Implicit Bit Transmission Among Stigmergic Robots *†

Yoann Dieudonné
MIS, Université of Picardie Jules Verne, France

Shlomi Dolev, *Senior Member, IEEE*
Department of Computer Science, Ben-Gurion University of the Negev, Israel

Franck Petit
LiP6/CNRS/INRIA-REGAL, Université Pierre et Marie Curie - Paris 6, France

Michael Segal, *Senior Member, IEEE*
Communication Systems Engineering Dept, Ben-Gurion University of the Negev, Israel

November 30, 2013

Abstract

In this paper we investigate avenues for the exchange of information (explicit communication) among deaf and dumb mobile robots scattered in the plane. