_2$, that we will use to defeat any possible algorithm $\mathcal{A}$. These two configurations are indistinguishable by some robots because of the limited visibility. Therefore, we will show that any algorithm that can be designed for one configuration will either lead to a deadlock situation (i,e. no progress toward gathering) or to a disconnection of the distance graph $G(0)$ in the other configuration (before GST).

In particular, we will show that starting from nine or more robots, there exists no oblivious algorithm that makes the robots keep the distance graph $G(0)$ connected while progressing toward gathering.

Before we proceed to the description of the proof in more detail, we establish the following lemmas.

Lemma 6.1.1 *The maximum distance travelled by a robot in one move must be less than VR in order to keep the distance graph connected.*

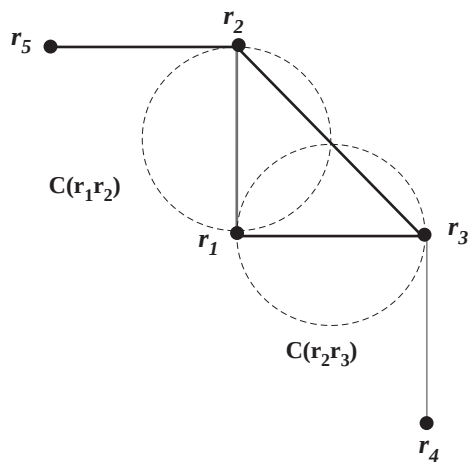


Figure 6.2: The destination of robot r_1 should be in the intersection of the two circles $C(\overline{r_1 r_2})$ and $C(\overline{r_1 r_3})$ in order to keep the vision connection with r_2 and r_3 if both of them move at the same time as r_1.

PROOF. The proof is trivial. Let r_1 and r_2 be two robots that are mutually visible at some time t, and are at distance $\delta > 0$. Assume by contradiction that r_1 and r_2 can move by distance VR in one cycle.

Consider robot r_1 and r_2 in the situation, where they did not reach yet the time GST, and they are activated simultaneously. Then, they can move in a completely opposite direction of each other because of the inconsistency of their compasses, and because they may see some other robots in the system. Consequently, r_1 and r_2 can get farther away from each other by a distance greater than VR, which results on a disconnection of the graph between r_1 and r_2. This terminates the proof. $\square_{\text{Lemma 6.1.1}}$

Lemma 6.1.2 *Given three robots r_1, r_2 and r_3 that are aligned, and are at distance VR from each other at time t, then any non null move executed by the robot in the middle of the line, say r_2, at time t will disconnect the distance graph (between r_1 and r_2 or between r_2 and r_3, or both r_2 and r_3) if neither r_1 nor r_3 move at time t.*

PROOF. Let Φ be the perpendicular to $\overline{r_1 r_3}$ passing through r_2. Let p be any point that belongs to Φ, such that $dist(p, r_2) > 0$.

First, assume that r_2 moves to $p \in \Phi$ at some time t. By hypothesis, $dist(r_1, r_2) = VR$, and r_1 does not move at time t. Then, by Lemma 2.4.8, $dist(p, r_1) > dist(r_1, r_2) = VR$. The same arguments holds for robot r_2.

Now, assume that r_2 moves to any point q that is to the left, or to the right of Φ at some time t, such that $dist(r_2, q) > 0$. Then, the triangle formed by $r_2(t)$, q, and either r_1 or r_3 is an obtuse angle at $r_2(t)$. Thus, by Lemma 2.4.7, $dist(r_1, q)$ or $dist(r_3, q)$ is larger than $dist(r_1, r_2)$ (i.e., larger than VR). This terminates the proof. $\square_{\text{Lemma 6.1.2}}$

Lemma 6.1.3 *Given a pair of robots r_1 and r_2 that are mutually visible at time t (time of the look operation performed by r_1), then the only way for r_1 to gather with r_2 is to move toward r_2 by a non null distance, or vise versa.*

The proof is trivial, thus omitted here.

Lemma 6.1.4 *Let three robots r_1, r_2 and r_3 be non collinear at some time t, with $\widehat{r_2 r_1 r_3} \geq \pi/2$, let H be the destination point computed by r_1, with $H \neq r_1$. Then, if H preserves the distance graph between r_2 and r_3, then H must belong to the triangle $\triangle(r_1, r_2, r_3)$ if both r_2 and r_3 move at time t.*

PROOF. Consider, the situation depicted in Figure 6.2, where each two visible robots are at distance VR from each other. Let r_1 perform a look operation at time t, and let H be its destination. Assume also that r_2 and r_3 perform a look operation at the same time t. Then, H must belong to the intersection of the two circles $C(\overline{r_1 r_2})$ and $C(\overline{r_1 r_3})$ in order to keep the vision connection with r_2 and r_3 if both of them move at the same time t. Consequently, H belongs to triangle $\triangle(r_1, r_2, r_3)$. This terminates the proof. $\square_{\text{Lemma 6.1.4}}$

6.2 The Impossibility Proof

As we stated above, the proof consists of showing that there is no *oblivious* algorithm $\mathcal{A}$ that starting from some configuration $\mathbb{E}$, lets the robots keep the distance graph $G(0)$ connected while progressing toward gathering. The proof is by contradiction. That is, we will assume that algorithm $\mathcal{A}$ exists and $\mathcal{A}$ is live, and then deduce that $\mathcal{A}$ is not safe.

Assume that $\mathcal{A}$ is an oblivious algorithm that starting from any configuration $\mathbb{E}$, preserves the connectivity of the distance graph before GST, and gathers the robots at one point in finite time after GST.

In the following, we assume that the system did not reach yet the time GST. Consider the two configurations $\mathbb{E}_1$ and $\mathbb{E}_2$ depicted in Figure 6.1, where each pair of robots connected with a line segment in the figure are mutually visible (i.e., they are at distance VR from each other). In particular, in configuration $\mathbb{E}_1$, robot r_1 and robot r_4 are not initially mutually visible. Also, robot r_1 and robot r_5 are not initially mutually visible.

We will show that there is no algorithm that can cope with these two configurations at the same time. For instance, these two configurations are indistinguishable by robot r_2 because of the limited visibility (r_2 can not know if the graph is a cycle or not).

Note that, since the algorithm $\mathcal{A}$ is *oblivious*, the computations done by the robots are based only on their current observations. Also, observe that algorithm $\mathcal{A}$ can not rely on the direction of the compasses to ensure safety, that is preserving the connectivity of the distance graph because compasses can point to any direction before GST.

We first give a lemma that states the different possible choices that algorithm $\mathcal{A}$ can make on the movements of robots in order to preserve the connectivity of the distance graph, and to gather them in finite time in configurations $\mathbb{E}_1$ and $\mathbb{E}_2$.

Lemma 6.2.1 *Given the robots in configurations $\mathbb{E}_1$ and $\mathbb{E}_2$, and an algorithm $\mathcal{A}$ for gathering these robots, then the only possible movements allowed by $\mathcal{A}$ are the following:*

1. *Algorithm $\mathcal{A}$ prevents all robots from moving.*
2. *Algorithm $\mathcal{A}$ allows robots to move according to some criteria.*

We will prove that for each case in the above lemma, we reach a contradiction.

- ***Case** 1**: Algorithm $\mathcal{A}$ prevents all robots from moving.***

  In this case, obviously, the robots will remain in the initial configuration $\mathbb{E}_1$, or $\mathbb{E}_2$ forever. Thus, $\mathcal{A}$ does not solve the gathering. A contradiction.

- ***Case** 2**: Algorithm $\mathcal{A}$ allows robots to move.***

  Assume that algorithm $\mathcal{A}$ is designed in a way that robots that can see exactly one robot at the time of their look operation can move, and others are prevented from moving. Then, it is easy to design an algorithm $\mathcal{A}$ that preserves the connectivity of the distance graph, and also it solves the gathering in configuration $\mathbb{E}_1$. However, if we consider configuration $\mathbb{E}_2$ (Figure 6.1(b)), where the robots form a polygon with sides' length VR, then algorithm $\mathcal{A}$ prevents all robots in configuration $\mathbb{E}_2$ from making any progress, which results in a global deadlock, since all the robots execute the same algorithm $\mathcal{A}$ by hypothesis. A contradiction.

  Now, we will consider that algorithm $\mathcal{A}$ allows robots that can see one or more robots at the time of their look operation to move, and the movements are devised by algorithm $\mathcal{A}$ according to some criteria in such away that they satisfy both preserving the connectivity of distance graph (safety) and progress toward gathering (liveness). We will show by contradiction that algorithm $\mathcal{A}$ is *not safe*.

  Todo so, we assume that $\mathcal{A}$ is safe. We also assume that there is an adversary in the system that plays against algorithm $\mathcal{A}$. The adversary can control: (1) the activation schedule; that is which robots in the system are activated at some time t. (2) the directions of compasses before GST. (3) initial configurations.

  In this case, in particular, we will consider mainly the behavior of r_1 and r_2 in configuration $\mathbb{E}_1$, and show that r_1 and r_2 can not make an appropriate decision to keep sight of each other while performing the gathering by any possible algorithm $\mathcal{A}$ given a specified *scenario*.

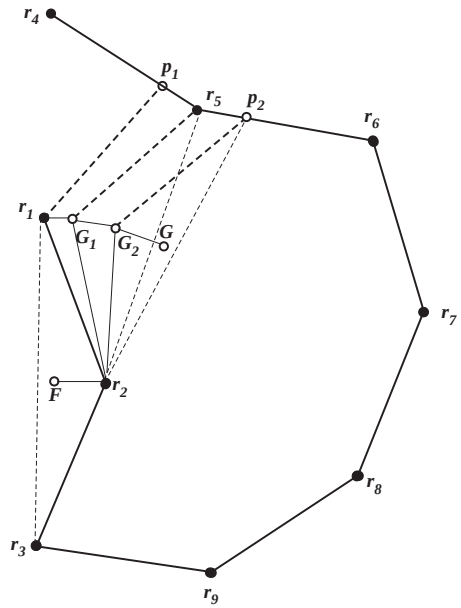


Figure 6.3: r_1 performs several activation cycles, and reaches the point G, while r_2 performs only one activation cycle, and moves to F.

In particular, we will show that the distance of the segment connecting robots r_1 and r_2, which is initially equal to VR, will become greater than VR given a specific scenario by any possible algorithm $\mathcal{A}$.

Since algorithm $\mathcal{A}$ needs to preserve the vision connectivity between r_1 and r_2, then, without loss of generality, it is reasonable to make the following assumption:

Assumption 6.2.2 *We assume that the destination point computed by robot r_1 always makes r_1 gets closer to r_2, while it preserves the vision connectivity with the other robots visible to r_1 at the time of the look operation. In other words, robot r_1 chooses the worst point for the adversary.*

If given the above assumption, we show that algorithm $\mathcal{A}$ fails to keep the vision connectivity between r_1 and r_2, then obviously $\mathcal{A}$ fails also to preserve the connectivity between these two robots if the destination of r_1 makes it move farther away from r_2. Now, consider the following scenario constructed by the adversary on configuration $\mathbb{E}_1$. The complete scenario is also depicted from Figure 6.5 to Figure 6.9 Recall that r_1 and r_4 are not mutually visible initially. Also, r_1 and r_5 are not mutually visible initially.

1. Each time robot r_1 performs a look operation, it will see two robots (although initially r_1 sees only r_2) because robots can see each other while moving by assumption in the CORDA model.
2. Robot r_1 will perform many activation cycles, while robot r_2 will perform only one activation cycle.

3. Let r_4 perform a look operation at time t_0. By Lemma 6.1.3, algorithm $\mathcal{A}$ makes r_4 move toward r_5. Let t_2 be the time when r_4 passes by the point p_1 (see Figure 6.3).

4. Now, let r_1 perform a look operation at time t_2, and sees r_4 at point p_1. Also, r_1 sees robot r_2. Let G_1 be the destination of robot r_1 given by algorithm $\mathcal{A}$. G_1 needs to preserve the connectivity of the graph between r_4 (at point p_1) and r_2. Then, by Lemma 6.1.4, $G_1 \in \triangle(r_1, r_2, p_1)$.

5. Let also t_2 be the time when r_2 performs a look operation, and computes as destination F by algorithm $\mathcal{A}$. Similarly, by Lemma 6.1.4, $F \in \triangle(r_1, r_2, r_3)$.

6. Let t_3 be the time when r_4 reaches r_5. Assume that $\forall t > t_3$, r_4 and r_5 are always activated together (we shall refer them by r_5 starting from time t_3).

7. At the point G_1, the distance $dist(G_1, r_5) = VR$.

8. The adversary sets an activation schedule, such that at time $t_4 > t_3$, r_5 (also r_4) is activated before robot r_1 reaches G_1, and thus, r_5 computes as destination r_6 by algorithm $\mathcal{A}$.

9. The adversary makes robot r_2 stay in an idle state for some finite time (during which robot r_1 performs $n > 1$ activation cycles) before it starts moving toward its destination F.

10. The adversary sets an activation schedule, such that when robot r_1 reaches G_1 (at time $t_5 > t_4$), it gets activated again and sees r_5 before it starts moving to r_6. Also r_1 sees r_2 since r_2 did not start moving yet. Let G_2 be the new destination computed by robot r_1 according to algorithm $\mathcal{A}$. Then, by Lemma 6.1.4, $G_2 \in \triangle(G_1, r_2, r_5)$.

11. The adversary sets an activation schedule, such that when r_1 reaches G_2, say at time $t_6 > t_5$, it gets activated again, and sees r_5 (while r_5 is moving toward r_6) at some point, say p_2 at distance VR from it. Then, r_1 computes its new destination point G according to algorithm $\mathcal{A}$, which belongs to the triangle $\triangle(G_2, p_2, r_2)$ by Lemma 6.1.4.

12. The adversary sets the initial configuration $\mathbb{E}_1$, such that when r_1 reaches G, it can not see the other robots in the system (for instance, r_6, r_7, r_8, r_9, r_3). Let $\bar{t}$ be the time when r_1 reaches the point G.

13. The adversary sets an activation schedule such that robots r_6, r_7, r_8, r_9, and r_3 are not activated between t_0 and $\bar{t}$.

14. The adversary sets the activation schedule such that, r_2 starts its move at time $t_7 > t_6$, and finishes moving toward its destination F at time $\bar{t}$.

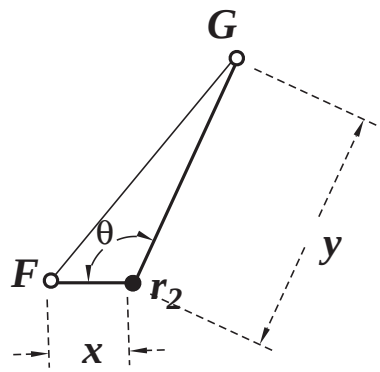


Figure 6.4: Illustration of the situation of r_1 and r_2 at time $\bar{t}$.

If algorithm $\mathcal{A}$ exists, it must guarantee that the distance graph remains connected between r_1 and r_2 at time $\bar{t}$, given the scenario described above. We will show that any algorithm $\mathcal{A}$ fails to deal with this scenario.

Before we proceed, we state the following lemma:

Lemma 6.2.3 *Given the scenario above, where F is the destination of robot r_2, and G is the destination of robot r_1 after performing $n \geq 3$ activation cycles, then in finite number of steps, $\widehat{Fr_2G} \geq \pi/2$.*

PROOF. Let $C(\overline{r_2r_3})$ the circle with diameter $\overline{r_2r_3}$. Let also Ψ, the line tangent to $C(\overline{r_2r_3})$ passing through r_2. By construction, we can have the segment $\overline{r_5r_2}$ perpendicular to Ψ. By Lemma 6.1.4, $F \in C(\overline{r_2r_3})$. Then, F is below Ψ. Consequently, $\widehat{Fr_2r_5} > \pi/2$.

Now, the proof consists of showing that the point G computed by robot r_1 is beyond the segment $\overline{r_2r_5}$, and that this point G can be reached by robot r_1 in finite number of steps.

Given the scenario described above, when r_1 reaches the point G_2, it can be activated many times as long as r_5 is moving toward r_6. Then, robot r_1 can compute a destination G which is beyond the line segment $\overline{r_4r_2}$ in a finite time. Thus, in finite number of steps, $\widehat{Fr_2G}$ is greater or equal to $\pi/2$. $\square_{\text{Lemma 6.2.3}}$

Let x be the distance travelled by robot r_2 in order to reach the point F at time $\bar{t}$. That is, $x = dist(r_2(t_2), F)$. Let also y be the distance between r_1 at time $\bar{t}$ and r_2 at time t_2. That is $y = dist(r_2(t_2), G))$ (refer to Figure 6.4).

Assume by contradiction that there exists $\mathcal{A}$ that preserves the connectivity of the distance graph between r_1 and r_2, assuming the above scenario. This means that the following lemma must be satisfied:

Lemma 6.2.4 *For each distance x travelled by robot r_2 (to reach F), there exists y, such that the distance between F and G is less than or equal to VR.*

In the following, we will show that Lemma 6.2.4 does not hold. To do so, we will show the following:

$$\forall x, \exists y \text{ such that, } y > VR - x \quad \text{, and} \tag{6.1}$$

$$y < VR \tag{6.2}$$

Condition 6.2 means that, although by algorithm $\mathcal{A}$, r_1 is getting closer to r_2 (Assumption 6.2.2), $\mathcal{A}$ fails to satisfy Lemma 6.2.4. Obviously, the lemma does not hold also if $\mathcal{A}$ makes r_1 move farther away from r_2.

By Lemma 6.1.1, we have $x < VR$. Then, in the following, we will assume that: $x = VR/K$, with $K > 1$. Consider the triangle $\triangle(F, r_2, G)$ (depicted in Figure 6.4). Let $\theta = \widehat{Fr_2G}$, then:

$$\begin{aligned} dist(F,G)^2 &= x^2 + y^2 - 2.x.y\cos\theta \\ dist(F,G)^2 &= (\frac{VR}{K})^2 + y^2 - 2\frac{VR}{K}.y\cos\theta \\ dist(F,G)^2 &= \frac{VR^2}{K^2} + y^2 - 2\frac{VR}{K}.y\cos\theta \end{aligned}$$

Assume that: $dist(F,G) = \delta.VR$, with $\delta > 0$. Then:

$$dist(F,G)^2 = \frac{VR^2}{K^2} + y^2 - 2\frac{VR}{K}.y\cos\theta \tag{6.3}$$

$$\delta^2.VR^2 = \frac{VR^2}{K^2} + y^2 - 2\frac{VR}{K}.y\cos\theta \tag{6.4}$$

By Lemma 6.2.3, $\theta \geq \pi/2$. We take the example where $\theta = \pi/2$. Then, $\cos\theta = 0$. By replacing $\cos\theta$ in Equation 6.4, we get:

$$\begin{aligned} \delta^2.VR^2 &= \frac{VR^2}{K^2} + y^2 - 2\frac{VR}{K}.y\cos\theta \\ \delta^2.VR^2 &= \frac{VR^2}{K^2} + y^2 \quad \text{, this implies that:} \end{aligned}$$

$$y^2 = \delta^2.VR^2 - \frac{VR^2}{K^2} \quad ,$$

$$y^2 = \frac{K^2.\delta^2.VR^2 - VR^2}{K^2} \quad ,$$

$$y^2 = \frac{VR^2(K^2.\delta^2 - 1)}{K^2}$$

Since the distance y is greater than zero, then:

$$y = \frac{VR\sqrt{K^2.\delta^2 - 1}}{K} \tag{6.5}$$

It follows that,

$$\frac{VR\sqrt{K^2.\delta^2 - 1}}{K} > 0 \tag{6.6}$$

$$K^2.\delta^2 - 1 > 0 \tag{6.7}$$

$$\delta > \frac{1}{K} \tag{6.8}$$

By assumption $K > 1$, and $\delta > 0$, then Equation 6.8 is true. Consequently, $y > VR - x$ exists, and Equation 6.1 is true.

Now, we will show that Equation 6.2 is also true.

Assume that $y < VR$, then from Equation 6.5, we get:

$$\frac{VR\sqrt{K^2\delta^2 - 1}}{K} < VR \tag{6.9}$$

$$\sqrt{K^2\delta^2 - 1} < K \tag{6.10}$$

$$\delta^2 < \frac{K^2 + 1}{K^2} \tag{6.11}$$

$$\delta < \frac{\sqrt{K^2 + 1}}{K} \tag{6.12}$$

By assumption, $K > 1$ and $\delta > 0$, then Equation 6.12 is true. So, Equation 6.2 is also true.

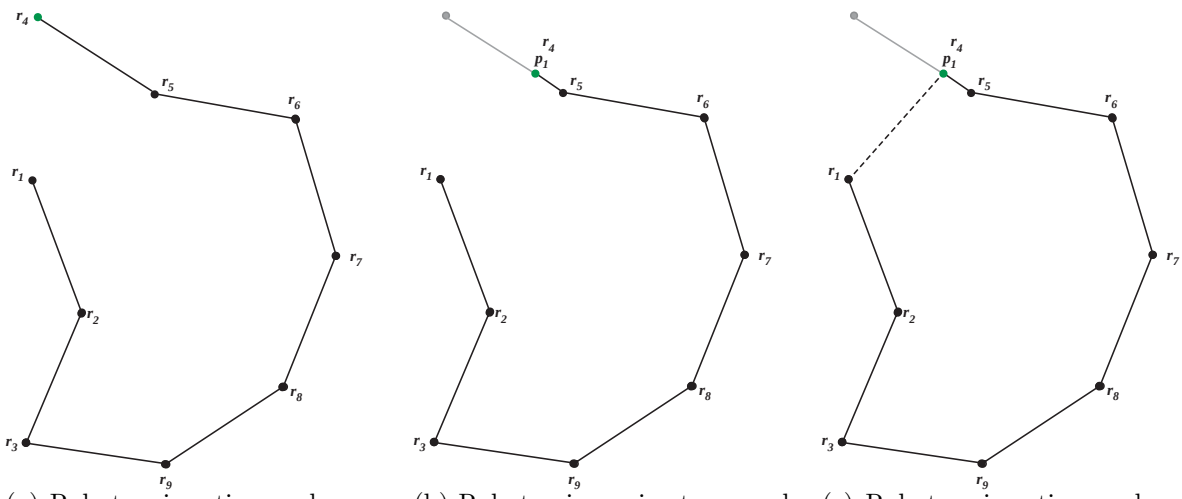
As a result, we conclude that Equation 6.1, and Equation 6.2 are true when $\theta = \pi/2$. Therefore, Lemma 6.2.4 does not hold. In other words, algorithm $\mathcal{A}$ is not safe because it fails to keep the vision connectivity between r_1 and r_2 given the scenario described earlier. A contradiction.

Lemma 6.2.1 covers every possible case that algorithm $\mathcal{A}$ can make on the movements of robots in order to gather them in configuration $\mathbb{E}_1$ and $\mathbb{E}_2$. However, in each case, we have shown a contradiction, where algorithm $\mathcal{A}$ fails to keep the distance graph connected between the robots. Thus, by Lemma 4.3.1, algorithm $\mathcal{A}$ does not solve the gathering problem. Consequently, algorithm $\mathcal{A}$ does not exist.

We have proved the case for nine robots, where nine robots are required to construct a distance graph connected in such away that when robot r_1 disconnect from robot r_2, it does not reconnect (see) with any of the other robots in the system. The same proof holds for ten or more robots. This terminates the proof.

6.3 Summary

In this chapter, we have shown that there are a certain number of robots from which it is impossible to achieve the gathering of asynchronous mobile robots when they are equipped with compasses that are unstable for some arbitrary periods, they are oblivious and they have limited visibility.

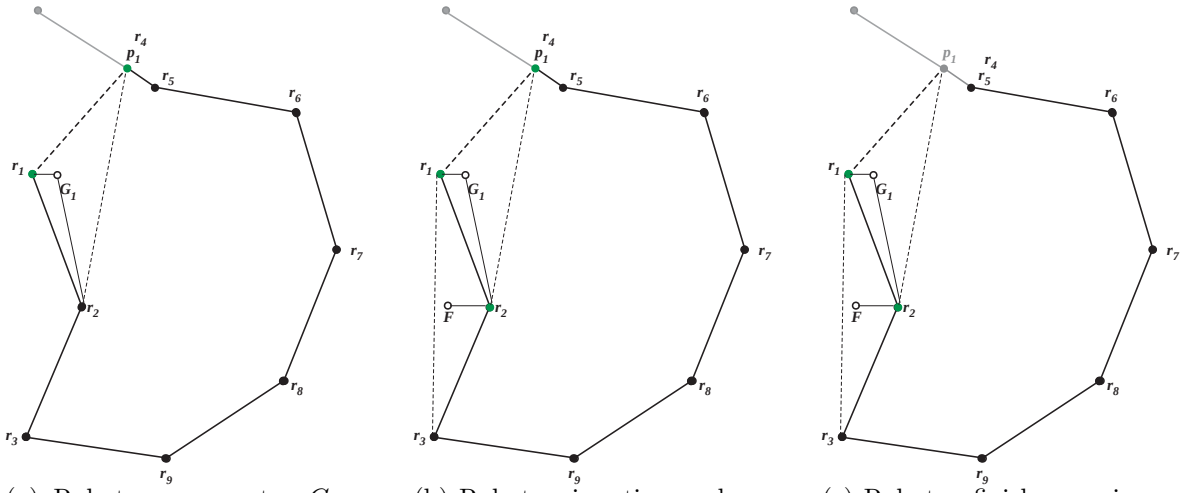


(a) Robot r_4 is active, and computes as destination r_5.

(b) Robot r_4 is moving to r_5, and passes by point p_1.

(c) Robot r_1 is active, and sees r_2 and r_4.

Figure 6.5: Scenario described in the proof.

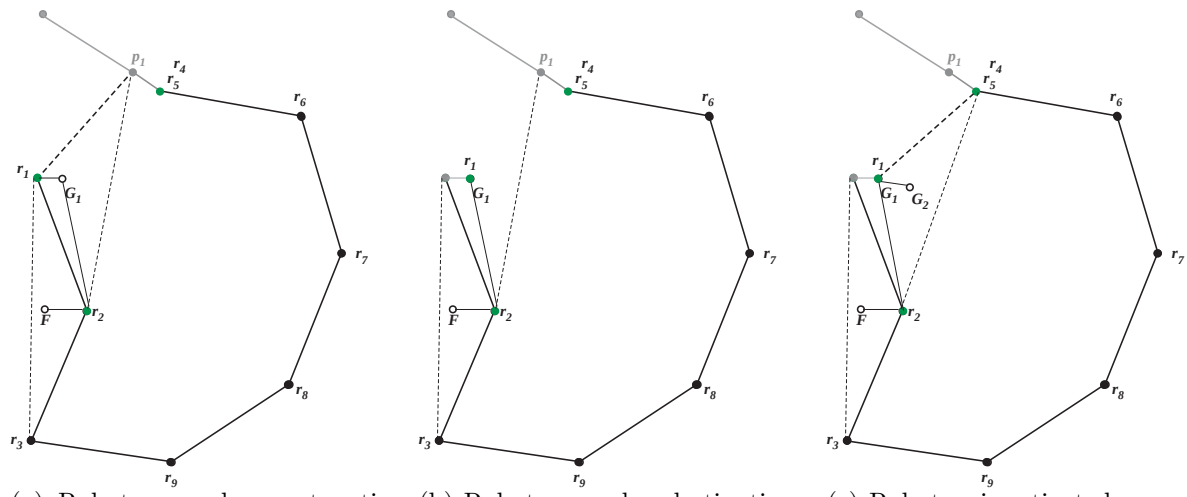


(a) Robot r_1 computes G_1 as destination.

(b) Robot r_2 is active, and computes as F as destination.

(c) Robot r_4 finishes moving and reaches r_5.

Figure 6.6: Continuation (1) of the scenario.

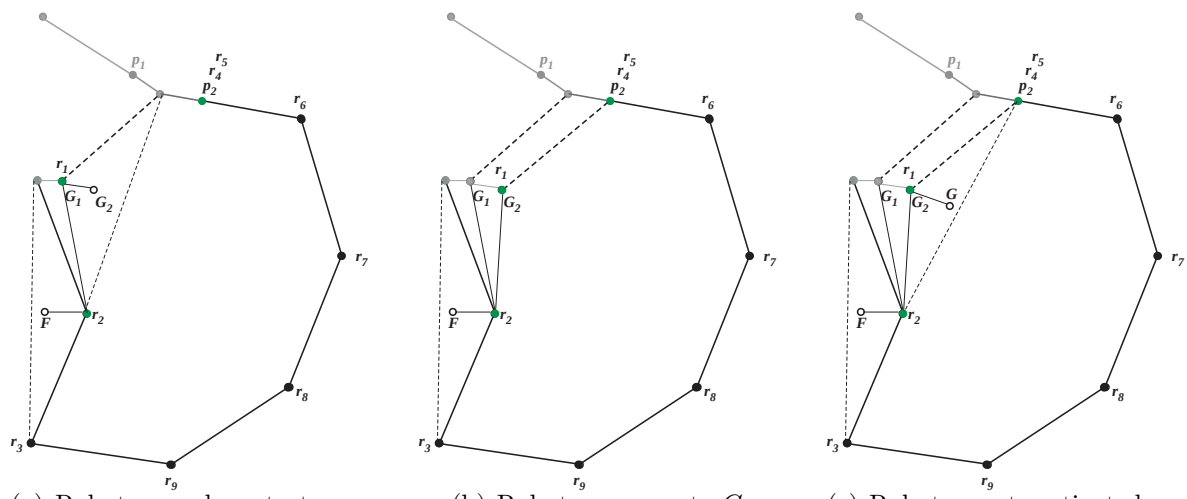


(a) Robot r_4 and r_5 get activated, and computes as destination r_6.

(b) Robot r_1 reaches destination G_1.

(c) Robot r_1 is activated, sees r_4 (r_5) and r_2, and computes as destination G_2.

Figure 6.7: Continuation (2) of the scenario.

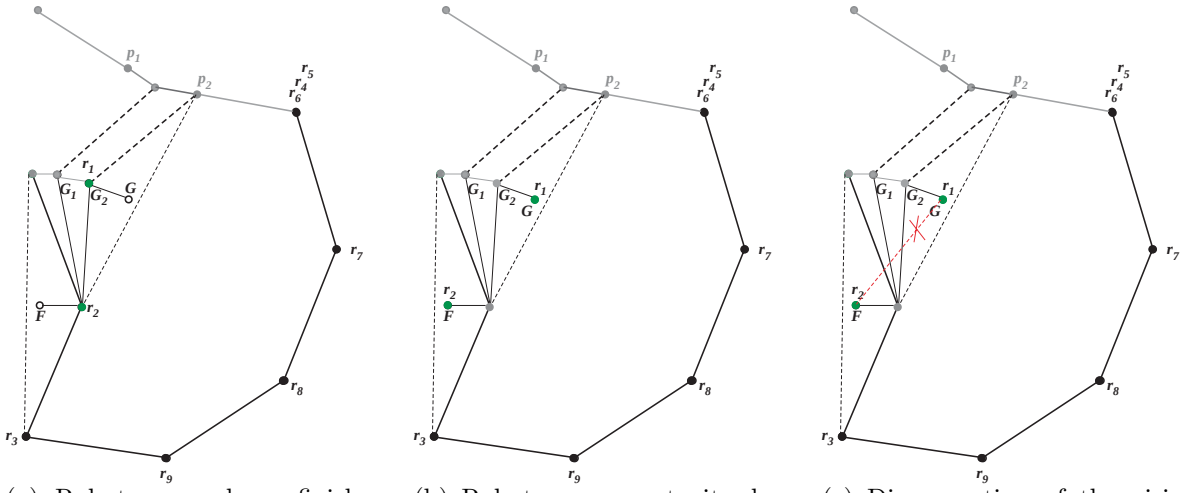


(a) Robot r_4 and r_5 starts moving, and reaches some point p_2.

(b) Robot r_1 moves to G_2.

(c) Robot r_1 gets activated, sees r_4 (r_5) at position p_2, and r_2, and computes as destination G.

Figure 6.8: Continuation (3) of the scenario.



(a) Robot r_4 and r_5 finishes moving to their destination r_6.

(b) Robot r_1 moves to its destination G, and also robot r_2 moves to its destination F.

(c) Disconnection of the vision graph between r_1 and r_2. Also, r_1 can not reconnect with other robots in the graph. Thus, r_1 is isolated and the vision graph is partitioned.

Figure 6.9: Continuation (4) of the scenario.

Chapter 7

Gathering Two Mobile Robots with 45° Inaccurate Compasses

A little inaccuracy sometimes
saves tons of explanation.
Hector Hugh Munro

In the two previous chapters, we concentrated on the gathering of oblivious mobile robots when they share compasses that are subject to instabilities in both SYm and Corda models. In this chapter, we look at the solvability of the gathering problem with another kind of unreliable compasses, called inaccurate compasses, in which compasses can be subject to inaccuracies and errors.

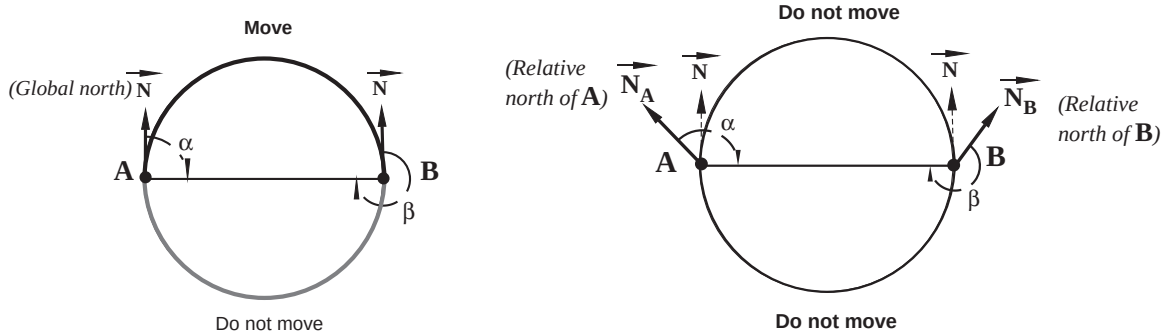
Prencipe [59] has shown that gathering stateless robots cannot be done deterministically without some additional assumptions. For instance, gathering is possible if robots share a common direction, as given by *perfect* compasses. Similarly, if robots can detect multiplicity (i.e., count robots that share the same location) gathering is possible for *three or more* robots.

This work is motivated by the pragmatic considerations that (1) compasses are error-prone devices in reality, and (2) multiplicity detection, while being easy to achieve, allows for gathering in situations with more than two robots.

Therefore, in this chapter we focus on the gathering of two asynchronous mobile robots equipped with *inaccurate* compasses. In particular, we provide a self-stabilizing algorithm to gather, in a finite time, two oblivious robots equipped with compasses that can differ by as much as $\pi/4$ assuming the Corda model.

7.1 Difficulty of Gathering with Inaccurate Compasses

In the Corda model, it is difficult to gather two robots or compare them in a consistent manner when they are equipped with inaccurate compasses. This is mainly due to the issue of breaking the symmetry between these robots. Let us illustrate this point using a simple



(a) The north of A and B are consistent.

(b) The north of A and B are inconsistent.

Figure 7.1: Difficulty of gathering with inaccurate compasses.

example. Assume that there exists a naive algorithm for comparing two asynchronous robots A and B in a consistent manner when their compasses are inaccurate. First, consider that A and B are equipped with accurate compasses, and place them at the two endpoints of a horizontal diameter of a unit circle. Then, a naive algorithm can be based on the comparison of the angles that A and B form respectively with some global North $\overrightarrow{N}$ (i.e., they share the same north) and the segment $\overline{AB}$ in clockwise direction. For instance, if the angle is less than or equal to $\pi/2$, the robot moves. Otherwise, if the angle is greater than $\pi/2$, the robot stays still. Then, a robot, say A, moves (see Figure 7.1(a)). Then, we rotate the diameter to exchange the positions of A and B. Now B moves. We thus color the perimeter of the circle Move and Do not move, where at any point which is colored Move or Do not move, A moves or stays still. Then, there is a point p that is a boundary between a Move and a Do not move segment. By introducing error to their compasses, at p, we can derive a contradiction. That is, we can not decide which robot moves, and which one stays still (see Figure 7.1(b)).[1]

7.2 Gathering Algorithm

The basic intuition behind the algorithm is to break the symmetry between two robots, that is, to forbid symmetric configurations of two robots. More precisely, with a perfect compass, it is easy to break the symmetry between two robots, for instance, by making one robot move and the other remain stationary. However, with inaccurate compasses, it is difficult to design an algorithm that breaks the symmetry between the two, as they can end up in a situation in which neither moves, which results in a deadlock situation, or in a situation in which both move in such a way that they continue moving forever. In order to avoid such situations, it is first necessary to ensure that the two robots do not see each other in the same zone.

[1]The argument is similar to the bi-valent argument in the impossibility result of the consensus problem [30].

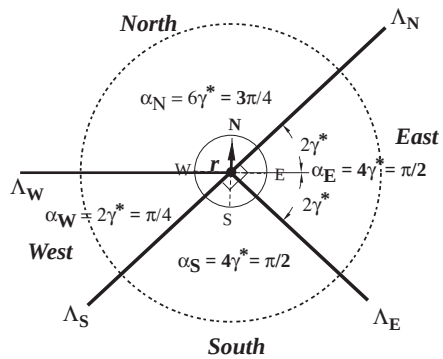


Figure 7.2: The four sectors *North*, *South*, *East* and *West* for robot r.

The main idea of our algorithm is to make each robot partition the plane into four different zones, so that two similar zones for two different robots should not overlap. Then, depending on the different possible configurations (resulting from the partitions) of the two robots, we design their movements such that a configuration is transformed to gathering, or to an intermediate configuration leading to gathering, without introducing cycles between configurations or deadlock situations.

Before we describe the algorithm in more detail, we will first explain how robots divide the plane.

7.2.1 Partitions

First, a robot needs to partition the plane into four sectors that do not overlap, namely the *North*, *South*, *East* and *West* sectors. Let α_N, α_S, α_E and α_W be the respective angular measurements of these sectors. Also, by Λ_N, Λ_S, Λ_E and Λ_W, we denote the rays delimiting these sectors, respectively (refer to Figure 7.2).

Now, let us assume there exits a constant $\gamma^* \geq 0$ that represents the maximum angle inaccuracy between the relative north $\overrightarrow{N_r}$ of some robot r and the absolute north $\overrightarrow{\mathcal{N}}$. Then, the following conditions must be satisfied in order to avoid a situation in which both robots see each other in the same sector because of compass inconsistencies.

$$\alpha_N \leq \pi - 2\gamma^* \tag{7.1}$$

$$\alpha_S \leq \pi - 2\gamma^* \tag{7.2}$$

$$\alpha_E \leq \pi - 2\gamma^* \tag{7.3}$$

$$\alpha_W \leq \pi - 2\gamma^* \tag{7.4}$$

We further set the following conditions on the sectors. These conditions will help to avoid the occurrence of infinite executions, i.e., having robots looping in the same configuration.

$$\alpha_E + \alpha_S \leq \pi \tag{7.5}$$

$$\alpha_N + \alpha_W \leq \pi \tag{7.6}$$

By summation of Equation (7.1) and Equation (7.5), we get:

$$\begin{aligned}
\alpha_N + \alpha_E + \alpha_S &\leq 2\pi - 2\gamma^* \quad \text{then,} \\
\alpha_N + \alpha_E + \alpha_S + \alpha_W &\leq 2\pi - 2\gamma^* + \alpha_W \\
2\pi &\leq 2\pi - 2\gamma^* + \alpha_W \\
2\gamma^* &\leq \alpha_W
\end{aligned}$$

After finding the condition in the *West* sector, we choose the minimal value for α_W. That is, $\alpha_W = 2\gamma^*$. Then, by summation of Equation (7.1), and Equation (7.2), we get:

$$\begin{aligned}
\alpha_N + \alpha_S &\leq 2\pi - 4\gamma^* \quad \text{then,} \\
\alpha_N + \alpha_S + \alpha_E &\leq 2\pi - 4\gamma^* + \alpha_E \\
\text{By hypothesis,} \quad \alpha_N + \alpha_S + \alpha_E &\leq 2\pi \quad \text{then, by subtraction, we get:} \\
0 &\leq -4\gamma^* + \alpha_E \quad \text{then,} \\
4\gamma^* &\leq \alpha_E
\end{aligned}$$

Thus, we choose $\alpha_E = 4\gamma^* = \alpha_S = \pi/2$ (From Equation (7.5)). This means that $\gamma^* = \pi/8$. It follows that, $\alpha_W = 2\gamma^* = \pi/4$. Finally, from Equation (7.1), and the fact that the sum of the four sectors is equal to 2π, we get, $\alpha_N = \pi - 2\gamma^* = 3\pi/4$. We have derived the condition that $\gamma^* \leq \pi/8$. Thus, in the remainder of the paper, we consider the largest inaccuracy value of γ^*, i.e., $\gamma^* = \pi/8$.

We now describe in more detail the features of each sector, as follows:

- *East*(r) sector: it is centered at r, has the East direction (indicated by its compass) $\overrightarrow{E_r}$ as bisector, and its angular sector α_E is equal to $4\gamma^*$, which is $\pi/2$. Note that *East*(r) is delimited by $\Lambda_N(r)$ and $\Lambda_E(r)$. However, only $\Lambda_E(r)$ is a part of *East*(r).

- *South*(r) sector: it is adjacent to *East*(r) in clockwise direction, and its angular sector α_S is equal to α_E, which is equal to $4\gamma^*$ (i.e., $\pi/2$). Note that *South*(r) is delimited by $\Lambda_E(r)$ and $\Lambda_S(r)$. However, only $\Lambda_S(r)$ is included in *South*(r).

- *West*(r) sector: it is adjacent to *South*(r) in clockwise direction and its angular sector α_W is equal to $2\gamma^*$, that is $\pi/4$. Note that *West*(r) is delimited by $\Lambda_W(r)$ and $\Lambda_N(r)$. However, only $\Lambda_W(r)$ is a part of *West*(r) sector.

- *North*(r) sector: this is the remaining sector, and its angular sector α_N is equal to $6\gamma^*$, that is $3\pi/4$. Note that *North*(r) is delimited by $\Lambda_N(r)$ and $\Lambda_W(r)$. However, only $\Lambda_N(r)$ is included in *North*(r) sector.

In the following, we will describe the possible configurations of the two robots, given the above partitions.

Table 7.1: Different configurations and movements of robot r and r' ($\gamma^* = \pi/8$).

	Robot r			
Robot r'	*North* (no movement)	*South* (direct move)	*East* (side move up)	*West* (side move down)
North (no movement)	no	○	○	○
South (direct move)	○	no	○	no
East (side move up)	○	○	no	○
West (side move down)	○	no	○	no

7.2.2 Valid Configurations

We consider two robots r and r' that are equipped with compasses that can diverge by as much as $2\gamma^*$, that is $\pi/4$. Let r and r' divide the plane as described in Section 7.2.1. Then, r and r' can only be in one of the following valid configurations, or a symmetric configuration:

1. Configuration *North*/*South*: $r' \in South(r)$ (i.e., robot r sees r' in its *South* sector) and $r \in North(r')$, or vice versa.
2. Configuration *North*/*East*: $r' \in East(r)$ and $r \in North(r')$, or vice versa.
3. Configuration *North*/*West*: $r' \in West(r)$ and $r \in North(r')$, or vice versa.
4. Configuration *East*/*West*: $r' \in West(r)$ and $r \in East(r')$, or vice versa.
5. Configuration *East*/*South*: $r' \in South(r)$ and $r \in East(r')$, or vice versa.

Based on the partitions described in Section 7.2.1, Table 7.1 summarizes possible and impossible configurations when robots's compasses are inaccurate by at most $\gamma^* = \pi/8$, with respect to some global north. By design, the partitions prevent the occurrence of some undesirable configurations, such as *North*/*North*, that could lead to a deadlock situation by using the algorithm [2] (see Section 7.2.3).

7.2.3 Movements

The algorithm is given in Algorithm 3, and Table 7.1 summarizes the different movements of robot r and r' (the table is symmetrical). Let us consider the movement of robot r. First, robot r creates the four sectors, and then it decides its movement based on the sector in which it has located robot r', as follows:

[2] It is important to mention that when γ^* is equal to zero, i.e., when the compasses of r and r' are consistent, the configurations *East*/*South* and *North*/*West* are impossible.

Algorithm 3 Gathering Two Robots with $\pi/8$-Inaccurate Compasses

1: Robot r divides the plane into four sectors: *North*, *South*, *East* and *West* (see Section 7.2.1);
2: $r' :=$ the other robot visible to r at some time t;
3: **if** (r sees only itself) **then** {*gathering terminated*}
4: Do_nothing();
5: **else**
6: **if** ($|South(r)| > 0$) **then** {r' is to the *South*: *direct move*}
7: Move(r');
8: **else if** ($|East(r)| > 0$) **then** {r' is to the *East*: *side move up*}
9: $\Psi_E(r) :=$ the parallel axis to $\Lambda_E(r)$ passing through r';
10: $H := \Lambda_N(r) \cap \Psi_E(r)$ (see Figure 7.3);
11: $Goal := p \in \Lambda_N(r)$ such that $dist(r, Goal) > dist(r, H)$ and $\widehat{rGoalr'} \geq \widehat{rr'Goal}$;
12: Move($Goal$);
13: **else if** ($|West(r)| > 0$) **then** {r' is to the *West*: *side move down*}
14: $\Psi_W(r) :=$ the parallel axis to $\Lambda_W(r)$ passing through r';
15: $H' := \Lambda_S(r) \cap \Psi_W(r)$ (see Figure 7.4);
16: $Goal := p \in \Lambda_S(r)$ such that $dist(r, Goal) > dist(r, H')$ and $\widehat{rGoalr'} \geq \widehat{rr'Goal}$;
17: Move($Goal$);
18: **else** {r' is to the *North*: *no movement.*}
19: Do_nothing();
20: **end if**
21: **end if**

- *No movement* (Algorithm3:line 18): If $r' \in North(r)$, then r does not move. That is, if r sees r' in its *North* sector, it remains stationary.

- *Direct move* (Algorithm3:line 6): If $r' \in South(r)$, then r moves directly in a linear movement to r'.

- *Side move up* (Algorithm3:line 8): If $r' \in East(r)$, then r performs a *side move up*. The need for such a move is explained as follows: given the valid configurations described in Section 7.2.2, if $r' \in East(r)$, then $r \in South(r')$ or $r \in North(r')$ or $r \in West(r')$. Robot r (also r') cannot figure out in which configuration they are, for instance the *East*/*South* or *North*/*East* configuration. Then, if we let robot r make a *direct move* toward r', then in the case when both robots are in the configuration *East*/*South*, they will swap their positions endlessly. Also, if we make robot r stay still, then, if both robots are in the configuration *North*/*East*, none of the robots will ever move and they will always remain in a deadlock situation. Therefore, the aim of this *side move up* is to bring both robots eventually into the configuration *North*/*South*, where one robot can move and the other remains stationary, which can lead to gathering by our algorithm.

  A *side move up* is computed by robot r as follows: let H be the intersection of $\Lambda_N(r)$

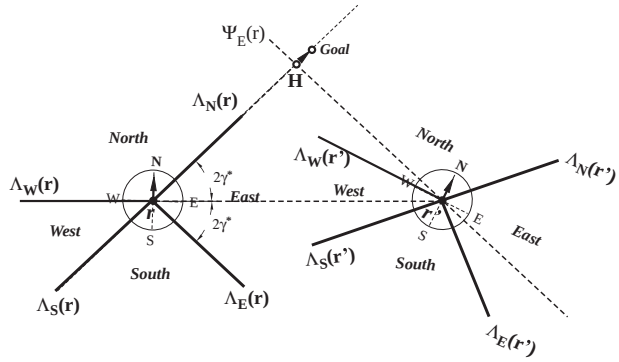


Figure 7.3: Side move up on $\Lambda_N(r)$: $r' \in East(r)$, then r performs a *side move up* to *Goal*.

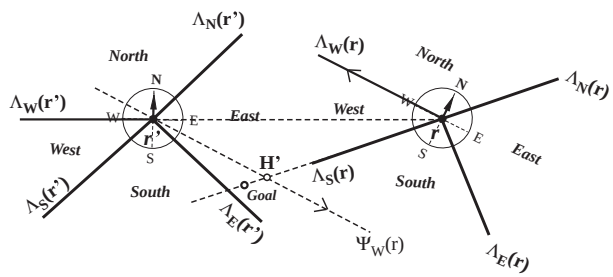


Figure 7.4: Side move down on $\Lambda_S(r)$: $r' \in West(r)$, then r performs a *side move down* to *Goal*.

and the axis $\Psi_E(r)$, with $\Psi_E(r)$ parallel to $\Lambda_E(r)$ passing through robot r'. Then, the destination *Goal* of robot r is any point that belongs to $\Lambda_N(r)$, such that the distance $dist(r, Goal) > dist(r, H)$, and the angle $\widehat{rGoalr'}$ is greater than or equal to the angle $\widehat{rr'Goal}$ (refer to Figure 7.3).

- *Side move down* (Algorithm3:line 13): If $r' \in West(r)$, then r performs a *side move down*. The aim of this move is similar to the *side move up*, and it is computed by robot r as follows: let H' be the intersection of $\Lambda_S(r)$ and the axis $\Psi_W(r)$, with $\Psi_W(r)$ parallel to $\Lambda_W(r)$ passing through robot r' (refer to Figure 7.4). Then, the destination *Goal* of robot r is any point that belongs to $\Lambda_S(r)$, such that the distance $dist(r, Goal) > dist(r, H')$, and the angle $\widehat{rGoalr'}$ is greater than or equal to the angle $\widehat{rr'Goal}$ (refer to Figure 7.4).

7.3 Correctness

In this section, we will prove that our algorithm solves the problem of gathering two robots in a finite time, assuming $\pi/8$-Inaccurate compasses.

We first state some lemmas, to illustrate that some incompatible configurations are ruled out by the algorithm. Second, we show how any possible configuration by the algo-

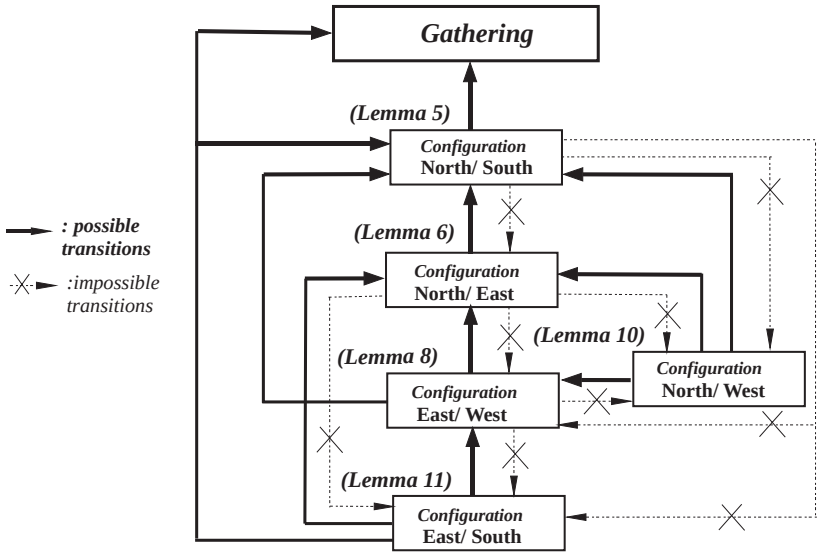


Figure 7.5: Different configurations allowed by Algorithm 3, and their transformation to gathering.

rithm is transformed into gathering in a finite time. Figure 7.5 summarizes the different possible configurations, and their transformation to gathering.

Under the partitions described in Section 7.2.1 and by considering $\gamma^* = \pi/8$, trivially, we derive the following two lemmas:

Lemma 7.3.1 *Under the partitions, and assuming $\pi/8$-Inaccurate compasses, the system can not be in the configuration North/North or East/East or South/South or West/West at any time t.*

Lemma 7.3.2 *Under the partitions, and assuming $\pi/8$-Inaccurate compasses, the system can not be in the configuration West/South at any time t.*

From the above two lemmas, we derive the following theorem:

Theorem 7.3.3 *By the algorithm, the possible configurations are North/South, North/East, North/West, East/West and East/South, and their symmetric ones (i,e. South/North, East/North, West/North, West/East and South/East).*

Lemma 7.3.4 *Given a robot r and its target point H with $r \neq H$, r reaches its target in a finite number of steps.*

PROOF. The proof derives from Assumption 2.3.1. In one cycle, r travels at least $\delta_r > 0$ of the desired distance. Besides, by Assumption 2.3.2, the cycle of a robot is finite. Thus, the number of steps required for robot r to reach its destination H is at most $\lceil dist(r, H)/\delta_r \rceil$, which is finite, and the lemma holds. $\square_{\text{Lemma 7.3.4}}$

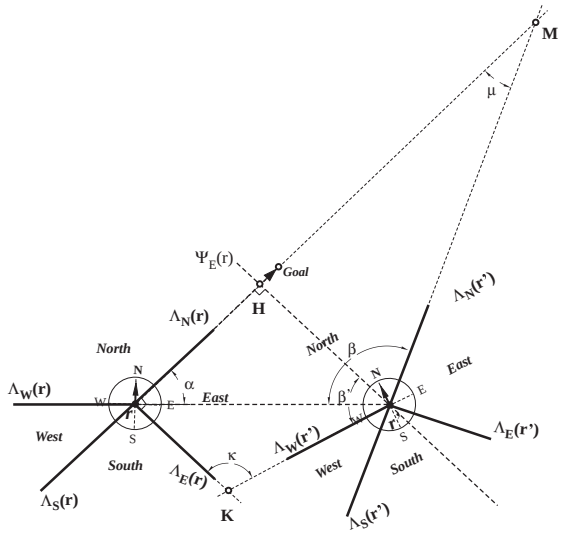


Figure 7.6: Transformation of *North/East* configuration.

Lemma 7.3.5 *Given two robots r and r' that are in the configuration North/East or East/West or East/South at some time t_0, with $r' \in East(r)$ and r either in $North(r')$ or $West(r')$ or $South(r')$, then the destination Goal computed by robot r (resulting from its side move up) is in the $North(r')$.*

Proof.

We will prove the *North/East* configuration only. The *East/West* and *East/South* configurations can be proved in a similar way.

Assume that $r' \in East(r)$ and $r \in North(r')$ at time t_0. First, observe that if $\Lambda_N(r) \cap \Lambda_N(r') = \emptyset$ (i.e., $\Lambda_N(r)$ and $\Lambda_N(r')$ are parallel or do not intersect), then $Goal \in North(r')$ because $r \in North(r')$, and $Goal \in \Lambda_N(r)$.

Now assume that, $\Lambda_N(r) \cap \Lambda_N(r') = M$. Let $H = \Psi_E(r) \cap \Lambda_N(r)$ (refer to Figure 7.6). To show that $Goal \in North(r')$, we will show that, always, $Goal \in \triangle(r, r', M)$. In other words, we need to show that $H \in \triangle(r, r', M)$ and the distance $dist(H, M) \neq 0$.

Consider the triangle $\triangle(r, r', M)$. Let α, β, and μ denote the angles at r, r' and M that are within the triangle $\triangle(r, r', M)$, respectively. First, if all three angles α, β, and μ are acute, then obviously the foot H of the perpendicular starting from r' is inside $\triangle(r, r', M)$, and $dist(H, M) \neq 0$. Second, if the angle β at r' is obtuse, then again the foot H of the perpendicular starting from r' is inside $\triangle(r, r', M)$, and $dist(H, M) \neq 0$. Now consider the angle α at r. By hypothesis, α_E is equal to $\pi/2$. This means that α cannot be an obtuse angle, and it is at most $\pi/2$. In this later case where $\alpha = \pi/2$, we have the foot H of the perpendicular starting from r' equal to r (in this case $\Lambda_E(r)$ passes by r'), and the triangle $\triangle(r', r, M)$ has a right angle at r. Consequently, $dist(r, M) = dist(H, M) \neq 0$ and $Goal \in \triangle(r, r', M)$.

Now, we will prove that the angle μ at M cannot be an obtuse angle (because if μ is an obtuse angle, H is outside $\triangle(r, r', M)$). Let $K = \Lambda_E(r) \cap \Lambda_W(r')$ and κ be the angle at K.

We also denote by β' the angle at r' formed by $\Psi_E(r)$ and $\Lambda_W(r')$. Consider the quadrilateral formed by r, H, r' and K. Then, we have: (1) $\kappa + \beta' = \pi$ since the respective angles at r and H are equal to $\pi/2$. Consider now the quadrilateral formed by r, K, r' and M. Then, we have: (2) $\kappa + \mu = 3\pi/4$ since $\alpha_E(r)$ is equal to $\pi/2$, and $\alpha_N(r')$ is equal to $3\pi/4$ by hypothesis. By subtraction of (1) from (2), we get: (3) $\beta' - \mu = \pi/4$. By assumption, $\beta' < 3\pi/4$ because $\Psi_E(r)$ can not be equal to $\Lambda_N(r')$ as $\Lambda_N(r')$ can not be perpendicular to $\Lambda_N(r)$ by the partitions described in Section 7.2.1. Consequently, the angle μ at M is less than $\pi/2$. Thus, μ can not be an obtuse angle. As a result, in all cases the foot H of the perpendicular starting from r' is inside the triangle $\triangle(r, r', M)$, and $dist(H, M) \neq 0$. Then, $\forall p \in \overline{HM}$, $p \in North(r')$. We have by the algorithm, $\widehat{rGoalr'} \geq \widehat{rr'Goal}$. Since μ is not an obtuse angle and $\widehat{rr'M}$ can be an obtuse angle, then the triangle $\triangle(r, r', Goal)$ is included in $\triangle(r, r', M)$. This proves that $Goal \in \triangle(r, r', M)$, and thus $Goal \in North(r')$. This completes the proof. $\square_{\text{Lemma 7.3.5}}$

In the following, we will show the different possible transitions that each valid configuration can take, in order to reach gathering in a finite time. The impossible transitions can be derived implicitly, so we do not prove them explicitly.

7.3.1 Transition of *North/South* Configuration to Gathering

Lemma 7.3.6 *Let r and r' be two robots that are in the configuration North/South with $r' \in South(r)$ at some time t_0. Then, there is a time $\bar{t} > t_0$ when r and r' gather at the same point. Moreover, r and r' can not shift to any other configuration except gathering.*

Proof. By the algorithm, r will perform a *direct move* toward r'. Also, during the movement of r, r' is unable to move. Consequently, by Lemma 7.3.4, r reaches r' in a finite time. This terminates the proof. $\square_{\text{Lemma 7.3.6}}$

7.3.2 Transition of *North/East* Configuration to Gathering

Lemma 7.3.7 *Let r and r' be two robots that are in the configuration North/East with $r' \in East(r)$, and $r \in North(r')$ at some time t_0. Then, there is a finite time $\bar{t}$ at which this configuration is transformed into North/South configuration with $r' \in South(r)$. Moreover, r and r' can not shift to any other configuration except the North/South configuration.*

Proof. The proof is a direct consequence from Lemma 7.3.5. Let $Goal$ be the destination of r. Initially, $r \in North(r')$. Besides, by Lemma 7.3.5, $\forall p \in \overline{rGoal}$, $p \in North(r')$. Then, r' is unable to move during the movement of r to $Goal$. When r reaches its destination $Goal$, $\Lambda_E(r)$ is above r', thus $r' \in South(r)$. Consequently, r and r' enter the

configuration *North/South* in a finite time. $\square_{\text{Lemma 7.3.7}}$

From Lemma 7.3.6 and Lemma 7.3.7, we conclude that:

Theorem 7.3.8 *Any North/East configuration of two robots equipped with $\pi/8$-Inaccurate compasses is transformed after a finite time to gathering.*

7.3.3 Transition of *East/West* Configuration to Gathering

Lemma 7.3.9 *Given two robots r and r' at some time t_0, where r and r' are in the configuration East/West, with $r \in West(r')$ and $r' \in East(r)$, then the destination $Goal'$ of r' (resulting from its side move down) belongs to $East(r)$ or $South(r)$.*

Proof. Let $H' = \Psi_W(r') \cap \Lambda_S(r')$. Consider the triangle $\triangle(r, r', Goal')$, and let α, α' and β be the angles at r, r' and $Goal'$, respectively. By hypothesis, $\alpha' \leq \alpha_W = \pi/4$. Then, $\alpha+\beta \leq 3\pi/4$. By the algorithm, $\alpha \leq \beta$. Thus, $\alpha \leq 3\pi/8 < \pi/2$. Let $M = \Lambda_E(r) \cap \Lambda_S(r')$. Then, the angle $\widehat{r'rM} \leq \pi/4$ since r and r' are in the configuration *East/West*. It follows that if $Goal' \in \overline{H'M}$, then $Goal' \in East(r)$. Otherwise, $Goal' \in South(r)$. $\square_{\text{Lemma 7.3.9}}$

Lemma 7.3.10 *Let r and r' be two robots that are in the configuration East/West, with $r' \in East(r)$, and $r \in West(r')$ at some time t_0. Then, there is a finite time $\bar{t}$ in which this configuration is transformed into North/East or North/South configuration. Moreover, r and r' cannot enter any other configuration except the North/East or North/South configuration.*

Proof. We distinguish several cases depending on the movement of each robot. We assume that both r and r' always reach their final destinations. All other cases where r or r' end their moves before destination are easy to deduce from previous lemmas.

1. **r moves/ r' does not move**: By the algorithm, r will perform a *side move up*. Let *Goal* be the destination of r and $\bar{t}$ be the time when r reaches its target. At $\bar{t}$, we have $r' \in South(r)$ (since at $\bar{t}$, r' is below $\Lambda_E(r)$). In addition, by Lemma 7.3.5, $Goal \in North(r')$. Then, at $\bar{t}$, $r \in North(r')$. Consequently, r and r' enter the configuration *North/South* in a finite time.

2. **r' moves/ r does not move**: By the algorithm, r' will perform a *side move down*. Let $Goal'$ be its destination and $\bar{t}'$ be the time when r' reaches $Goal'$.

   At time $\bar{t}'$, r is above $\Lambda_W(r')$, thus $r \in North(r')$. In addition, by Lemma 7.3.9, $r' \in East(r)$ or $r' \in South(r)$ at $\bar{t}'$. Consequently, r and r' leave the configuration *East/West* in a finite number of steps, and enter the configuration *East/North* or *North/South*.

3. **both r and r' move**: By the algorithm, r will perform a *side move up* and r' will perform a *side move down*. Let $Goal$ and $Goal'$ be their respective destinations and $\bar{t}$ and $\bar{t}'$ be the times when they end their moves, respectively. At $\bar{t}$, $\forall p$ that is below $\Lambda_E(r(\bar{t}))$, $p \in South(r)$. Since, at $\bar{t}$, $r' \in \overline{r'Goal'}$, and by Lemma 7.3.9, $Goal' \in East(r(t_0))$ or $Goal' \in South(r(t_0))$, thus, $r' \in South(r)$ at $\bar{t}$ because $\Lambda_E(r(\bar{t}))$ is above $Goal'$ and r'.

   When r' reaches $Goal'$, r is above $\Lambda_W(r')$. Consequently, at $\bar{t}'$, $r \in North(r')$. Since, r and r' reach their respective targets in a finite time, we hence conclude that they enter the configuration $North/South$ in a finite time.

$\square_{\text{Lemma 7.3.10}}$

From Lemma 7.3.10, Lemma 7.3.6 and Theorem 7.3.8, we conclude:

Theorem 7.3.11 *Any East/West configuration of two robots equipped with $\pi/8$-Inaccurate compasses is transformed after a finite time to gathering.*

7.3.4 Transition of *North/West* Configuration to Gathering

Lemma 7.3.12 *Given two robots r and r' at some time t_0, where r and r' are in the configuration North/West, with $r \in West(r')$ and $r' \in North(r)$, then the destination $Goal'$ of r' (resulting from its side move down) belongs to $East(r)$.*

The proof is very similar to the proof of Lemma 7.3.9, and thus omitted here.

Lemma 7.3.13 *Let r and r' be two robots that are in the configuration North/West, with $r \in West(r')$, and $r' \in North(r)$ at some time t_0. Then, there is a finite time $\bar{t}$ in which this configuration is transformed into North/East or East/West or North/South configuration. Moreover, r and r' can not enter any other configuration except the North/East or East/West or North/South configuration.*

Proof.

By the algorithm, r' will make a *side move down*. Let $Goal'$ be its destination. Then, by Lemma 7.3.12, $Goal' \in East(r)$. As long as $r' \in North(r)$, r remains stationary. While r' is moving toward its target, it crosses $East(r)$ sector. Then, r and r' enter the configuration $East/West$ if $\Lambda_W(r')$ is still above r. Otherwise, they enter the configuration $North/East$, with $r \in North(r')$ if r' reaches $Goal'$ and r still did not move. Finally, r and r' enter the configuration $North/South$ if r performs a look operation when $r' \in East(r)$, and moves to it destination. From Lemma 7.3.4, these transformations are done in a finite time, and the lemma holds. $\square_{\text{Lemma 7.3.13}}$

From Lemma 7.3.13, Theorem 7.3.8 and Theorem 7.3.11, we conclude:

Theorem 7.3.14 *Any North/West configuration of two robots equipped with $\pi/8$-Inaccurate compasses is transformed after a finite time to gathering.*

7.3.5 Transition of *East/South* Configuration to Gathering

Lemma 7.3.15 *Let r and r' be two robots that are in the configuration East/South at some time t_0, with $r' \in East(r)$ and $r \in South(r')$ Then, there is a finite time t in which this configuration is transformed into North/South or North/East or East/West or gathering.*

Proof. By the algorithm, r' will make a *direct move* toward r, and r will make a *side move up*. Then, we distinguish several cases, depending on where each robot sees the other one, and where it ends its move. By using similar arguments as in previous lemmas, it is easy to show that r and r' shift to the *North/South* or *North/East* or *East/West* configuration, or gathering in a finite time. $\square_{\text{Lemma 7.3.15}}$

From Lemma 7.3.6, Lemma 7.3.15, Theorem 7.3.8 and Theorem 7.3.11, we conclude that:

Theorem 7.3.16 *Any East/South configuration of two robots equipped with $\pi/8$-Inaccurate compasses is transformed in a finite time to gathering.*

Theorem 7.3.17 *In a system, with 2 anonymous, oblivious mobile robots relying on inaccurate compasses, the gathering problem is solvable in a finite time for $\pi/8$-Inaccurate compasses.*

Proof.

Theorem 7.3.3 states the different valid configurations under the algorithm. Also, from Lemma 7.3.6, Theorem 7.3.8, Theorem 7.3.11, Theorem 7.3.14 and Theorem 7.3.16, any valid configuration is transformed into gathering in a finite time (see Figure 7.5), thus the theorem holds. $\square_{\text{Theorem 7.3.17}}$

7.4 Gathering Robots with Volume

In this section, we show that Algorithm 3 solves also the problem if we consider robots are not represented by points in the plane, but they have a volume, and occupy some space in the plane.

Adding the realistic aspect of volume to the robots makes the problem more complex. In particular, robots need to avoid collisions, and also they should solve the problem of *partial visibility*, that is some robots should not obstruct vision of others. In the case of two robots, the problem of partial visibility does not exist, however, robots need to avoid collisions between each other.

We will show that Algorithm 3 solves the gathering of two fat robots that are represented by unit discs in the plane, and are equipped with $\pi/8$-Inaccurate compasses. The two robots achieve the gathering, if one robot touches the other robot, that is, the circles representing these robots become tangent, and both robots stop.

Theorem 7.4.1 *Algorithm 3 is a correct gathering algorithm for two robots with volume, and equipped with $\pi/8$-Inaccurate compasses.*

PROOF. The proof consists of showing that the trajectory of the two robots does not intersect during the entire execution of the algorithm. Since, there is only two robots in the system, then the two robots are always visible because there is no other robot that obstructs their sight.

We will show that the different movements allowed by the algorithm for the two robots do not bring them in a situation, where their line of moves intersect.

From Table 7.1, the different combinations of movements of robots r and r' are analyzed as follow:

- *No movement/Direct move*: in this case, obviously the line of moves of the two robots do not intersect since only one robot is allowed to move, and the other robot is unable to move during the entire execution of the algorithm.

- *No movement/Side move up* and *No movement/Side move down*: the same arguments holds for this case as one robot is able to move and the other one stays.

- *Direct move/Side move up*: Let robot r be the robot executing the direct move, and robot r' be the robot executing the side move up. Let also, H' be the destination of r'. By the algorithm, $H' \neq r$. Then, the trajectory of robot r' is the segment $\overline{r'H'}$. For robot r, its destination is any point $p \in \overline{r'H'}$, in which r' is occupying at the time when robot r performed its look operation or a point that is already passed by r'. Consequently, robot r does not interfere on the trajectory of r', and vise versa.

- *Side move up/Side move down*: Let robot r be the robot executing the side move up, and robot r' be the robot executing the side move down. Let H and H' be the destinations of r and r', respectively. By the algorithm, H is above the segment $\overline{rr'}$ because $H \in \Lambda_N(r)$. Also, H' is below the segment $\overline{rr'}$ because $H' \in \Lambda_S(r')$. Consequently, r and r' are moving on opposite directions, and hence they will never intersect their lines of move.

$\square_{\text{Theorem 7.4.1}}$

7.5 Complexity Analysis

In this section, we give an analytic analysis of the complexity of the algorithm, and the time of its termination.

Complexity of the algorithm. The complexity of Algorithm 3 is a *constant* because all computations can be done in a constant time.

Number of steps of termination of the algorithm. Recall Assumption 2.3.1 in the CORDA model, which states that the minimum distance travelled by one robot in one move is at least Δ_r. Assume that the distance between robots r_1 and r_2 is equal to D.

To compute the number of steps required by the algorithm in order to terminate, we will first compute the number of steps that each configuration takes in order to reach the gathering configuration. For the sake of analysis, we assume that robots r_1 and r_2 are always activated simultaneously.

- Let robots r_1 and r_2 be in the configuration *North/South*. Then, one of these robots will make a direct move to the other one, while the other robot remains stationary. Thus, the maximum number of steps required to reach the gathering configuration is: $S1 = D/\Delta_r$.

- Let robots r_1 and r_2 be in the configuration *North/East* with r_2 to the east of r_1, then this configuration is transformed by Lemma 7.3.7 to the configuration *North/South*. This transformation takes also D/Δ_r because r_1 travels on Λ_N au maximum the distance $dist(r_1, r_2) = D$ by the algorithm. Thus, the maximum number of steps required to reach the gathering configuration is: $S2 = S1 + D/\Delta_r = 2D/\Delta_r$.

- Let robots r_1 and r_2 be in the configuration *East/West*. By Lemma 7.3.10, this configuration is transformed to the *North/East* configuration. Then, the maximum number of steps required to reach the gathering configuration is: $S3 = S2 + D/\Delta_r = 3D/\Delta_r$.

- Let robots r_1 and r_2 be in the configuration *North/West*. By Lemma 7.3.13, this configuration is transformed to the *East/West* configuration. Then, the maximum number of steps required to reach the gathering configuration is: $S4 = S3 + D/\Delta_r = 4D/\Delta_r$.

- Let robots r_1 and r_2 be in the configuration *East/South*. By Lemma 7.3.15, this configuration is transformed to the *East/West* configuration. Then, the maximum number of steps required to reach the gathering configuration is: $S5 = S3 + D/\Delta_r = 4D/\Delta_r$.

In conclusion, the maximum number of steps of termination of Algorithm 3 is: $4D/\Delta_r$.

7.6 Summary

In this chapter, we have proposed a self-stabilizing algorithm with which two asynchronous robots can gather in finite time using inaccurate compasses with a divergence of as much as 45°, and relying on oblivious computations. Our algorithm is self-stabilizing, and tolerates any number of transient errors. We can also argue that even with weaker compasses that

fluctuate for some arbitrary periods, if eventually they stabilize to some bounded errors that are less than or equal to 45° (eventually bounded error compass), our algorithm is still valid and solves the problem in a finite time. Finally, we have proved that our algorithm still works if we consider robots with dimension.

In the next chapter, we further extend this work by proving a tight bound on the degree of divergence of robots' compasses for solving the gathering problem for two robots.

Chapter 8

Tight Bound on the Gathering of Two Robots with Inaccurate Compasses

To be is to be the value
of a bound variable.
Willard Van Quine

In the previous chapter, we have proposed an algorithm that gathers two oblivious mobile robots in finite time provided that the divergence between their compasses is at most 45°. The question remained open, however, as to whether the problem could still be solved with a larger divergence.

In this chapter, we propose a distributed algorithm that solves the gathering problem with two asynchronous robots, when their compasses can differ by any angle less than 180°, which is obviously the largest divergence for which the compasses can still provide any useful information.

8.1 Gathering with Inaccurate Compasses for $\theta < \pi$

In this section, we provide an algorithm for solving the gathering of two asynchronous oblivious mobile robots when their compasses diverge by an angle $\theta < \pi$. [1]

8.1.1 Algorithm Overview

The algorithm is described informally as follows. Consider a local x-y coordinate system where the positive y-axis points North, and hence the positive x-axis points East. Let also the location of the robot be the origin of its local coordinate system.

Let A be some robot, and let B be the position at which the other robot is located. We denote by α the angle between the y-axis of robot A, namely y_A and the segment $\overline{AB}$.

[1] θ is equal to $2\gamma^*$-Inaccurate compasses.

That is, $\alpha = 0$ when B is on the positive y_A axis and $\alpha = \pi/2$ when B is on the positive x-axis of robot A. Finally, let θ be the difference in north direction indicated by the two local coordinate systems of robots A and B. In our algorithm, we assume that $0 \leq \theta < \pi$. Then, robot A decides its movement as follows:

- If the angle α between y_A and $\overline{AB}$ in clockwise direction is strictly greater than 0 and smaller than or equal to π, then robot A moves directly on the segment $\overline{AB}$ to B. We refer to this move as ***direct move***.
- If the angle α is strictly greater than π and smaller than $\pi + \theta$, then robot A moves towards its south by the distance $dist(A, B)$. We will refer to this move as ***side move south***.
- If the angle α is strictly greater than $\pi + \theta$ and smaller than or equal to 2π, then robot A does not move. We refer to this move as ***no move***.

The algorithm is given in Algorithm 4, and Table 8.1 summarizes the different movements of robots A and B (the table is symmetrical).

Algorithm 4 Gathering two asynchronous robots, when compass divergence $\theta < \pi$.

```
if (r sees only itself) then {gathering terminated}
    Do_nothing();
else
    B := position of the other robot B;
    y_A := y-axis of robot A;
    α := angle between y_A and AB in clockwise direction;
    if (0 < α ≤ π) then {direct move}
        robot A moves to robot B;
    else if (π < α < π + θ) then {side move south}
        robot A moves toward its south by distance dist(A, B);
    else if (π + θ < α ≤ 2π) then {no move}
        Do_nothing();
    end if
end if
```

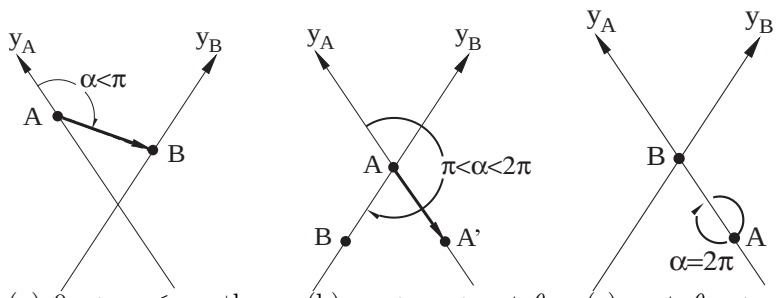
8.1.2 Description of Situations

In this section, we define the different possible situations of robots A and B, when their compasses are inconsistent by $0 \leq \theta < \pi$. Without loss of generality, we consider that the north of robot B, denoted by y_B is always on the right hand side of the north of robot A, denoted by y_A. Thus, we define the following 10 situations:[2]

[2] If the north of robot B is on the left hand side of the north of robot A, then by symmetry we will have the same 10 situations.

Table 8.1: Combination of movements of robots A and B allowed by the algorithm ($\theta < \pi$).

	Robot A		
Robot B	$0 < \alpha \leq \pi$ *(direct move)*	$\pi < \alpha < \pi + \theta$ *(side move south)*	$\pi + \theta < \alpha \leq 2\pi$ *(no move)*
$0 < \alpha \leq \pi$ *(direct move)*	○	○	○
$\pi < \alpha < \pi + \theta$ *(side move south)*	○	○	○
$\pi + \theta < \alpha \leq 2\pi$ *(no move)*	○	○	not applicable



(a) $0 < \alpha \leq \pi$, then A moves directly to B. :**Direct move**

(b) $\pi < \alpha < \pi + \theta$, then A moves by distance $dist(A, B)$ to A': **Side move south**.

(c) $\pi + \theta < \alpha \leq 2\pi$, then A does not move: **No move**.

Figure 8.1: Principle of the algorithm.

1. **Situation (1):** the y_A axis of robot A and y_B of robot B are equal ($y_A = y_B$), and A and B are located on the same y-axis (refer to Figure 8.2(a)).

2. **Situation (2):** the y_A axis of robot A and y_B of robot B are parallel. That is, A and B are not located on the same y-axis (refer to Figure 8.2(b)).

   Situations (1) and (2) refer to cases when θ is equal to zero. In the following cases, we consider that θ is other than zero. Let I be the intersection of y_A and y_B. Then, four cases arise when both A and B are not at I.

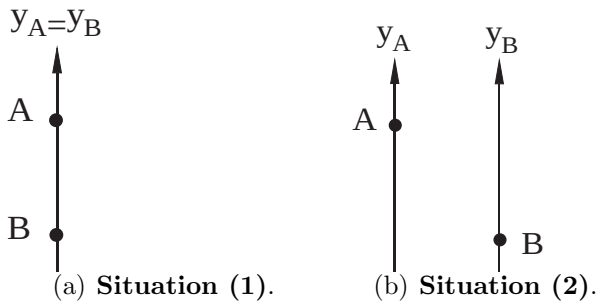
3. **Situation (3):** in this situation, A is below I, and B is above I (see Figure 8.3(a)).

4. **Situation (4):** in this situation, both A and B are above I (see Figure 8.3(b)).

5. **Situation (5):** in this situation, A is above I, and B is below I (see Figure 8.3(c)).

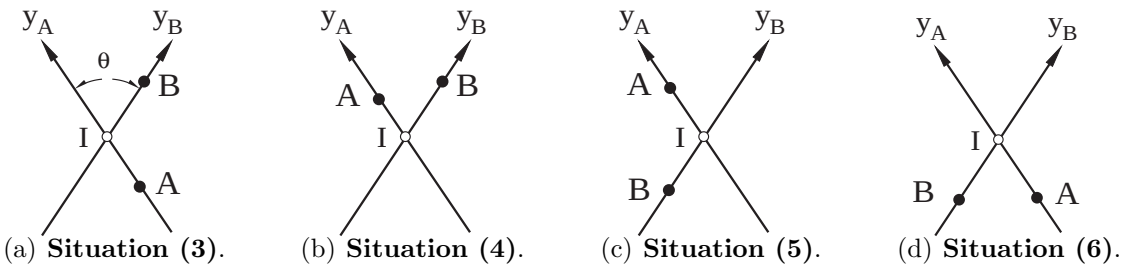
6. **Situation (6):** in this situation, both A and B are below I (see Figure 8.3(d)).

   Finally, we distinguish the following four cases (refer to Figure 8.4) when either robot A or B is at I.



(a) **Situation (1)**. (b) **Situation (2)**.

Figure 8.2: Situations of robots A and B where $\theta = 0$.



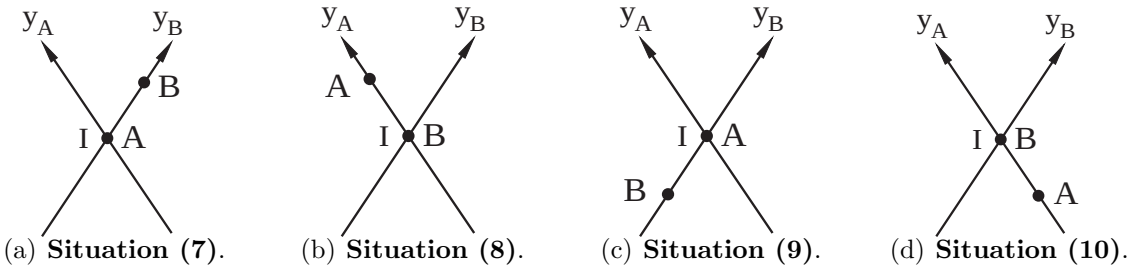
(a) **Situation (3)**. (b) **Situation (4)**. (c) **Situation (5)**. (d) **Situation (6)**.

Figure 8.3: Situations of A and B where $\theta \neq 0$ and both A and B are not at I.

7. **Situation (7):** in this situation, A is at I and B is above I.
8. **Situation (8):** in this situation, A is above I and B is at I.
9. **Situation (9):** in this situation, A is at I, and B is below I.
10. **Situation (10):** in this situation, A is below I, and B is at I.

8.2 Correctness

In this section, we will prove that our algorithm solves the problem of gathering two robots in a finite time, when their compasses diverge by an angle that is strictly smaller than π. To do so, we show how any possible situation is transformed into gathering in a finite



(a) **Situation (7)**. (b) **Situation (8)**. (c) **Situation (9)**. (d) **Situation (10)**.

Figure 8.4: Situations of A and B where $\theta \neq 0$ and either A or B is at I.

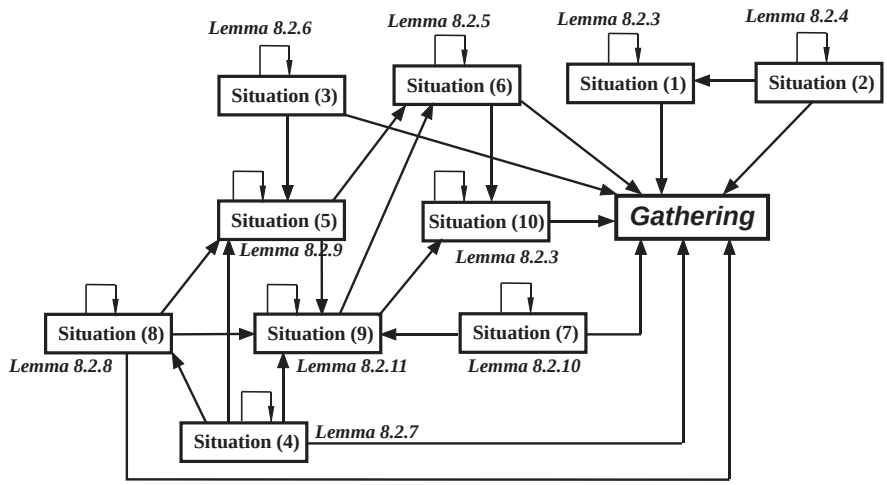


Figure 8.5: Diagram of possible transitions between situations.

time. Figure 8.5 shows a diagram of all possible transitions between situations. Assume without loss of generality that y_B is to the right of y_A, and $0 \leq \theta < \pi$, then trivially, we derive the following lemmas:

Lemma 8.2.1 *The situations* $1 - 10$ *(Sec. 8.1.2) form a list of all possible positions of robots* A *and* B*.*

Lemma 8.2.2 *Under Algorithm 4, there exists no situation where both robots* A *and* B *perform* ***no move****.*

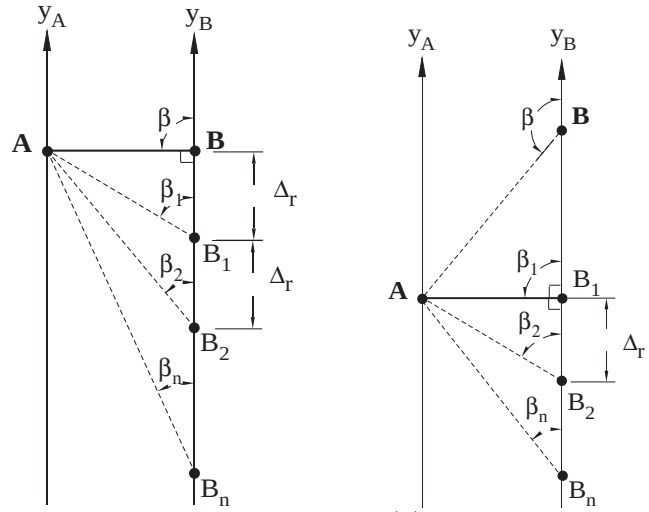
In the remainder of the text, we denote by α_A the angle from y_A to robot B in clockwise direction with respect to the local coordinate system of A, and by α_B the angle from y_B to robot A in clockwise direction with respect to the local coordinate system of B. We also denote by I, the intersection of y_A and y_B.

Lemma 8.2.3 *In a finite number of cycles,* ***Situation*** (1) *and* ***Situation*** (10) *are transformed into gathering.*

Proof. Let two robots A and B be in *Situation* (1). Without loss of generality, let A be above B. According to the algorithm, as long as A is above B, B performs *no move* because $\alpha_B = 2\pi$. Consider now the movement of robot A. We have $\alpha_A = \pi$. Then, A performs a *direct move* to B. By Assumption 2.3.1, in one cycle, A travels at least Δ_r. Consequently, A reaches B in a finite number of steps.

The proof of transformation of *Situation* (10) to gathering is similar to the proof of *Situation* (1), and thus omitted here. $\square_{\text{Lemma 8.2.3}}$

Lemma 8.2.4 *In a finite number of cycles,* ***Situation*** (2) *is transformed into* ***Situation*** (1) *or to gathering.*



(a) Initially, β is equal to $\pi/2$. (b) Initially, β is an obtuse angle.

Figure 8.6: Robot B stops (forever) at B_n in finite number of steps.

PROOF. By the algorithm, B moves on y_B by the distance $dist(A, B)$ toward its south, and A performs a direct move to B. First, it is easy to see that if A moves to the position of B, and B has already left its position (by moving on y_B toward its south), then, since the cycle of A and B is finite, they will reach **Situation** (1) in finite time.

Assume now the worst case, where B is activated infinitely many times, however A is not. Since, by the algorithm B moves on y_B toward its south by $dist(A, B)$, then, we need to show that there will be a time after which B stops (forever) moving toward its south, and this happens in finite time.

Let β be the angle between the segment $\overline{AB}$ and y_B in clockwise direction. The proof consists of showing that: (1) β is monotonically decreasing when B moves, and (2) β becomes less than $\pi - \theta$ in a finite number of steps.

Consider first the situation in Figure 8.6(a), where $\overline{AB}$ is perpendicular to y_B.

Assume that B is activated at time t, while A is not. By Assumption 2.3.1, in the worst case, B moves toward its destination on y_B by $\Delta_r > 0$. Let B_1 be the new destination of B. Consider the triangle $\triangle(A, B, B_1)$, then it is easy to see that the angle at B_1 is less than the angle at B. Let B_2 be a new destination of B, which is at distance Δ_r from B_1. Then, it is also easy to observe that the angle at B_2 is less than the angle at B_1. We thus, conclude that β is monotonically decreasing when B moves on y_B toward the south.

Now we will show that β becomes less than $\pi - \theta$ in a finite number of steps. Let n be the maximal number of steps that robot B takes in order for β to become less than $\pi - \theta$.

We assume the worst case where in every step (cycle), B moves on y_B by Δ_r. Let B_n be the position at which robot B stops after n cycles. This means that, at B_n, $\beta < \pi - \theta$. Then, we get:

$$\begin{aligned}\tan(\pi-\theta) &= dist(A,B)/dist(B,B_n)\\ &= dist(A,B)/n.\Delta_r\end{aligned}$$

Thus:

$$n = dist(A,B).\cot(\pi-\theta)/\Delta_r \tag{8.1}$$

We have, the distance $\|\overline{AB}\| > 0$ by hypothesis, and it is a constant. Also, Δ_r is a constant. By definition, $\pi-\theta$ is a strictly positive value. Consequently, $\cot(\pi-\theta) > 0$, and n is finite.

Now consider the situation in Figure 8.6(b), where the angle formed by $\overline{AB}$ and y_B is an obtuse angle ($\pi/2 < \beta < \pi$). Let B_1 be the perpendicular to y_B starting at A.

In this case, from above, we can conclude that from B_1 to B_n, β is monotonically decreasing, and β takes n finite steps to become smaller than $\pi-\theta$. Besides, by considering the triangle $\triangle(A,B,B_1)$, it is easy to show that β is monotonically decreasing while B is moving toward B_1. In addition, robot B takes a finite number of steps to reach B_1 because $dist(B,B_1)$ is less than $dist(A,B)$, which is finite, and B travels at least Δ_r in one cycle. Consequently, from B to B_n, β is monotonically decreasing and, β becomes less than $\pi-\theta$ in a finite number of steps.

Now since B has stopped moving in a finite number of steps at B_n, eventually A will do a direct move to B. Since the distance $dist(A,B)$ is finite, and by Assumption 2.3.1, A travels at least $\Delta_r > 0$ in one cycle, thus, A reaches B in a finite number of cycles. $\square_{\text{Lemma 8.2.4}}$

Lemma 8.2.5 *In a finite number of cycles,* ***Situation*** (6) *is transformed into* ***Situation*** (10) *or to gathering.*

The proof of this lemma is similar to the proof of Lemma 8.2.4, and thus omitted.

Lemma 8.2.6 *In a finite number of cycles,* ***Situation*** (3) *is transformed into* ***Situation*** (5) *or to gathering.*

Proof.

Let A and B be in Situation (3). By the algorithm, A performs a direct move to B, and B performs a direct move to A. Assume first that A performs a look operation at time t, while B does not. Subsequently, if B does not perform any look operation while A is moving toward it, then, A will gather with B at position B in finite time. Similarly, both robots gather at A in finite time if B is activated while A is not.

Now we will show how Situation (3) can be transformed to Situation (5). The proof consists of showing that there will be a finite time, where A is to the left of y_B, and B is to the left of y_A.

Let y'_A be the parallel to y_A passing through B. Let also y'_B be the parallel to y_B passing through A.

Assume that A performs a look operation at time t, and B also performs a look operation at time $t' \geq t$. Then, by the algorithm, A moves to $B(t)$, and B moves to $A(t')$. By Assumption 2.3.2, the cycle of a robot is finite, and the distance $dist(A, B)$ is finite. Then, in a finite number of steps, A and B exchange positions on the segment $\overline{AB}$. Let t'' be the time when this happens. Then, at time t'', B is to the left of y'_A, and also A is to the left of y'_B. Let also $I' = y_A(t'') \cap y_B(t'')$, thus at time t'', A is above I', and B is below I'. This terminates the proof. $\square_{\text{Lemma 8.2.6}}$

Lemma 8.2.7 *In a finite number of cycles,* ***Situation*** (4) *is transformed into* ***Situation*** (5) *or* ***Situation*** (8) *or* ***Situation*** (9) *or to gathering.*

PROOF.

By the algorithm, A executes a direct move to B, and B performs a side move to south. We distinguish the following cases depending on the activation of A and B:

- *Transformation to Situation* (5) *or* (8): Let B perform a look operation at time t, while A remains inactive. Assume also that $dist(A, B) > dist(I, B)$, where $I = y_A \cap y_B$. Then, first, if B stops at I, then A and B enter Situation (8), where B is at I and A above I. Trivially, this transformation is done in finite time by Assumption 2.3.2. Now, if B stops after the point I, then B is below I. Subsequently, A and B enter Situation (5), where A is above I, and B is below I. This transformation is also done in finite time by the same argument.

- *Transformation to Situation* (9): Let B perform a look operation at time t. Then, B executes a side move south. Let t' be the time when robot B passes by I. Suppose that A also performs a look operation at time t' and sees B at position I. Then, A performs a direct move to I (since at t', $\alpha_A = \pi$). Let t'' be the time when A finishes its move to I. Consequently, at time t'', A and B enter Situation (9), where A is at I, and B is below I.

- *Transformation to gathering*: This case is trivial. A and B gather in finite time at B by Assumption 2.3.1 if A performs a look operation before B, and during the movement of A to B, robot B does not perform any look operation.

$\square_{\text{Lemma 8.2.7}}$

Lemma 8.2.8 *In a finite number of cycles,* ***Situation*** (8) *is transformed into* ***Situation*** (5) *or* ***Situation*** (9) *or to gathering.*

The proof of this lemma is similar to the proof of Lemma 8.2.7, and thus omitted here.

Lemma 8.2.9 *In a finite number of cycles,* ***Situation*** (5) *is transformed into* ***Situation*** (6) *or* ***Situation*** (9).

PROOF. Let A and B be in Situation (5). Then, by the algorithm, both A and B execute a side move south. The proof is straightforward. If A stops at I, then we get A and B in Situation (9) because B remains below I. This transformation is done in a finite number of steps by Assumption 2.3.1. If A stops after I, then A is below I. Since B is also below I, then A and B reach Situation (6) in a finite number of steps by similar arguments. $\square_{\text{Lemma 8.2.9}}$

Lemma 8.2.10 *In a finite number of cycles,* ***Situation*** (7) *is transformed into* ***Situation*** (9) *or to gathering.*

PROOF.

Let A and B be in Situation (7), where B is to the right of y_A, and A is on y_B. By the algorithm, A performs a direct move to B, and B performs a direct move to A. Trivially, if one robot, say A, is activated and moves to the position of B, while B does not perform any look operation during the movement of A toward it, then both A and B gather at B in finite time by Assumption 2.3.2.

Now consider that both A and B are activated simultaneously. We will show that they will reach Situation (9) in finite time. The proof consists of showing that there will be a finite time, where B arrives at the left of y_A.

Assume that A performs a look operation at time t, and B also performs a look operation at time $t' \geq t$. Then, by the algorithm, A moves to $B(t)$, and B moves to $A(t')$. By Assumption 2.3.2, the cycle of a robot is finite, and the distance $dist(A, B)$ is finite. Then, in a finite number of steps, A and B exchange positions on the segment $\overline{AB}$. Let t'' be the time when this happens, and let $I' = y_A(t'') \cap y_B(t'')$. Then, at time t'', B is to the left of $y_A(t'')$. Since $A \in y_B$, then $A(t'') = I(t'')$. Thus, at time t'', B is below I', and A is at I', which represents Situation (9). This terminates the proof. $\square_{\text{Lemma 8.2.10}}$

Lemma 8.2.11 *In a finite number of cycles,* ***Situation*** (9) *is transformed into* ***Situation*** (6) *or* ***Situation*** (10).

PROOF. Let A perform a look operation at time t. Then, A performs a side move south. Let t' be the time when A finishes its move. At time t', A is below I because A must

move at least by Δ_r toward its target, according to Assumption 2.3.1. Then, if B does not perform any look operation between t and t', A and B enter Situation (6) (both below I).

Now, assume that B performs a look operation at time $t'' > t$, and that at t'', A already has left I. Then, by the algorithm B executes a direct move to $A(t'')$. Let t_f be the time when B finishes its move. Consequently, at time t_f, B is at I, and A is below I, which represents Situation (10). Since this transition is done in a finite number of steps, the lemma holds. $\square_{\text{Lemma 8.2.11}}$

Theorem 8.2.12 *Algorithm 4 correctly solves the gathering of two asynchronous mobile robots in finite time as long as their compasses diverge by $\theta < \pi$.*

PROOF. Lemma 8.2.1 states the different situations of robot A and B when $0 \leq \theta < \pi$. From Lemma 8.2.3 to Lemma 8.2.11, we show that every situation is transformed to gathering in finite time. Also, the diagram of all possible transitions between situations depicted in Figure 8.5 shows no cycles. Thus, the theorem holds. $\square_{\text{Theorem 8.2.12}}$

Now, we can directly derive the following corollary from Theorem 8.2.12, and the fact that the problem is impossible when θ is equal to π because it is as if robots do not have compasses.

Corollary 8.2.13 *$\theta < \pi$ is a tight bound on solving the gathering of two oblivious mobile robots with inaccurate compasses.*

8.3 Complexity Analysis

In this section, we give an analytic analysis of the complexity of the algorithm, and the number of steps of its termination.

Complexity of the algorithm. The complexity of Algorithm 4 is a constant because all computations can be done in a *constant* time.

Number of steps of termination of the algorithm. For the sake of analysis, we consider a strict model, where robots A and B are fully synchronized. That is, at each time instant, both A and B are activated.

In order to analyze the number of steps of convergence of the algorithm toward the gathering, we will consider first robots in situation 2, where they share the same north ($y - axis$), and they are not aligned. Then, we will consider robots in situation 3 where they do not share the same north.

Recall Assumption 2.3.1 in the CORDA model, which states that the minimum distance travelled by one robot in one move is at least Δ_r.

Let robots A and B be in one of the following situations:

- Robots A and B share the same north; Situation 2: Let robots A and B be in situation 2 as depicted in Figure 8.6(b). The worst case is when robot B moves to B_n, and then after robot B has already finishes moving, robot A starts to move to robot B at position B_n. Then, the maximum number of steps $S2$ required by A to reach B is equal to the number of steps that B takes to reach B_n in addition to the number of steps that A takes to move to B_n. Thus, $S2$ is given by the following formula:

$$\begin{aligned} S2 &= dist(B, B_n)/\Delta_r + dist(A, B_n)/\Delta_r \\ &= dist(B, B_1)/\Delta_r + dist(B_1, B_n)/\Delta_r + dist(A, B_n)/\Delta_r \end{aligned}$$

  Let h be the distance between y_A and y_B. Then, $dist(B, B_1) = \sqrt{dist(A, B)^2 - h^2}$.

  Let β_n be the angle from the segment $\overline{AB_n}$ to y_B in clockwise direction.

  Since robots are in situation 2, then $\beta_n < \pi$. From Lemma 8.2.4, we know that $\beta_n > 0$.

  We have $\sin \beta_n = \frac{h}{dist(A,B_n)}$, with $\beta_n > 0$. Then:

$$dist(A, B_n) = \frac{h}{\sin \beta_n} \tag{8.2}$$

  Also, $\tan(\beta_n) = \frac{h}{dist(B_1,B_n)}$. Then:

$$dist(B_1, B_n) = h \times \cot \beta_n \tag{8.3}$$

  Consequently,

$$\begin{aligned} S2 &= dist(B, B_1)/\Delta_r + dist(B_1, B_n)/\Delta_r + dist(A, B_n)/\Delta_r \\ &= \frac{\sqrt{dist(A, B)^2 - h^2} + h \times \cot \beta_n + \frac{h}{\sin \beta_n}}{\Delta_r} \\ &= \frac{\sqrt{dist(A, B)^2 - h^2} + \frac{h(\cos(\beta_n)+1)}{\sin \beta_n}}{\Delta_r} \end{aligned}$$

- Robots A and B does not share the same north. Let A and B in Situation 3 for instance. For this configuration, the longest steps that this configuration takes in order to reach the gathering is to pass through situation 5, then situation 9, and then situation 10 (or situation 6).

The maximum number of steps required for situation 3 to be transformed to situation 5 is: $S3 = \frac{dist(A,B)}{\Delta_r}$.

Now, let robots A and B be in situation 5, then this configuration requires $S5 = \frac{dist(I,A)}{\Delta_r}$ to be transformed to situation 9.

$$dist(A,B)^2 = dist(I,A)^2 + dist(I,B)^2 - 2dist(I,A) \times dist(I,B)\cos(\pi - \theta) \quad (8.4)$$

$$dist(I,A)^2 - 2dist(I,A) \times dist(I,B) + dist(I,B)^2 - dist(A,B)^2 = 0 \quad (8.5)$$

Then,

$$\begin{aligned} \Delta &= 4dist(I,B)^2\cos^2\theta - 4(dist(I,B)^2 - dist(A,B)^2) \\ &= 4dist(I,B)^2(\cos^2\theta - 1) + 4dist(A,B)^2 \\ &= 4(dist(I,B)^2\sin^2\theta + dist(A,B)^2) \end{aligned}$$

Thus,

$$\begin{aligned} dist(I,A) &= \frac{-2dist(I,B)\cos\theta + 2\sqrt{dist(I,B)^2\sin^2\theta + dist(A,B)^2}}{2} \\ &= -dist(I,B)\cos\theta + \sqrt{dist(I,B)^2\sin^2\theta + dist(A,B)^2} \end{aligned}$$

Consequently,

$$S5 = \frac{-dist(I,B)\cos\theta + \sqrt{dist(I,B)^2\sin^2\theta + dist(A,B)^2}}{\Delta_r} \quad (8.6)$$

Now, let robots A and B be in situation 9, then this configuration requires $S9 = \frac{dist(A,B)}{\Delta_r}$ to be transformed to situation 10.

Finally, let robots A and B be in situation 10, then this configuration requires $S10 = \frac{dist(A,B)}{\Delta_r}$ in order for A and B to gather.

Consequently, the maximum number of steps S required by situation 3 to be transformed to the gathering is:

$$\begin{aligned} S &= S3 + S5 + S9 + S10 \\ &= 3\frac{dist(A,B)}{\Delta_r} + \frac{-dist(I,B)\cos\theta + \sqrt{dist(I,B)^2\sin^2\theta + dist(A,B)^2}}{\Delta_r} \\ &= \frac{3dist(A,B) - dist(I,B)\cos\theta + \sqrt{dist(I,B)^2\sin^2\theta + dist(A,B)^2}}{\Delta_r} \end{aligned}$$

For the other configurations, the number of steps they require to reach the gathering configuration is less than or equal to S, and the analysis can be done in the same way.

Table 8.2: Solvability of the gathering deterministically for two robots.

	Compasses		
Model	**Inaccurate (Fixed-compass)**	**Semi-dynamic**	**None**
CORDA/SYm	Solvable for $\theta < \pi$ *(Proved in Chapter 8)*	Solvable for $\theta \leq \pi/4$ *(Proved in [41])*	Impossible *(Proved in [59, 69])*

8.4 Summary

In this chapter, we presented a tight bound on the degree of divergence of robots' compasses for solving the gathering of two asynchronous oblivious mobile robots. In particular, we gave an algorithm that solves the problem in finite time when compasses can diverge by an angle strictly less than 180°, which represents a tight bound, since it is obviously impossible to do better than this. Also, our algorithm is self-stabilizing.

It is arguable, that Algorithm 8 does not solve the gathering of robots with volume because the algorithm allows the robots to pass through each other, which is not the case in reality.

Table 8.2 summarizes the solvability of gathering deterministically for two oblivious robots.

Chapter 9

Compasses: Practical Issues

Scientists investigate that which already is;
Engineers create that which has never been.
Albert Einstein

In this chapter, we discuss practical issues related to compasses, and the implication of our results in practise. In Section 9.1, we explain the basic principle of compasses. In Section 9.2, Section 9.3, and Section 9.4, we give some examples of compasses used in robotic applications. Finally, in Section 9.5, we analyze the effectiveness of our results with respect to real compasses, and their practical applications.

9.1 Basic Principle of a Compass

A compass is the basic direction-finding device in terrestrial navigation. There are three categories of compasses: magnetic compasses depend on the earth's magnetic field, gyro-compasses (Gyro-Magnetic compasses) rely on the rotation of the earth, and solar compasses use the location of the sun and stars.

Magnetic compasses are commonly used for navigation and localization purposes. Since these two tasks are almost required in all mobile robot projects, magnetic compasses are used in several applications. Examples of such uses are in maze navigation, RoboSoccer competition, marine navigation, etc.

Manufacturing has made available different types of magnetic compasses that differ by their degree of accuracy and their cost. In general, the accuracy of compasses typically varies within 1 degree to within 10 degrees, depending on sensor quality (cost) and environment conditions.

In the following, we give a short description of the different types of magnetic compasses that are usually used in robotics applications.

Figure 9.1: The Devantech Magnetic Compass CMPS03.

9.2 Magnetic Compasses

The magnetic compass works on the principle that the earth is a giant magnet and generates a magnetic field running north and south exactly like a bar magnet. A freely suspended iron needle will align itself with the lines of magnetic force of the earth.

The magnetic compass has two important advantages:

1. It is simple.
2. It requires no power.

However, there are several difficulties with the use of a magnetic compass:

1. The earth's magnetic field does not coincide exactly with its polar axis. Thus, the magnetic north pole and the true north pole are some distance apart.
2. Metal objects, such as belt buckles, boats, automobiles, or even large iron ore deposits in the earth, can cause a compass to give misleading information. The metal interferes with the compass needle's response to the magnetic north pole.
3. The robot can create interference for its own sensor.

9.2.1 Examples of Magnetic Compasses

Devantech Magnetic Compass CMPS03:

The Devantech Magnetic Compass CMPS03 [71], is manufactured by Robot Electronics, and used for robots for navigation purposes. It uses the Philips KMZ51 magnetic field sensor, which is sensitive enough to detect the earth's magnetic field. The output from two of them mounted at right angles to each other is used to compute the direction of the horizontal component of the Earths magnetic field.

The CMPS03 compass, shown in Figure 9.1 is accurate to within 3 or 4 degrees (after calibration), and it costs about 50 US dollars.

Figure 9.2: The Hitachi HM55B Compass.

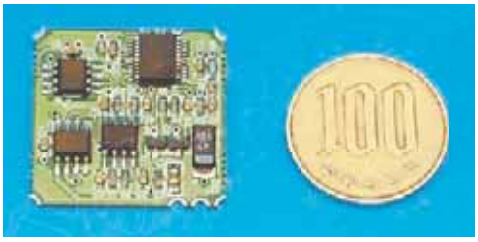


Figure 9.3: The RDCM-802 Compass.

The Hitachi HM55B Compass:

The Hitachi HM55B compass [73], shown in Figure 9.2 is made by Parallax. It is a dual-axis magnetic field sensor built around the Hitachi HM55B IC. The Hitachi HM55B compass is sensitive to microtesla (μT) variations in magnetic field strength. The degree of accuracy of the Hitachi HM55B compass varies within 6 to 11 degrees (after calibration), and it costs about 30 US dollars.

The RDCM-802 Compass:

The RDCM-802 compass [74], depicted in Figure 9.3 is another example of digital compass commercially available. It has $+/-$ 10 degrees of accuracy, and it also costs about 30 US dollars.

9.3 Gyrocompasses (Gyro-Magnetic Compasses)

To overcome the disadvantages of the magnetic compass, the gyrocompass was introduced (see Figure 9.4). A gyrocompass is a compass which finds true north by using an (electrically powered) fast spinning wheel and friction forces in order to exploit the rotation of the earth. It uses a spinning gyroscope which keeps the compass pointing not to the magnetic north, but to earth's true north. A rapidly spinning gyroscope [1] is at the heart of the gyrocompass. Gyrocompasses are widely used on ships.

[1] A gyroscope is a disk mounted on a base in such a way that the disk can spin freely on its X and Y axes; that is, the disk will remain in a fixed position in whatever directions the base is moved.

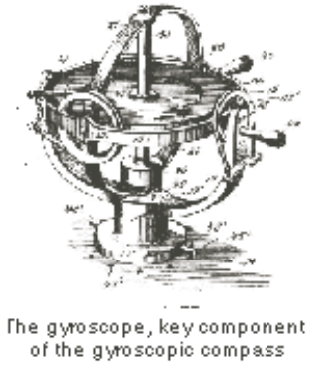


Figure 9.4: The Gyrocompasses.

Figure 9.5: The HMR3600 Digital Magnetic Compass.

Gyrocompasses have two main advantages over magnetic compasses:

1. They find true north, i.e. the direction of earth's rotational axis, as opposed to magnetic north.
2. They are not sensitive to short term transient magnetic disturbances.
3. They are not affected by metals.

However, a gyrocompass presents the following disadvantages:

1. It constantly requires electric power.
2. It must be calibrated.
3. It is more expensive than a magnetic compass.

9.3.1 Examples of Gyrocompasses

The HMR3600 Digital Magnetic Compass:

The HMR3600 digital magnetic compass [75] is a new generation sensor that combines the gyro technology with digital magnetic compass hardware and software. Its accuracy is within 0.5 degree. The inclination range is +/− 80°. The gyro compensates for short

term transient magnetic disturbances, keeping the heading output accurate, even in the presence of unexpected magnetic fields. The HMR3600 compass is depicted in Figure 9.5.

9.4 GS: Global Positioning System

Global Positioning Satellites (GS) offer the latest method of high accuracy navigation by using a constellation of 24 satellites orbiting the earth and transmitting microwave band radio frequencies across the globe. The location accuracy is anywhere from within 100 meters to within 10 meters for most equipments. GS receivers capture at least 4 of the satellite transmissions, and use difference in signal arrival times to triangulate the receiver's location. This location information is provided in the classic latitude (north-south) and longitude (east-west) coordinates given in degrees, minutes and seconds.

9.5 Implication of Results on Robotic Engineering

As we mentioned above, compasses have some limitations. In general, they are error-prone devices. Also, magnetic compasses are sensitive to magnetic interferences, and unable to tolerate short interferences. Gyrocompasses, are considered a new technology to overcome the disadvantages of magnetic compasses, are expensive, and also have some degree of inaccuracy. For these reasons, reliable software solutions are needed to circumvent these disadvantages.

Our work with eventually consistent compasses is useful in the sense that it provides a way to tolerate short interferences, and also transient failures of compasses. This means that we can equip robots with cheaper compasses (for instance magnetic compasses), and by providing algorithms based on eventually consistent compasses, we can provide more reliable solutions that overcome the disadvantages of magnetic compasses. This means that, we have made an improvement in terms of cost, and guaranteed reliability.

With gyrocompasses, although the north is fixed, there is still a divergence error. This means that, our work on inaccurate compasses (or fixed compass) is useful to overcome the limitation of the gyrocompass. In other words, robots can be equipped with gyrocompasses, but still their compasses have some divergence error. By combining, gyrocompasses and algorithmic solutions based on inaccurate compasses, we can provide robust solutions, and overcome the disadvantages of gyrocompasses.

Chapter 10

Algorithm for Circle Formation by Oblivious Robots

A circle is a round straight line
with a hole in the middle.
Online Math Learning

In this chapter, we present the complete proofs of correctness of a distributed circle formation algorithm presented within my master thesis research [64]. The algorithm allows a group of mobile robots to self-organize and position themselves into forming a circle in the SYm [69] model.

In particular, we studied the problem when robots are anonymous, oblivious, unable to communicate directly, and share no common coordinate system. More precisely, the proposed algorithm ensures that the robots deterministically form a circle, in a finite number of steps, and converges to a situation in which all robots are located evenly on the boundary of the circle. In addition, thanks to the nature of the assumed model (i.e., oblivious robots), the algorithm is also self-stabilizing. Among other things, the ability to form a circle means that the robots are spontaneously able to reach an agreement on an origin, and unit distance, albeit not on a complete coordinate system.

10.1 Problem Definition

We decompose the question of circle formation into two parts: (1) forming a circle (possibly an irregular one), and (2) positioning the robots evenly along the boundary of the circle. More rigorously, the problem is defined as follows.

Problem 10.1.1 (Uniform Circle Formation) *Given a group of n robots $r_1, r_2, \ldots, r_n$ with distinct positions and located arbitrarily on the plane, eventually arrange them at regular intervals on the boundary of some non-degenerate circle (i.e., with finite radius greater than zero).*

We also consider a weaker problem that requires the robots to form a circle, but not necessarily to be positioned at regular intervals. This weaker problem is expressed more rigorously as follows.

Problem 10.1.2 (CIRCLE FORMATION) *Given a group of n robots $r_1, r_2, \ldots, r_n$ with distinct positions and located arbitrarily on the plane, arrange them to eventually form a non-degenerate circle.*

In terms of reaching agreement, it must be obvious that the weaker problem also provides an agrement on an origin and a unit distance. At the same time, while it is conjectured that Problem 10.1.1 cannot be solved deterministically with oblivious robots, we show that Problem 10.1.2 can. In fact, we show that our algorithm solves Problem 10.1.2 within a finite number of steps, and converges toward a uniform solution (Prob. 10.1.1).

Before we proceed to the description of the algorithm, we introduce the following definitions:

Smallest enclosing circle. The *smallest enclosing circle* of a set of points P is denoted by $\mathcal{C}$, and its center is called o. It can be defined by either two opposite points, or by at least three points. The smallest enclosing circle is unique, and can be computed in $O(n)$ time [80]. We shall denote by R, the radius of $\mathcal{C}$.

Position. Given a robot r_i, $r_i(t)$ denotes its position at time t, according to some global x-y coordinate system, and $r_i(0)$ is its initial position. $P(t) = \{r_i(t) | 1 \leq i \leq n\}$ denotes the multiset of the positions of all robots at time t.

We sometimes express positions according to a polar coordinate system, with the center of the smallest enclosing circle as origin. Given a point p, we denote its polar coordinates by ρ_p and θ_p, where ρ_p is the length of the segment $\overline{op}$, and θ_p is the angle that the segment $\overline{op}$ makes with the x positive axis (in trigonometric orientation).

Alignment with the origin. Two robots are said to be *aligned with the origin* if they both have the same angular position (according to the polar coordinates). In other words, two robots are considered to be *aligned with the origin* only if they are located on the same radius (i.e, between the center and the boundary of the circle). In particular, two robots that lie on the same diameter, but on opposite sides with respect to the center, are not together aligned with the origin. This is because their respective angular positions differ by π.

Virtual ring. The robots form a virtual ring according to their respective positions. The ring is defined by looking at the angular part of the polar coordinates of the robots. Given a robot r_i, robot $prev_{r_i}$ is its direct neighbor clockwise, and robot $next_{r_i}$ is its

direct neighbor counterclockwise. In the case when robots are *aligned with the origin*, the distance from the origin is used to define the sequence. In other words, when the angle of two robots is the same, a shorter distance is regarded as being a null angle clockwise (and counterclockwise for a longer distance).

10.2 Circle Formation Algorithm

10.2.1 Algorithm Intuition

Given the SYm model [69] with oblivious robots, and an initial configuration in which a collection of robots are located arbitrarily on the plane, the algorithm ensures that the system (1) solves the *Circle Formation* problem (Prob. 10.1.2) deterministically, and (2) converges toward a solution to the *Uniform Circle Formation* problem (Prob. 10.1.1).

Informally, the algorithm relies on the fact that the smallest circle enclosing all robots is unique, and depends only on the relative positions of the robots. So, the algorithm makes sure that the smallest enclosing circle remains invariant, and uses it as a common reference. The invariance is ensured by self-imposing some constraints on the movements of the robots (Section 10.2.2). Then, robots that are in the interior of the circle are made to move toward its boundary, while the robots that are already on the boundary are made to move along the circumference.

In order to prevent the situation of inseparable robots discussed earlier, the algorithm must guarantee that no two robots move to the same location. To do so, the algorithm defines an exclusive zone for each robot, and for each activation step within which the robot must make its movement. Doing so ensures that no two robots can be at the same place at the same time. Our algorithm must rely on the fact that activations are atomic, and thus two robots activated simultaneously observe the exact same configuration (albeit according to their respective coordinate systems).[1]

10.2.2 Restrictions on Movement

We first present two restrictions imposed on the movement of robots that are located on the boundary of the smallest enclosing circle. The aim of these restrictions is to preserve the invariance of the smallest enclosing circle, that is, to prevent the robots from making movements that may lead to breaking this circle. For the sake of clarity, these restrictions do not appear explicitly in the algorithm, but must be enforced nevertheless.

Restriction 10.2.1 *Robots located on the circumference of the smallest enclosing circle do not move unless there are at least three such robots with distinct positions.*

[1] It is not difficult to extend the algorithm to work in a more loosely synchronized model in which some "fast" robots may be activated up to k-times during a single activation of the "slowest" robot, where k is a known bound.

If the smallest enclosing circle is defined by only two points, these points define a diameter of the circle. Thus, if one of the robots moves, the circle is broken.

Restriction 10.2.2 *Let $P_c(t)$ be the set of robots on the boundary of $\mathcal{C}$ at time t, and r_i be one such robot. Let $prev_{r_i}(t)$ (resp., $next_{r_i}(t)$) denote the direct clockwise (resp., counter-clockwise) neighbor of r_i on $P_c(t)$. Let also $\alpha_{prev_{r_i}}(t)$ and $\alpha_{next_{r_i}}(t)$ be the angular distance from r_i to $prev_{r_i}(t)$ and $next_{r_i}(t)$, respectively. Then, the angular movement of r_i at time $t+1$, denoted by $\alpha_m(t+1)$ is restricted as follows:*

$$\frac{\alpha_{prev_{r_i}}(t) - \pi}{2} \leq \alpha_m(t+1) \leq \frac{\pi - \alpha_{next_{r_i}}(t)}{2}$$

The above restriction ensures that the movement of robots located on the smallest enclosing circle does not leave an empty angle greater than π, or else $\mathcal{C}$ would no longer be the smallest circle enclosing all robots.

10.2.3 Algorithm Description

We now describe the algorithm in more detail, and give a pseudo-code description (see Algorithm 5).[2] As already mentioned, the robots use the smallest circle enclosing all robots $\mathcal{C}$ as the target circle for solving the problem. Starting from any configuration in which the robots are located arbitrarily on the plane (but with distinct locations), the algorithm ensures that robots located in the interior of $\mathcal{C}$ reach its boundary in a finite number of activations (Prob. 10.1.2), and that the robots located on the boundary converge to a situation where they are evenly spread on this boundary (Prob. 10.1.1). In fact, the algorithm can be seen as a combination of two algorithms that solve the two problems simultaneously.

The algorithm works as follows: when a robot r_i becomes active, it executes the following steps.

1. r_i computes the smallest enclosing circle $\mathcal{C}$, based on the observed position of the robots (Alg. 5, line 1), and changes its coordinate system to a polar one, with the origin located at point o; the center of $\mathcal{C}$.

2. If r_i happens to be located at o, then r_i moves out of the center (in any arbitrary direction) by a distance smaller than the minimal radial position of all other robots (Alg. 5, line 3). END.

3. Otherwise, r_i locates two robots $prev_{r_i}$ and $next_{r_i}$, according to the description of the virtual ring (Alg. 5, line 5).

[2]The problem is trivially solved by doing nothing for cases where there are only one or two robots. Therefore, in the rest of the section we consider the cases with three or more robots.

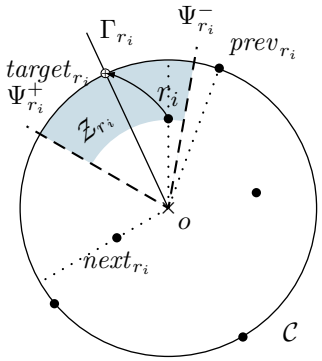


Figure 10.1: Principle of the algorithm.

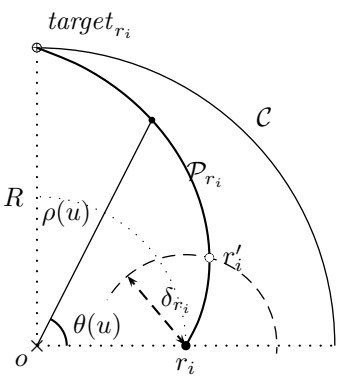


Figure 10.2: Parametric path $\mathcal{P}_{r_i}$ computed by robot r_i.

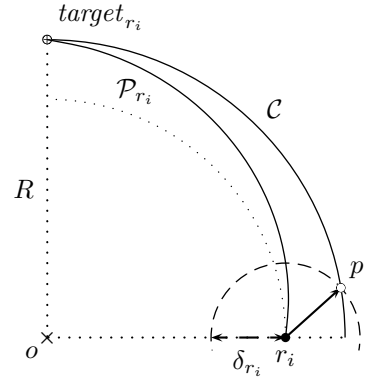


Figure 10.3: $target_{r_i}$ is out of reach, while $\mathcal{C}$ is not; r_i joins $\mathcal{C}$ at point p.

Algorithm 5 Circle Formation Algorithm for Oblivious Robots

function $\varphi_{circle\_uniform}(P, r_i)$

$\mathcal{C}$:= smallest circle enclosing all points in P;
if (r_i = center of $\mathcal{C}(P)$) **then**
  r_i moves to an arbitrary location by some radius ρ_{r_i} less than the minimum radius of all other robots;
else
  Compute $prev_{r_i}$ and $next_{r_i}$ (see Sect. 10.1)
  if ($prev_{r_i}, r_i, next_{r_i}$) are *aligned with the origin* **then**
    stay still;
  else
    $\alpha_{prev_{r_i}}$:= angular distance between r_i and $prev_{r_i}$ in clockwise orientation;
    $\alpha_{next_{r_i}}$:= angular distance between r_i and $next_{r_i}$ in counterclockwise orientation;
    $\Psi^-_{r_i}$:= bisector of the angle $\alpha_{prev_{r_i}}$;
    $\Psi^+_{r_i}$:= bisector of the angle $\alpha_{next_{r_i}}$;
    Γ_{r_i}:= bisector of the angle formed by $\Psi^-_{r_i}$ and $\Psi^+_{r_i}$;
    $target_{r_i} := \Gamma_{r_i} \cap \mathcal{C}$;
    Compute path $\mathcal{P}_{r_i}$ from r_i to $target_{r_i}$ (Eq. (10.2));
    if $dist(r_i, \mathcal{C}) \leq \delta_{r_i}$ **then**
      Move to $\mathcal{C}$;
    else
      Move along $\mathcal{P}_{r_i}$ toward $target_{r_i}$ by δ_{r_i};
    end if
  end if
end if

4. If $prev_{r_i}$, r_i, and $next_{r_i}$ are together *aligned with the origin*, then r_i does nothing (Alg. 5, line 7). END.

5. If not, then r_i computes three rays starting from o, called $\Psi^-_{r_i}$, $\Psi^+_{r_i}$, and Γ_{r_i} (see Figure 10.1). $\Psi^-_{r_i}$ is defined as the bisector of the angle $\alpha_{prev_{r_i}} = \angle r_i o prev_{r_i}$, and $\Psi^+_{r_i}$ is defined similarly for $next_{r_i}$. Γ_{r_i} is the bisector of the angle formed by $\Psi^-_{r_i}$ and $\Psi^+_{r_i}$ (Alg. 5, line 13).

The algorithm must prevent two robots activated simultaneously from moving to the same location because, otherwise, it may become impossible to separate them (i.e., there exists an activation schedule whereby the robots always move together). To prevent this situation from occurring, we define a zone in which r_i alone is allowed to move during that activation. We call such a zone the *exclusive zone* of robot r_i for activation time t, denoted $\mathcal{Z}_{r_i}(t)$, and defined as follows:

$$\mathcal{Z}_{r_i}(t) = \{r_i(t)\} \cup \left\{ p \in \mathbb{R}^2 \ \mid \ (\rho_{r_i(t)} \leq \rho_p \leq R) \wedge (\alpha_{\Psi^-_{r_i}(t)} < \alpha_p < \alpha_{\Psi^+_{r_i}(t)}) \right\} \tag{10.1}$$

The zone is depicted as a gray area in Figure 10.1. It is important to stress that the bisectors $\Psi^-_{r_i}$ and $\Psi^+_{r_i}$ do not belong to the exclusive zone of r_i. In fact, when the three robots $prev_{r_i}$, r_i, and $next_{r_i}$ are aligned with the origin, $\Psi^+_{r_i}$ and $\Psi^-_{r_i}$ are coincident, and thus $\mathcal{Z}_{r_i}$ includes only the current position of r_i. We now resume the description of the algorithm.

6. Based on Γ_{r_i}, r_i computes a target location $target_{r_i}$, as the intersection of Γ_{r_i} with $\mathcal{C}$. Notice that, by definition, $target_{r_i}$ is always located in $\mathcal{Z}_{r_i}$ (Alg. 5, line 14).

7. If r_i can reach $target_{r_i}$ directly, then it moves there. END.

8. If r_i cannot reach $target_{r_i}$ directly, but can reach $\mathcal{C}$, then it moves[3] to the reachable point on $\mathcal{C}$ that is nearest to $target_{r_i}$ (see Figure 10.3). Note that this point must be within $\mathcal{Z}_{r_i}$ of r_i. (Alg. 5, line 17) END.

9. Otherwise, r_i computes a parametric path $\mathcal{P}_{r_i}$ from r_i to $target_{r_i}$, as a linear motion in the polar space (see definition of $\mathcal{P}_{r_i}$ below). r_i moves as far as possible (i.e, maximum is δ_{r_i}) along this path (see Figure 10.2). END.

The parametric path $\mathcal{P}_{r_i}$ computed by a robot r_i at time t is defined by the following equations:

$$\mathcal{P}_{r_i}(t) = \begin{cases} \theta(u) = \theta_{r_i(t)} + u(\theta_{target_{r_i}(t)} - \theta_{r_i(t)}) \\ \rho(u) = \rho_{r_i(t)} + u(R - \rho_{r_i(t)}) \\ 0 \leq u \leq 1 \end{cases} \tag{10.2}$$

[3]The movement of Step 8 may seem surprising at first. This movement is used to compute an upper bound on the number of activations necessary for robot r_i to reach the boundary of $\mathcal{C}$ (see Lemma 10.3.13). Without this movement, some situation may occur when $target_{r_i}$ remains out of reach at every activation (because it rotates), and robot r_i is unable to reach $\mathcal{C}$ in finite time due to the Zeno paradox.

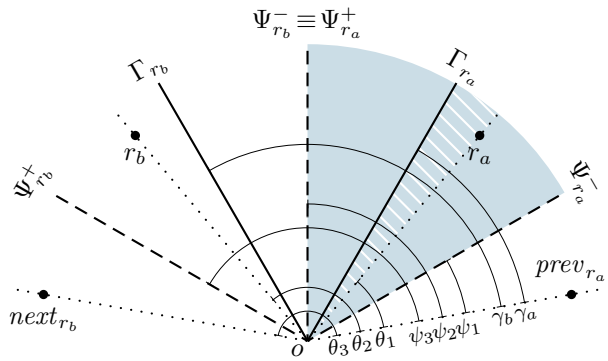


Figure 10.4: Invariance of virtual ring: consecutive robots r_a and r_b.

10.3 Correctness

In this section, we prove the correctness of our algorithm by first showing that no two robots ever move to the same location (Theorem 10.3.5). Second, we prove that the smallest enclosing circle remains invariant (Theorem 10.3.6). Then, we show that all robots reach the boundary of the circle in finite time (Theorem 10.3.15). Finally, we prove that the algorithm converges toward a configuration wherein all robots are located at regular intervals on the circle (Theorem 10.3.22).

We first state two lemmas that derive trivially from Algorithm 5.

Lemma 10.3.1 *No robot ever moves beyond the boundary of the smallest circle enclosing all robots.*

Lemma 10.3.2 *All robots located on the boundary of the smallest enclosing circle remain on that boundary.*

10.3.1 Non-overlapping Zones

We begin by establishing the common context in which we prove several lemmas.

Let us consider some arbitrary time t, and an arbitrary pair of robots r_a and r_b, such that $r_b = next_{r_a}$ at time t (i.e., r_a and r_b are consecutive at t) and no two robots are located at the same position. The rest of the argument can be repeated for any time and any pair of consecutive robots.

We consider the four robots $prev_{r_a}$, r_a, r_b, and $next_{r_b}$ and their relative angles at time t. We set the reference angle of our polar coordinate system to be the angular position of robot $prev_{r_a}$ (see Figure 10.4). Let θ_1, θ_2, and θ_3 denote the angles of robots r_a, r_b, and $next_{r_b}$, respectively. We also consider the bisectors $\Psi^-_{r_a}$, $\Psi^+_{r_a}$, $\Psi^+_{r_b}$, used in the definition of the movement. Notice that $\Psi^-_{r_b} \equiv \Psi^+_{r_a}$ because $r_a = prev_{r_b}$. Let ψ_1, ψ_2, and ψ_3 denote the angles of $\Psi^-_{r_a}$, $\Psi^+_{r_a}$, and $\Psi^+_{r_b}$, respectively. Finally, we consider the two second-order bisectors Γ_{r_a} and Γ_{r_b}, and let γ_a and γ_b denote their respective angles. Remember that the respective targets of r_a and r_b are located on Γ_{r_a} and Γ_{r_b}.

From this, we obtain the following relations between those angles.

$$\begin{array}{ccccccccccccc} 0 & \leq & \psi_1 & \leq & \theta_1 & \leq & \psi_2 & \leq & \theta_2 & \leq & \psi_3 & \leq & \theta_3 \\ & & \| & & & & \| & & & & \| & & \\ & & \psi_1 & \leq & \gamma_a & \leq & \psi_2 & \leq & \gamma_b & \leq & \psi_3 & & \end{array} \tag{10.3}$$

Lemma 10.3.3 *There is no overlap between the exclusive zones of any two consecutive robots.*

PROOF. We consider the situation above and reason about the angles. The exclusive zone of robot r_a consists of the position of r_a and a zone included in the open angular interval $(\psi_1; \psi_2)$. Note that, because it is open, the interval can possibly be empty (when $\psi_1 = \psi_2$). Similarly, the zone of r_b consists of the position of r_b and a zone included in the interval $(\psi_2; \psi_3)$.

1. The locations of r_a and r_b are distinct by hypothesis.
2. The intervals do not intersect. The intervals are open, which means that the points on the rays do not belong to the zones. We simply need to show that $\psi_1 < \psi_3$, but this is already obvious from Relation (10.3).
3. The location of one of the two robots (say r_a) does not belong to the interval of the other robot (say r_b). Consider the angular position of r_a, θ_1, and the interval of r_b, $(\psi_2; \psi_3)$. By Relation (10.3), we have that $\theta_1 \leq \psi_2 \leq \psi_3$. Since the rays do not belong to the interval, r_a is not in the interval of r_b, even when $\theta_1 = \psi_2$.

$\square_{\text{Lemma 10.3.3}}$

Lemma 10.3.4 *There is no overlap between the exclusive zones of any two robots.*

PROOF. The proof is a generalization of Lemma 10.3.3, by a simple induction on a string of consecutive robots.

A special case occurs when a robot is located at the center of the smallest enclosing circle. This is treated separately. Let r_o be that robot. It must be unique by hypothesis. The zone of r_o is defined by the circle centered at o and with radius r, such that $r < \min_{r \in \mathcal{R} \setminus \{r_o\}} \rho_r$. Since the points in the zone of any other robot r must have a radial position of at least ρ_r, there can be no intersection with the zone of r_o. $\square_{\text{Lemma 10.3.4}}$

Theorem 10.3.5 *Under Algorithm 5, no two robots ever move to the same location.*

PROOF. We show that a robot r_i always moves to a location within its own exclusive zone $\mathcal{Z}_{r_i}$, and the rest follows from the fact that the zones of two robots do not intersect (Lemma 10.3.4). Let us consider a robot r_i and its new location r'_i. There are two cases.

In the first case, $prev_{r_i}$, r_i, and $next_{r_i}$ are aligned together with the origin. The location of r_i belongs to the zone ($\mathcal{Z}_{r_i}$ is equal to the location of r_i), and r_i does not move.

In the second case, $prev_{r_i}$ and $next_{r_i}$ are not aligned. Then, Γ_{r_i} is located between $\Psi^-_{r_i}$ and $\Psi^+_{r_i}$, and all three are distinct. It follows that $target_{r_i}$ is strictly between $\Psi^-_{r_i}$ and $\Psi^+_{r_i}$ (and thus lies in $\mathcal{Z}_{r_i}$). r_i is also between $\Psi^-_{r_i}$ and $\Psi^+_{r_i}$, but not strictly (i.e., r_i can be on either one of the two axes). Because r_i belongs to its zone, and because the angle of points in the path are defined linearly, all points between r_i and $target_{r_i}$ must be in $\mathcal{Z}_{r_i}$. $\square_{\text{Theorem 10.3.5}}$

10.3.2 Invariance of the Smallest Enclosing Circle

From Lemma 10.3.1, Lemma 10.3.2, Restriction 10.2.1-10.2.2 and Theorem 10.3.5, we obtain the following theorem:

Theorem 10.3.6 *The smallest enclosing circle $\mathcal{C}$ is invariant.*

PROOF. Let $\mathcal{C}(t)$ and $\mathcal{C}(t+1)$ denote the smallest enclosing circle at time instants t and $t+1$ respectively. We prove that, regardless of the activation schedule, $\mathcal{C}(t)$ and $\mathcal{C}(t+1)$ must be identical, and the rest follows by induction.

Assume, by contradiction, that there is a time instant t for which $\mathcal{C}(t)$ and $\mathcal{C}(t+1)$ are different. First, we observe that this cannot be caused by the movement of a robot located at the interior of $\mathcal{C}(t)$. Indeed, such a robot could change the smallest enclosing circle only by moving outside of it, (a contradiction with Lemma 10.3.1). Therefore $\mathcal{C}(t+1)$ must be defined by the movement of robots located at the boundary of $\mathcal{C}(t)$. There are four cases left to consider, depending on the number of robots at the boundary of $\mathcal{C}(t)$, and their respective positions:

1. *(2 robots)* The smallest enclosing circle $\mathcal{C}(t)$ is defined by only two robots. Those robots cannot move by Restriction 10.2.1 and hence $\mathcal{C}(t+1) = \mathcal{C}(t)$.
2. *(3 robots; one quits the circle)* The smallest enclosing circle $\mathcal{C}(t)$ is defined by three robots, one of which moves outside the boundary of $\mathcal{C}(t)$. This is a contradiction of Lemma 10.3.1.
3. *(3 robots; two distinct points)* The smallest enclosing circle $\mathcal{C}(t)$ is defined by three robots, two of which move to the same location. This is in contradiction of Theorem 10.3.5.
4. *(3 robots; angular distance greater than diameter)* If the angular distance between two of the three robots is larger than the diameter, then the circle defined by the three robots and the smallest enclosing circle for the two robots are different. Since $\mathcal{C}(t)$ is the smallest enclosing circle at time t, the angular distance between any two

of the three robots must be not greater than the diameter. By Restriction 10.2.2, the movement of two consecutive robots cannot lead them further away from each other than π, regardless of their activation schedule.

When there are more than three robots on the boundary of $\mathcal{C}(t)$, the situation can always be reduced to one of the four cases mentioned above. It follows that $\mathcal{C}(t)$ and $\mathcal{C}(t+1)$ cannot be different; a contradiction. $\square_{\text{Theorem 10.3.6}}$

The following lemma is obtained easily from the algorithm.

Lemma 10.3.7 *For any robot r_i, its radial position $\rho_{r_i}(t)$ is nondecreasing.*

Lemma 10.3.8 *There is a time since after which no robot is at the center of $\mathcal{C}$.*

PROOF. Let r_o be a robot located at the center of $\mathcal{C}$. By the fairness of the activation, there is a time t when it becomes active. From line 3 of Algorithm 5, r_o is no longer at the center at time $t+1$. From Lemma 10.3.7, the radial position is nondecreasing, and thus no robot can be located at the center of $\mathcal{C}$ after time t. $\square_{\text{Lemma 10.3.8}}$

10.3.3 Invariance of the Virtual Ring

Theorem 10.3.9 *From the time when no robot is located at the center of $\mathcal{C}$, the virtual ring remains invariant.*

PROOF. We consider again the situation of Section 10.3.1, and we must show that, at time $t+1$, r_a must be before r_b, and the rest follows by applying the same argument to all pairs of consecutive robots.

The position of r_a at time $t+1$ must be between the axes of r_a and Γ_{r_a} (i.e., the hatched zone in Figure 10.4). This means that the angular position must be in the angular interval $I_a = [\min(\theta_1, \gamma_a); \max(\theta_1, \gamma_a)]$. Similarly, the new position of r_b must be in the interval $I_b = [\min(\theta_2, \gamma_b); \max(\theta_2, \gamma_b)]$.

By definition, the position that r_a will take at time $t+1$ must also be located within the zone of r_a at time t.

Then, we need to distinguish two cases.

1. $\theta_1 < \theta_2$. From this and the fact that most angles are defined as bisectors, we can refine Relation (10.3) as follows.

$$\begin{array}{ccccccccccccc} 0 & \leq & \psi_1 & \leq & \theta_1 & < & \psi_2 & < & \theta_2 & \leq & \psi_3 & \leq & \theta_3 \\ & & \| & & & & \| & & & & \| & & \\ & & \psi_1 & < & \gamma_a & < & \psi_2 & < & \gamma_b & < & \psi_3 & & \end{array}$$

From the above relation, we can directly derive.

$$\max(\theta_1, \gamma_a) < \min(\theta_2, \gamma_b)$$

Thus, the order between r_a and r_b is preserved.

2. $\theta_1 = \theta_2$. The two robots r_a and r_b are aligned with the origin. The only points of that ray that belong to their zone are their respective locations. In this case, the order is defined by the distance from the origin, which cannot change at time $t+1$ because of the invariance of the smallest enclosing circle (Theorem 10.3.6). Since all other points in the zone of r_a, if they exist, have an angle strictly smaller than $\theta_1 = \theta_2$, and strictly greater for r_b, the order between r_a and r_b is preserved.

$\square_{\text{Theorem 10.3.9}}$

10.3.4 Circle Formation

In the following, we will show that all robots located in the interior of $\mathcal{C}$ reach its boundary after a finite number of activation steps.

We have observed that, at each time instant a robot r_i becomes active, it computes a new target (the target is dynamic). Depending on the activation of the neighbors of r_i, its target at time $t+1$ can be closer or farther than at time t. However, we also observed that the maximum angle that can separate a robot from its target is $\frac{\pi}{4}$. Then, before proceeding, we establish the following lemma.

Lemma 10.3.10 *The angle that separates a robot r_i from its target $target_{r_i}$ is at most $\frac{\pi}{4}$.*

Proof. By Restriction 10.2.2, the maximum angular distance that can separate any two consecutive robots is π. Consider some robot r_i, the extreme case occurs where r_i forms a minimal angle with one of its neighbors, say $prev_{r_i}$, and a maximal angle with its other neighbor, say $next_{r_i}$. Let us thus consider the situation where r_i and $prev_{r_i}$ are aligned with the origin at angle 0, and where the angular distance between r_i and $next_{r_i}$ is π.

It follows that $\Psi^-_{r_i}$ is at a null angle with respect to r_i, while $\Psi^+_{r_i}$ is at angle $\frac{\pi}{2}$. Being the bisector of $\Psi^-_{r_i}$ and $\Psi^+_{r_i}$, Γ_{r_i} is at angle $\frac{\pi}{4}$. Since $target_{r_i}$ is located on Γ_{r_i}, this proves the lemma. $\square_{\text{Lemma 10.3.10}}$

Lemma 10.3.11 *For any robot r_i that is not aligned with the origin, and with its previous and next neighbors, there exists a minimum distance $d_{min,r_i} > 0$ that r_i can progress toward the boundary of the circle.*

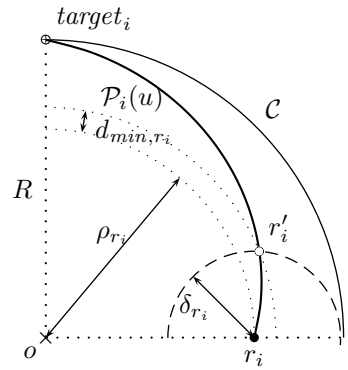


Figure 10.5: The minimum distance of progress of r_i toward the boundary of the circle is d_{min,r_i}.

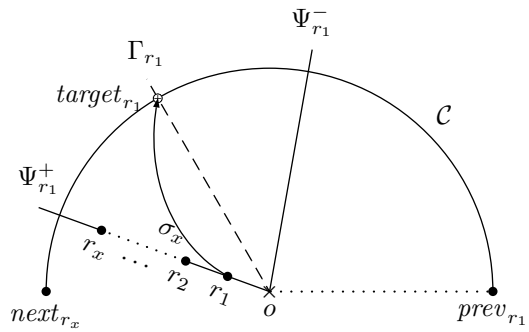


Figure 10.6: String of robots *aligned with the origin.*

PROOF. To prove the lemma, we consider the situation where r_i can progress the least. It is easy to see that this situation occurs when the angular distance with the target is maximal (i.e., $\frac{\pi}{4}$ by Lemma 10.3.10) and r_i is as close as possible to $\mathcal{C}$ without being able to reach it (see Figure 10.5).

Observe that r_i can progress away from the center of $\mathcal{C}$ by at least d_{min,r_i} when moving toward $target_{r_i}$. In this situation, the range of r_i (δ_{r_i}) is just too short to reach $\mathcal{C}$. Thus, r_i will move to location r'_i. d_{min,r_i} is equal to the difference between $\rho_{r'_i}$ and ρ_{r_i}, and it is positive. Thus, $d_{min,r_i} > 0$ represents the minimum distance that r_i can move away from the center of $\mathcal{C}$, and the lemma holds. $\square_{\text{Lemma 10.3.11}}$

Lemma 10.3.12 *By the algorithm, starting from any configuration in which some robots are aligned with the origin, there is a time after which no two robots are aligned with the origin.*

PROOF. We consider an arbitrary string of x robots $\sigma_x = r_1, \cdots, r_x$ with increasing distance from the origin, and aligned together with the origin (see Figure 10.6). First, it is easy to see that no new robot joins σ_x (see proof of Theorem 10.3.5), and then the rest of the proof is by induction on x, the number of robots at σ_x.

Basis: ($x = 1$). The lemma holds trivially.

Induction Step: Assume that the lemma holds for any string σ_y shorter than x ($y < x$), and let us prove that the lemma holds for a string σ_x of length x. Let us consider one of the two robots at the extremity of the string, say r_1 (the argument is the same for r_x).

By assumption, the scheduler is fair, hence eventually r_1 becomes active. Since r_1 is at the extremity of the string, r_1 and $prev_{r_1}$ cannot be aligned together with the origin, and thus the test on line 6 in the algorithm results in false. So, r_1 computes a path $\mathcal{P}_{r_1}$ at line 15.

r_1 and $prev_{r_1}$ not being aligned with the origin, means that $\Psi^-_{r_1}$ and $\Psi^+_{r_1}$ are distinct, and so is Γ_{r_1}. It follows that $target_{r_1}$ has an angular position different from that of r_1. Thus, except for the initial location of robot r_1, no other point on $\mathcal{P}_{r_1}$ is aligned with $\Psi^+_{r_1}$ and the other robots of the string. Because δ_{r_1} is greater than zero, the destination r'_1 of r_1 cannot be aligned with the robots of σ, regardless of the test in line 16. Thus, after its move, r_1 no longer belongs to the string σ, thus decreasing its length by one. This proves the induction step. $\square_{\text{Lemma 10.3.12}}$

Lemma 10.3.13 *All robots located in the interior of $\mathcal{C}$ reach its circumference in finite time.*

PROOF. By Lemma 10.3.12, if there exists a configuration wherein some robots are *aligned with the origin*, there is a finite number of steps, in which this configuration is reduced to the general case. From Lemma 10.3.11, at each activation step, a robot r_i, not located on the boundary of $\mathcal{C}$, can progress by at least a radial distance $d_{min,r_i} > 0$ toward the periphery of the circle. It follows that, regardless of the initial position of some robot r_i, the number of activation steps it takes for r_i to reach the boundary of $\mathcal{C}$ is bounded above by $\frac{R}{d_{min,r_i}}$. Thus, due to the fairness of the activation schedule, the boundary of $\mathcal{C}$ is reached in finite time, and the lemma holds. $\square_{\text{Lemma 10.3.13}}$

Lemma 10.3.14 *The global predicate that all robots are located on the boundary of $\mathcal{C}$ is stable.*

PROOF. Let us denote by C_{circle}, the set of all configurations in which all robots are located on the boundary of $\mathcal{C}$. Then, we show that, for any configuration c in C_{circle}, the algorithm always leads to a configuration c' in C_{circle}.

Consider some robot r_i that becomes active. By the algorithm, r_i computes a new $target_{r_i}$, located on $\mathcal{C}$. Because r_i is also on $\mathcal{C}$, the entire path $\mathcal{P}_{r_i}$ is located on $\mathcal{C}$. Thus, r_i can only move to a location on the boundary of $\mathcal{C}$. It follows that configuration c' is in C_{circle}. $\square_{\text{Lemma 10.3.14}}$

Theorem 10.3.15 *The algorithm solves the circle formation problem deterministically.*

PROOF. There is a time after which all robots are located on the boundary of the circle (Lemma 10.3.13), and this situation is stable (Lemma 10.3.14). $\square_{\text{Theorem 10.3.15}}$

10.3.5 Uniform Transformation

We now show that our algorithm *converges* toward a uniform distribution of robots along the boundary. Before we proceed, we give a few additional definitions:

Definition 10.3.16 *For any robot r_i, let $\alpha_{r_i}(t)$ denote the angular distance between r_i and $next_{r_i}$. Thus, $\alpha_{r_i}(t) = \theta_{next_{r_i}}(t) - \theta_{r_i}(t)$.*

Definition 10.3.17 *Let $\alpha_{max}(t)$ (resp., $\alpha_{min}(t)$) be the maximal (resp., minimal) angular distance between any two consecutive robots, at time t. Thus, $\alpha_{max}(t) = \max_{r_i} \alpha_{r_i}(t)$ and $\alpha_{min}(t) = \min_{r_i} \alpha_{r_i}(t)$.*

Lemma 10.3.18 *The function $\alpha_{max}(t)$ is nonincreasing, and the function $\alpha_{min}(t)$ is nondecreasing.*

PROOF. We only prove the lemma for $\alpha_{max}(t)$, as the proof for $\alpha_{min}(t)$ is then easily derived by symmetry.

Let t be some time, and r_i a robot. Obviously, $\alpha_{r_i}(t+1)$ is maximized when (1) both robots r_i and $next_{r_i}$ are active at time t, (2) they are moving away from each other, and (3) they can reach their respective target points.

Thus, assuming that both robots r_i and $next_{r_i}$ are active at time t, we obtain:

$$\begin{aligned} \alpha_{r_i}(t+1) &= \frac{\alpha_{r_i}(t)/2 + \alpha_{next_{r_i}}(t)/2}{2} + \frac{\alpha_{r_i}(t)/2 + \alpha_{prev_{r_i}}(t)/2}{2} \\ &= \frac{2\alpha_{r_i}(t) + \alpha_{next_{r_i}}(t) + \alpha_{prev_{r_i}}(t)}{4} \\ &\leq \alpha_{max}(t) \end{aligned} \tag{10.4}$$

The inequality is obtained by replacing $\alpha_{r_i}(t)$, $\alpha_{prev_{r_i}}(t)$ and $\alpha_{next_{r_i}}(t)$ by $\alpha_{max}(t)$. It follows that, for any time t, $\alpha_{max}(t+1) \leq \alpha_{max}(t)$. $\square_{\text{Lemma 10.3.18}}$

Corollary 10.3.19 $\forall t, \forall r_i : \alpha_{min}(t) \leq \alpha_{r_i}(t+1) \leq \alpha_{max}(t)$

Lemma 10.3.20 *Every configuration in which all robots are uniformly distributed on the boundary of the circle is stable.*

PROOF. Assume that, at some time t, the robots are uniformly distributed. In such a configuration, the angular distance between any two consecutive robots must be the same: $\frac{2\pi}{n}$. It follows that, $\alpha_{min}(t) = \alpha_{max}(t) = \frac{2\pi}{n}$, from which we derive,

$$\forall t, \forall r_i : \frac{2\pi}{n} = \alpha_{min}(t) \leq \alpha_{r_i}(t+1) \leq \alpha_{max}(t) = \frac{2\pi}{n}$$

and this completes the proof. $\square_{\text{Lemma 10.3.20}}$

Lemma 10.3.21 *The function $\Delta(t) = \alpha_{max}(t) - \alpha_{min}(t)$ is monotonically decreasing and converges to zero.*

PROOF. First of all, from Lemma 10.3.18, we can deduce that $\Delta(t)$ is nonincreasing. We must show that, for any time t, if $\alpha_{min}(t) < \alpha_{max}(t)$, then, eventually, either $\alpha_{min}(t')$ increases or $\alpha_{max}(t')$ decreases. In other words,

$$\forall t : \alpha_{min}(t) < \alpha_{max}(t) \Rightarrow (\exists t' > t : (\alpha_{max}(t') < \alpha_{max}(t)) \vee (\alpha_{min}(t) < \alpha_{min}(t')))$$

First, let us show that an angle $\alpha_{r_i}(t)$ strictly smaller than $\alpha_{max}(t)$ at time t, must always be smaller than $\alpha_{max}(t)$ after time t (although $\alpha_{r_i}(t)$ can possibly increase). In other words,

$$\forall t \forall r_i : \alpha_{r_i}(t) < \alpha_{max}(t) \Rightarrow (\forall t' > t : \alpha_{r_i}(t') < \alpha_{max}(t))$$

This is done easily by induction. Consider that, at time t, $\alpha_{r_i}(t) < \alpha_{max}(t)$. From Equation (10.4) in the proof of Lemma 10.3.18, we have:

$$\alpha_{r_i}(t+1) = \frac{2\alpha_{r_i}(t) + \alpha_{prev_{r_i}}(t) + \alpha_{next_{r_i}}(t)}{4}$$

From which we deduce that $\alpha_{r_i}(t+1) < \alpha_{max}(t)$. Since, by Lemma 10.3.18, $\alpha_{max}(t+1) \leq \alpha_{max}(t)$, we indeed have that, for any time t' after t, $\alpha_{r_i}(t') < \alpha_{max}(t)$.

To complete the proof of the lemma, we must now show that, if an angle $\alpha_{r_i}(t)$ is maximal at time t ($\alpha_{r_i}(t) = \alpha_{max}(t)$), then there must be a time t' in the future when it becomes smaller. In other words,

$$\forall t \forall r_i : \alpha_{r_i}(t) = \alpha_{max}(t) \Rightarrow (\exists t' > t : \alpha_{r_i}(t') < \alpha_{max}(t))$$

Observe that if $\alpha_{r_i}(t)$ is equal to $\alpha_{max}(t)$, then $\alpha_{r_i}(t)$ decreases only when $\alpha_{prev_{r_i}}(t)$ is less than $\alpha_{max}(t)$.

Assume that $\alpha_{r_i}(t) = \alpha_{prev_{r_i}}(t) = \alpha_{max}(t)$. Since, $\alpha_{min}(t) < \alpha_{max}(t)$ by hypothesis, and there is a finite number of robots, then there must be some robot r_j such that $\alpha_{r_j}(t) \leq \alpha_{max}(t)$ and $\alpha_{prev_{r_j}}(t) < \alpha_{max}(t)$.

By the fairness of the scheduler, there must be a time t'' for r_j when $\alpha_{r_j}(t'') < \alpha_{max}(t)$. By applying induction repeatedly on the robots, we obtain that from some time t''', and for all robots r_k, $\alpha_{r_k}(t''') < \alpha_{max}(t)$.

The same proof can be adapted for the minimum, and we have that, for any time t when $\alpha_{min}(t) < \alpha_{max}(t)$, there will be a time t' in the future when $\alpha_{max}(t') < \alpha_{max}(t)$ and $\alpha_{min}(t') > \alpha_{min}(t)$. Thus, $\Delta(t) = \alpha_{max}(t) - \alpha_{min}(t)$ converges toward zero. $\square_{\text{Lemma 10.3.21}}$

Theorem 10.3.22 *Algorithm 5 converges toward a configuration wherein all robots are arranged at regular intervals on the boundary of the circle.*

The theorem comes as a direct consequence of Lemma 10.3.20 and Lemma 10.3.21.

10.4 Summary

In this chapter, we have complemented a prior work in the circle formation problem, by developing complete and rigorous proofs for correctness of a prior algorithm presented within my master thesis research. The algorithm allows a team of oblivious mobile robots to self-organize to form a circle. The algorithm allows a team of oblivious robots to deterministically form the circle within a finite number of activation steps, and asymptotically converges toward a uniform distribution of the robots along the circumference of the circle. Moreover, it is intrinsically self-stabilizing, due to the assumption that robots are oblivious.

Our algorithm has several important advantages over the algorithm proposed by Défago and Konagaya [23]. Most importantly, it is simpler in many different ways. Firstly, it elegantly combines the solution of the two problems into a single algorithm. Secondly, the only somewhat complex geometric computation on which it relies is the smallest enclosing circle, which is well-known and well-studied problem in computational geometry. Finally, the computation complexity is smaller. Indeed, finding the smallest enclosing circle can be achieved in $O(n)$, whereas computing the Voronoi diagram is normally done in $O(n \log n)$.

For systems in which robots do have a memory, the algorithm can be further optimized by relying on the invariance of the smallest enclosing circle. Then, the self-stabilizing properties can still be preserved, provided that the validity of the smallest enclosing circle cached in memory is verified before each activation.

Chapter 11

Discussion: Gathering Robots with Volume

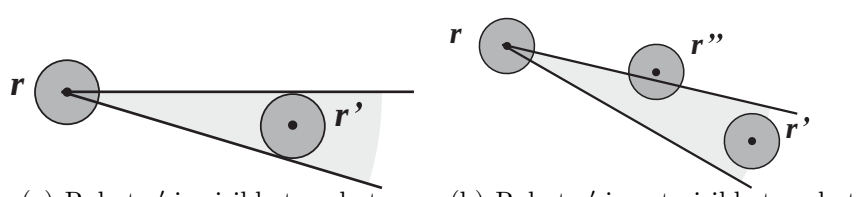
In this chapter, we give a brief discussion on the gathering of robots with volume by pointing out the different problems that arise once we add the realistic aspect of volume to the robots.

In this dissertation, we considered system models where robots are represented by points in the plane. However, these models are not realistic with respect to this aspect of representation of robots by points. In reality, even very small robots occupy some space.

In this section, we would like to discuss the difficulties of solving problems when we add the realistic aspect of volume to the robots. To the best of our knowledge, only Czyzowicz et al. [20] have considered this problem so far. In particular, they proposed an algorithm that solves the gathering for at most four robots. The question of the existence of a solution for any number of robots was however left as an open question. This is can be explained by the difficulty of the task.

Czyzowicz et al. [20] have observed that adding the realistic aspect of volume to the robots (fat robots), significantly complicates the task of gathering. In particular, the fatness of robots results into two main complications. First, some robots may prevent full visibility of others. Second, some robots may mechanically obstruct the motion of others, staying or getting in their line of move.

In other words, while solving problems, robots need to avoid *collisions*, and also they should solve the problem of *partial visibility*, that is some robots should not obstruct vision



(a) Robot r' is visible to robot r.

(b) Robot r' is not visible to robot r.

Figure 11.1: Visibility of robots with volume.

of others. This results on proposing new definitions for problems, and also for visibility between robots.

We introduce the following definitions concerning robots with volume that are represented by unit discs in the plane.

Definition 11.0.1 (Visible Robots) *Informally, we say that robot r' is visible to robot r if r' belongs to the field of vision of robot r and there is no other robot that is located between r and r'.*

Formally, robot r' is ***visible*** *to robot r if the sector passing through the center of robot r encapsulates all the disc of robot r', and there is no other robot that intersects this sector.*

An example of two visible and non visible robots are depicted in Figure 11.1

Definition 11.0.2 (Gathering Robots with Volume) *We say that robots achieve the gathering if they form a pattern with a minimal diameter of convex hull, such that any two robots have full visibility of each other.*

Definition 11.0.3 (Convergence of Robots with Volume) *We say that robots achieve the convergence toward the gathering if they converge toward a pattern with a minimal diameter of convex hull, such that any two robots have full visibility of each other.*

Solving the gathering problem for robots with volume as defined above is very complicated starting from five robots. The first main problem is to find the pattern with minimal diameter of convex hull in which these robots can be placed, and they all have full visibility of each other. We thus, propose a weaker definition of the gathering where robots can be placed in a minimal ring. Solving the convergence is less difficult, however it depends to what extent we can ignore errors in practise.

Definition 11.0.4 (Weak Gathering of Robots with Volume) *A weak gathering is the formation of a minimal circle in which all robots have a full visibility of each other, and they are uniformly spread over the circle.*

This problem of weak gathering has interesting applications. For instance, the ability to gather on a ring means that robots can reach an agreement on a common origin, and a unit distance.

At a first glance, we may think that solving the *weak gathering* is easy, and this problem can be reduced to solving the circle formation problem. However, this is not true. In particular, when solving the weak gathering of robots with volume, we need to deal in addition with the following problems:

- Finding minimal radius of circle: We need to find a formula that determines the minimal radius of the circle in which robots with volume can be placed so that they can gather as in Definition 11.0.4.

- Solving the problem of partial visibility: Some robots may obstruct vision of each other. So, each robot need to move in such a way to get full visibility with all other robots in the system to be aware that gathering is accomplished.

- Detection of full visibility: Robots need to detect when all robots in the system are visible in order to start to form the gathering circle. This problem is easy to deal with when the number of robots in the system is known. However, the problem becomes more complicated when the number of robots in the system in unknown to the robots.

- Collision avoidance: While moving, robots need to avoid collisions between each other. This can be done for instance by restricting their movements within Voronoi cells as in Algorithm 5.

- Breaking the symmetry: Since all robots execute the same deterministic algorithm, there exists some symmetric configurations where we need to break the symmetry between the robots.

Note that most of algorithms for circle formation that were proposed so far in the literature can not solve the gathering of robots with volume on a minimal circle for many reasons. First, most of the algorithms rely on the computation of the smallest circle enclosing all the robots, which is unique. However, when considering robots with volume, these robots can not compute the same smallest enclosing circle because some robots may hide the vision of others, and thus they can see only some of the robots in the system. Second, most of algorithms are designed in such a way to form a circle with an arbitrary radius.

We have thought of a partial solution for solving the weak gathering of robots with volume on a minimal ring, where the number of robots is not known in the system. This solution works for *non oblivious* robots. Initially, robots need to spread on a bigger portion of the plane in order to have a full visibility of each other. For avoiding collisions, we use Voronoi diagrams to restrict the movements of robots. For detecting the full visibility, we suppose that when all robots see the same configuration twice, they all become visible. When all robots become aware of each other, they start to form the smallest enclosing circle of the current configuration, which may have a radius bigger than the gathering circle. By using a simple algorithm, which based on computing the mid point, robots can spread uniformly on the circumference of the current circle. Finally, robots need to move on the radius in order to form the target circle with the minimal radius. However, we need to find a way to synchronize between the current circle where robots are located, and the target circle in order to maintain the origin of the smallest enclosing circle, and thus the origin of the target circle.

Concerning the solvability of the weak gathering with *oblivious* robots, we conjecture that there exists no algorithm for solving the weak gathering of robots with volume when

the number of robots is not known in the system. This is mainly due to the fact that robots are unable to compute the same origin of the target circle. However, the existence of a solution in the oblivious setting when the number of robots is known in the system mainly depends if we can break symmetric configurations

Chapter 12

Conclusion

What we call the beginning is often the end.
And to make an end is to make a beginning.
The end is where we start from.
Thomas Stearns Eliot

We first summarize our contributions and discuss the results of the thesis in Section 12.1. Then, in Section 12.2, we present our plans for future work.

12.1 Research Assessment

In this dissertation, we studied the problem of coordination of a group of mobile robots in a totally distributed fashion, and from an algorithmic standpoint. The robots are represented as points that evolve in an environment devoid of any landmarks or common coordinate system, they are equipped with sensors, and are able to move in the two dimensional plane. In addition, robots are identical (i.e., indistinguishable by the algorithm), they do not have any direct communication between them, and robots do not retain any information between activations (i.e., oblivious). The robots, executing their own instance of the same algorithm, must cooperate to accomplish some given task within finite time.

In particular, we studied the impact of sensor errors and instabilities in solving the gathering of multiple mobile robots at the exact same location, not determined a priori.

More specifically, we defined a model in which compass sensors are unreliable, and we studied the solvability of gathering with compasses that are unstable for some arbitrary long periods, provided that they stabilize eventually (eventually consistent compasses), and with compasses that are subject to inaccuracies (inaccurate compasses).

This research has led to four major contributions.

First, we have studied the solvability of the gathering problem deterministically in the face of eventually consistent compasses in oblivious and limited visibility settings, assuming the semi-synchronous model SYm, and we provided a solution to the problem. The proposed solution guarantees that the robots gather at a single point in finite time, if

their compasses provide correct output after some unknown period of instability, during which our algorithm can tolerate any number of transient failures of the compasses.

Second, we have investigated the solvability of the gathering problem in the asynchronous model CORDA, relying on eventually consistent compasses when robots are oblivious, and they have limited visibility. We have shown that our gathering algorithm proposed for the SYm model, with eventually consistent compasses, solves the problem in the CORDA model, in finite time, for up to three robots. Furthermore, we have presented a deterministic solution to the gathering problem in the CORDA model for a maximum of four robots. Thus, we can argue that *eventually consistent compass* have the same computational power as *perfect compasses* for solving the gathering problem of at most four robots.

Then, we have shown that the gathering has no deterministic solution for nine or more robots in the asynchronous model CORDA with eventually consistent compasses. This means that there is an inherent trade-off between the degree of synchrony of the system and the reliability of sensors.

Third, we have studied the solvability of the gathering of two asynchronous mobile robots in the face of compass inaccuracies, and we have shown that the problem could be solved, in a finite number of steps, provided that the divergence between the compasses is at most 45°. Besides, we have proved that our algorithm is also correct if we consider robots with volume. This result is important in practice since compasses are inaccurate devises by engineering.

Fourth, we have extended previous work by proving a tight bound on the degree of divergence of robots' compasses in solving the gathering of two robots. In particular, we have presented a self-stabilizing algorithm that gathers two asynchronous robots when their compasses can differ by any angle less than 180°, which is obviously the largest divergence for which the compasses can still provide any useful information.

Finally, we have complemented a prior work by developing complete and rigorous proofs for a circle formation algorithm. The algorithm allows a group of mobile robots, sharing no common coordinate system, to self-organize into forming a circle when starting from any configuration in the SYm model.

The proposed algorithm ensures that robots deterministically form a circle in a finite number of steps, and converges to a situation in which all robots are located evenly on the boundary of the circle. In addition, the proposed algorithm has the useful property that it allows robots to be added, removed, or relocated during its execution. A circle is guaranteed to be reformed and remain stable after external changes have come to an end. Moreover, it is intrinsically self-stabilizing, due to the assumption that robots are oblivious.

12.2 Open Questions and Future Directions

Besides the contributions presented in the previous section, this work has raised several issues that deserve further investigations. In the following, we describe some open questions related to this research and some future directions.

- **Gathering with eventually bounded error compass**. In Chapter 4, we have showed that gathering can be solved in the SYm [69] model, in finite time, with compasses that are unstable for some arbitrary long periods with the guarantee that they eventually become perfect. An other interesting question to investigate is can we still solve the gathering problem when compasses fluctuate for some arbitrary periods, and then stabilize to certain bounded errors?

- **Solvability of gathering with non oblivious robots and EVC compasses**.

  In Chapter 6, we have shown that the gathering problem has no solution in the Corda model for nine or more robots when they are equipped with compasses that are unstable for some arbitrary periods, that are oblivious and that have limited visibility. It would be interesting to determine the strict number of robots from which the problem is unsolvable.

  The impossibility of the problem comes from the fact that any possible algorithm fails to keep the distance graph connected during its execution (because of the oblivious feature of the robots, and the inconsistencies of the compasses), which is a necessary condition for solving the gathering. Alternatively, it would be beneficial to equip the robots with a bounded memory (non-oblivious), and see if this can be useful in solving the gathering with unstable compasses, and how it affects the self-stabilizing feature of oblivious algorithms.

- **Solvability of gathering with many robots and inaccurate compasses.** In Chapter 7 and Chapter 8, we have studied the solvability of the gathering of two asynchronous mobile robots with inaccurate compasses. However, the natural problem of generalizing our algorithm to an arbitrary finite number of robots remains open. We conjecture that a small bound on the degree of divergence of the compasses is required, and that this bound varies between the SYm and Corda models.

  Another interesting issue to investigate is to consider the variance in the north directions indicated by compasses over time, and how it affects the solvability of the gathering problem. To the best of our knowledge, only the work of Katayama et al. [41] addressed this issue. However, many interesting questions remain open, for instance, we know that gathering is solvable when robots' compasses diverge by up to an angle strictly less than π in the fixed compass model, then it would be good to see if the same bound holds in the dynamic compass models, and if the different classes of dynamic compass can be transformed to the fixed compass class.

- **Circle formation with limited visibility**. In Chapter 10, we have shown that the circle formation problem is solvable with oblivious robots in the SYm model, provided that robots have an unlimited range of vision, by proposing a self-stabilizing algorithm for the problem.

  It is also interesting to see whether the problem can still be solved deterministically with limited visibility or inaccurate sensors. Indeed, the proposed algorithm must rely on unlimited (or "sufficiently wide") visibility in order to compute the smallest enclosing circle. With limited visibility, it is no longer possible for the robots to compute this circle. This actually raises the question of the existence of a deterministic solution in that model.

  Finally, it would be beneficial to see whether replacing vision with other communication models (e.g., ad hoc networking with directional antennas) still allows for solving the circle formation problem.

- **Computer simulation**. In this dissertation, we analyzed the complexity of convergence of our algorithms. Among our future work, we would like also to implement these algorithms and quantify their convergence using computer simulation.

Among our future research directions, we aim at addressing the following issues:

- **Robots with volume.** Our work can still be extended to deal with real world applications. In particular, the models we considered so far are not realistic in one aspect, which is the representation of robots by points. In reality, robots have some volume, which makes problems more difficult to solve. For instance, some robots may obstruct visibility of others, and some robots may prevent the motion of others, by interfering in their trajectories. Therefore, it would be interesting to investigate the impact of considering robots with dimensions on the overall correctness of algorithms on autonomous mobile robots proposed so far in the literature.

  To the best of our knowledge, only the work of Czyzowicz et al. [20] has addressed this issue by considering robots as unit discs in the plane, and they proposed an algorithm for gathering up to four robots in the CORDA model.

- **Fault tolerance in multi-robot systems.** The aspect of resilience to failure in multiple robot systems has been studied only very superficially. In fact, most of the results we are aware of rely on the assumption that the robots function properly, and behave correctly. Among our future work, we aim at developing robust distributed algorithms for reaching agreement among a set of autonomous mobile robots when they exhibit faulty behavior. Concretely, some robots may crash or behave maliciously. Our algorithms must be designed so as to deal properly with such failures.

We also aim at defining an appropriate failure model in which we can study agreement problems for cooperative mobile robotics. We thus aim to investigate the minimal conditions under which robots can cooperate in the presence of failure, and develop resilient distributed algorithms for these problems. The integration of these algorithms in a single framework will provide the necessary basis for decentralized control of robots and smooth the progress toward real world applications. Our research aims at emphasizing the guaranteed reliability of multiple robot systems, by favoring fully decentralized solutions, developing fault-tolerant mechanisms, and using an algorithmic approach to allow for formal verification and correctness.

- **Explicit communication between robots.** Another significant issue that we want to investigate is the issue of asynchronous, geographically free communication between mobile robots (for instance ad hoc communication), and its role in solving cooperation problems in the presence of failure, and in developing more robust solutions. More precisely, the robots may need to exchange information on their states (positions, trajectories, orientation, etc.) to construct a complete configuration of the team in order to cooperate. Many problems are impossible to solve when robots rely on vision only, for instance robots may need communication to synchronize. We believe that asynchronous communication can play a crucial role in their ability to cooperate in the face of failures, since robots may not initially agree on a common coordinate system.

  Finally, we intend to use these problems as a starting point for studying the roles and strengths of different communication models in mobile robotics research in order to progress toward a more precise science.

Bibliography

[1] N. Agmon and D. Peleg. Fault-tolerant gathering algorithms for autonomous mobile robots. *SIAM J. Comput.*, 36(1):56–82, 2006.

[2] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita. Distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Trans. on Robotics and Automation*, 15(5):818–828, 1999.

[3] H. Ando, I. Suzuki, and M. Yamashita. Formation and agreement problems for synchronous mobile robots with limited visibility. In *IEEE International Symposium on Intelligent Control*, pages 453–460, 1995.

[4] F. Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, September 1991.

[5] T. Balch and R. C. Arkin. Behavior-based formation control for multi-robot teams. *IEEE Trans. on Robotics and Automation*, 14(6):926–939, 1998.

[6] B. R. Bellur, M. G. Lewis, and F. L. Templin. An ad-hoc network for teams of autonomous vehicles. *In Proc. of IEEE Symposium on Autonomous Intelligence Networks and Systems*, 2002.

[7] G. Beni and S. Hackwood. Coherent swarm motion under distributed control. In *Proc. International Symposium on Distributed Autonomous Robotic Systems (DARS'92)*, pages 39–52, 1992.

[8] G. Beni and J. Wang. Swarm intelligence. In *Proc. of the Seventh Annual Meeting of the Robotics Society of Japan*, pages 425–428, 1989.

[9] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1999.

[10] R. A. Brooks. New approaches to robotics. *Science*, 253(5025):1227–1232, 1991.

[11] W. Burgard, M. Moors, and F. Schneider. Collaborative exploration of unknown environments with teams of mobile robots. In *Springer Verlag*, 2002.

[12] S. Camazine, J.-L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau. *Self-Organization in Biological Systems*. 2001.

[13] I. Chatzigiannakis, M. Markou, and S. Nikoletseas. Distributed circle formation for anonymous oblivious robots. In C. C. Ribeiro and S. L. Martins, editors, *Proc. 3rd Workshop on Efficient and Experimental Algorithms (WEA-3)*, volume 3059 of *Lecture Notes Comput. Sci.*, pages 159–174, Angra dos Reis, Rio de Janeiro, Brazil, May 2004. Springer-Verlag.

[14] I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis. On the average and worst-case efficiency of some new distributed communication and control algorithms for ad-hoc mobile networks. In *Proc. 1st ACM Int'l Workshop on Principles of Mobile Computing (POMC'01)*, pages 1–19, Newport, RI, USA, August 2001.

[15] Q. Chen and J.Y. S. Luh. Coordination and control of a group of small mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 2315–2320, 1994.

[16] M. Cieliebak. Gathering non-oblivious mobile robots. In *Proc. 6th Latin American Symp. on Theoretical Informatics (LATIN'04)*, pages 577–588, 2004.

[17] M. Cieliebak, P. Flocchini, G. Prencipe, and N. Santoro. Solving the robots gathering problem. In *Proc. Intl. Colloquium on Automata, Languages and Programming (ICALP'03)*, pages 1181–1196, 2003.

[18] R. Cohen and D. Peleg. Robot convergence via center of gravity algorithms. In *Proc. Colloquium on Structural Information and Communication Complexity (SIROCCO'04)*, volume 3104, pages 79–88, 2004.

[19] R. Cohen and D. Peleg. Convergence of autonomous mobile robots with inaccurate sensors and movements. In *Proc. 23rd International Symposium on Theoretical Aspects of Computer Science (STACS'06)*, number LNCS 3884, pages 549–560, 2006.

[20] J. Czyzowicz, L. Gasieniec, and A. Pelc. Gathering few fat mobile robots in the plane. In *Proc. 10th International Conference On Principles Of Distributed Systems (OPODIS'06)*, volume 4305, pages 350–364, 2006.

[21] X. A. Debest. Remark about self-stabilizing systems. *Communications of the ACM*, 38(2):115–117, February 1995.

[22] X. Défago, M. Gradinariu, S. Messika, and P. Raipin-Parvédy. Fault-tolerant and self-stabilizing mobile robots gathering. In *Proc. 20th Intl. Symp. on Distributed Computing (DISC'06)*, volume 4167, pages 46–60, 2006.

[23] X. Défago and A. Konagaya. Circle formation for oblivious anonymous mobile robots with no common sense of orientation.

[24] M. B. Dias and A. Stentz. A free market architecture for distributed control of a multirobot system. In *Proc. 6th Intl. Conf. on Intelligent Autonomous Systems (IAS-6)*, pages 115–122, Venice, Italy, July 2000.

[25] Y. Dieudonne, O. Labbani-Igbida, and F. Petit. Circle formation of weak mobile robots. In *Proc. of 8th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS'06)*, volume 4280 of *Lecture Notes Comput. Sci.*, pages 262–275. Springer-Verlag, 2006.

[26] Y. Dieudonne and F. Petit. Circle formation of weak robots and lyndon words. *Information Processing Letters*, 104(4):156–162, 2007.

[27] S. Dolev. *Self-Stabilization*. MIT Press, 2000.

[28] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. A taxonomy for swarm robots. In *Proc. of Intelligent Robots and Systems (IROS'93)*, pages 441–447, Yokohama, Japan, 1993.

[29] A. Dumitrescu, I. Suzuki, and M. Yamashita. Motion planning for metamorphic systems: Feasibility, decidability and distributed reconfiguration. *IEEE Transactions on Robotics and Automation*, 20(3):409–418, 2004.

[30] M. Fisher, N. Lynch, and M. S. Paterson. Impossibility of distributed consensus with one faulty process. *J. ACM.*, 32(2):374–382, 1985.

[31] P. Flocchini, G. Prencipe, and N. Santoro. Self-deployment algorithms for mobile sensors on a ring. In *Proc. 2nd International Workshop on Algorithmic Aspects of Wireless Sensor Networks (Algosensors'06)*, pages 59–70, 2006.

[32] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Hard tasks for weak robots: The role of common knowledge in pattern formation by autonomous mobile robots. In *Proc. 10th Int'l Symp. on Algorithms and Computation (ISAAC'99)*, volume 1741 of *LNCS*, pages 93–102, 1999.

[33] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Pattern formation by autonomous robots without chirality. In *Proc. 8th Intl. Colloquium on Structural Information and Communication Complexity (SIROCCO 2001)*, pages 147–162, Vall de Núnia, Spain, jun 2001.

[34] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Gathering of asynchronous robots with limited visibility. *Theor. Comput. Sci.*, 337(1–3):147–168, 2005.

[35] V. Gervasi and G. Prencipe. Flocking by a set of autonomous mobile robots. Technical Report TR-01-24, Dipartimento di Informatica, Università di Pisa, Italy, October 2001.

[36] V. Gervasi and G. Prencipe. Need a fleet? use the force! In *Fun With Algorithms 2 (FUN 2001)*, pages 149–164, Elba, Italy, 2001.

[37] V. Gervasi and G. Prencipe. Coordination without communication: The case of the flocking problem. *Discrete Appl. Math.*, 144(3):324–344, 2004.

[38] H. Imazu, N. Itoh, Y. Katayama, N. Inuzuka, and K. Wada. A gathering problem for autonomous mobile robots with disagreement in compasses (in japanese). In *In: 1st Workshop on Theoretical Computer Science*, pages 43–46, Izumo, Japan, 2005.

[39] E. Izquierdo-Torres. Collective intelligence in multi-agent robotics: Stigmergy, self-organization and evolution. In *University of Sussex*, 2004.

[40] J.S. Jennings, G. Whelan, and W.F. Evans. Cooperative search and rescue with a team of mobile robots. In *Proc. 8th International Conference on Advanced Robotics*, pages 193–200, 1997.

[41] Y. Katayama, Y. Tomida, H. Imazu, N. Inuzuka, and K. Wada. Dynamic compass models and gathering algorithms for autonomous mobile robots. In *Proc. 14th Colloquium on Structural Information and Communication Complexity (SIROCCO '07)*, 2007.

[42] B. Katreniak. Biangular circle formation by asynchronous mobile robots. In A. Pelc and M. Raynal, editors, *Proc. 12th Intl. Colloquium on Structural Information and Communication Complexity (SIROCCO 2005)*, volume 3499, Mont St-Michel, France, May 2005. Springer-Verlag.

[43] Y. Kawauchi, M. Inaba, and T. Fukuda. A principle of distributed decision making of cellular robotic system (cebot). In *Proc. IEEE International Conference on Robotics and Automation*, volume 3, pages 833–838, 1993.

[44] K. Konolige, C. Ortiz, R. Vincent, A. Agno, M. Eriksen, B. Limketkai, M. Lewis, L. Briesemeister, E. Ruspini, D. Fox, J. Ko, B. Stewart, and L. Guibas. Centibots: Large-scale robot teams. *In Multi-Robot Systems: From Swarms to Intelligent Autonoma*, 2003.

[45] M. J. B. Krieger, J.-B. Billeter, and L. Keller. Ant-like task allocation and recruitment in cooperative robots. *Nature*, 406(6799):992–995, August 2000.

[46] L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.

[47] A. Martinoli and F. Mondada. Collective and cooperative group behaviours: Biologically inspired experiments in robotics. In *Proc. 4th Symposium on Experimental Robotics*, pages 3–10, 1995.

[48] M. J. Matarić. Minimizing complexity in controlling a mobile robot population. In *Proc. IEEE International Conference on Robotics and Automation*, pages 830–835, Nice, France, 1992.

[49] M. J. Matarić. Designing emergent behaviors: From local interactions to collective intelligence. In *Proc. Intl. Conf. on Simulation of Adaptive Behavior*, pages 423–441, 1993.

[50] M. J. Matarić. From local interactions to collective intelligence. In *Proc. The Biology and Technology of Intelligent Autonomous Agents*, volume 144, pages 275–295, 1995.

[51] U. Nehmzow. Mobile robotics: Research, applications and challenges. In *Proc. Future Trends in Robotics*, 2001.

[52] E. Nett and S. Schemmer. Reliable real-time communication in cooperative mobile applications. *IEEE Trans. Computers.*, 52(2):166–180, 2003.

[53] L. Parker. Alliance: An architecture for fault-tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, 1998.

[54] D. Peleg. Distributed coordination algorithms for mobile robot swarms: New directions and challenges. In *Proc. 7th International Workshop on Distributed Computing (IWDC'05)*, pages 1–12, 2005.

[55] G. Prencipe. Instantaneous actions vs. full asynchronicity: Controlling and coordinating a set of autonomous mobile robots. In *Proc. 7th Italian Conf. on Theoretical Computer Science (ICTCS'01)*, pages 154–171, Torino, Italy, oct 2001.

[56] G. Prencipe. CORDA: Distributed coordination of a set of autonomous mobile robots. In *Proc. 4th European Research Seminar on Advances in Distributed SystemsDependable Systems (ERSADS'01)*, pages 185–190, 2001.

[57] G. Prencipe. *Distributed Coordination of a Set of Autonomous Mobile Robots*. PhD thesis, Universitá di Pisa, 2002.

[58] G. Prencipe. The effect of synchronicity on the behavior of autonomous mobile robots. In *Proc. Theory of Computing Systems*, volume 38, pages 539–558, 2005.

[59] G. Prencipe. On the feasibility of gathering by autonomous mobile robots. In *Proc. Colloquium on Structural Information and Communication Complexity (SIROCCO'05)*, pages 246–261, 2005.

[60] R. O. Saber and R. M. Murray. Flocking with obstacle avoidance: Cooperation with limited communication in mobile networks. In *Proc. 42nd Conference on Decision and Control*, 2003.

[61] P. Schermerhorn and M. Scheutz. Social coordination without communication in multi-agent territory exploration tasks. In *Proc. 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 654–661, Hakodate, Japan, 2006.

[62] M. Schneider. Self-stabilization. *ACM Computing Surveys*, 25(1):45–67, 1993.

[63] K. Schreiner. NASA's JPL nanorover outposts project develops colony of solar-powered nanorovers. *In IEEE DS Online*, 2(3), 2001.

[64] S. Souissi. On distributed cooperative mobile robotics: Decomposition of basic problems and study of a self-stabilizing circle formation algorithm. Master's thesis, Japan Advanced Institute of Science and Technology, School of Information Science, September 2004.

[65] S. Souissi, X. Défago, and M. Yamashita. Gathering asynchronous mobile robots with inaccurate compasses. In *Proc. 10th International Conference On Principles Of Distributed Systems (OPODIS'06)*, volume 4305, pages 333–349, 2006.

[66] S. Souissi, X. Défago, and M. Yamashita. Using eventually consistent compasses to gather oblivious mobile robots with limited visibility. In *Proc. 8th Intl. Symp. on Stabilization, Safety, and Security of Distributed Systems (SSS'06)*, volume 4280, pages 471–487, 2006.

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

[68] I. Suzuki and M. Yamashita. Agreement on a common x-y coordinate system by a group of mobile robots. *Intelligent Robots: Sensing, Modeling and Planning, World Scientific*, pages 305–321, 1997.

[69] I. Suzuki and M. Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM Journal of Computing*, 28(4):1347–1363, 1999.

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

[71] Devantech compass. Acroname Robotics. http://www.acroname.com/robotics/parts/R117-COMPASS.html.

[72] 'smart dust' to explore planets. news.bbc.co.uk. http://news.bbc.co.uk/2/hi/science/nature/6566317.stm.

[73] Hitachi hm55b compass. www.Parallax.com. http://www.parallax.com.

[74] Rdcm-802 digital compass model. www.rakuten.co.jp. http://www.rakuten.co.jp/tsukumo/487433/487437/622043/.

[75] The hmr3600 digital magnetic compass. Solid State Electronics Center. http://www.ssec.honeywell.com/magnetic/products.html.

[76] Definition of compass. wikipedia.org. http://en.wikipedia.org/wiki/Compass.

[77] Lyndon words. wikipedia.org. http://en.wikipedia.org/wiki/Lyndon_word.

[78] J. E. Walter, J.L. Welch, and N. M. Amato. Distributed reconfiguration of metamorphic robot chains. *Distrib. Comput.*, 17(2):171–189, 2004.

[79] E. W. Weisstein. Reuleaux triangle. From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/ReuleauxTriangle.html.

[80] E. Welzl. Smallest enclosing disks (balls and ellipsoids). In H. Maurer, editor, *New Results and New Trends in Computer Science*, volume 555 of *Lecture Notes Comput. Sci.*, pages 359–370. Springer-Verlag, 1991.

[81] R. Yared, J. Cartigny, X. Défago, and M. Wiesmann. Locality-preserving distributed path reservation protocol for asynchronous cooperative mobile robots. In *Proc. Intl. Symp. on Autonomous Decentralized Systems (ISADS'2007)*, Sedona, AZ, USA, 2007.

[82] R. Yared, X. Défago, and M. Wiesmann. Collision prevention using group communication for asynchronous cooperative mobile robots. In *Proc. The IEEE 21st International Conference on Advanced Information Networking and Applications (AINA'07), to appear*, 2007.

[83] D. Yoshida, T. Masuzawa, and H. Fujiwara. Fault-tolerant distributed algorithms for autonomous mobile robots with crash faults. In *Proc. of Systems and Computers in Japan*, number 28, pages 33–43, 1997.

Chapter 13

Publications

International Journals:

[1] S. Souissi, X. Défago and M. Yamashita: "Using Eventually Consistent Compasses to Gather Memory-less Mobile Robots with Limited Visibility". *Submitted to ACM Transactions on Autonomous and Adaptive Systems (TAAS)*

[2] X. Défago and S. Souissi: "Self-Stabilizing Algorithm for Circle Formation by Memory-less Mobile Robots". *Submitted to Journal of Theoretical Computer Science.*

International Conferences:

[3] S. Souissi, X. Défago and M.Yamashita: "Gathering Asynchronous Mobile Robots with Inaccurate Compasses". *Proc. 10th International Conference On Principles Of Distributed Systems (OPODIS '06).* LNCS. 4305, pp. 333–349. (Dec. 2006).

[4] S. Souissi, X. Défago and M. Yamashita: "Using Eventually Consistent Compasses to Gather Oblivious Mobile Robots with Limited Visibility". *Proc. 8th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS '06)*, LNCS. 4280, pp. 471–487 (Nov. 2006).

[5] S. Souissi, X. Défago and T. Katayama: "Decomposition of Fundamental Problems for Cooperative Autonomous Mobile Systems". *Proc. of 2nd International Workshop on Mobile Distributed Computing, (ICDCSW'04).* IEEE. Computer Society Press. pp. 554-560 (Mar. 2004).

[6] M. Yamashita, S. Souissi and X. Défago: "Gathering Two Stateless Mobile Robots with Very Inaccurate Compasses in Finite Time". *First International Conference on Robot Communication and Coordination (ROBOCOMM '07), Sigmobile. To appear.*

[7] S. Souissi, X. Défago and M. Yamashita: "Tradeoff between Asynchrony and Unreliability of Compasses". *Submitted to the 18th International Symposium on Algorithms and Computation (ISAAC '07).* Sendai, Japan.

Domestic conferences/ Workshops/ Posters:

[8] S. Souissi, X. Défago and M. Yamashita: "Solvability of the Gathering Problem with Oblivious Robots and Inconsistent Compasses". *Proc. Scientific French-speaking Workshops (JSF'06)*, (Dec. 2006). Tokyo, Japan.

[9] S. Souissi, X. Défago and M. Yamashita: "Unreliable Compasses for Robust Gathering of Asynchronous Mobile Robots". *Proc. Scientific French-speaking Workshops (JSF'05)*, (Nov. 2005). Tokyo, Japan.

[10] S. Souissi, X. Défago and T. Katayama: "Convergence of a Self-Stabilizing Circle Formation Algorithm for Cooperative Mobile Robotics". *Proc. Scientific French-speaking Workshops (JSF'04)*, (Nov. 2004). Tokyo, Japan.

[11] S. Souissi, X. Défago and T. Katayama: "Convergence of a Uniform Circle Formation Algorithm for Distributed Autonomous Mobile Robots". *Proc. Joint Japan-Tunisia Workshop on Computer Systems and Information Technology (JT-CSIT'04)*, (Jul. 2004). Tokyo, Japan.

[12] S. Souissi, X. Défago and T. Katayama: "Specification of Recurrent Problems in Distributed Cooperative Mobile Robotics". *Proc. Scientific French-speaking Workshops (JSF'03)*, (Nov. 2003). Tokyo, Japan.

Research Reports:

[13] S. Souissi, X. Défago and M. Yamashita: "Tight Bound on the Gathering of Oblivious Mobile Robots with Inconsistent Compasses". *Research Report*, JAIST, March 2007, Ishikawa, Japan.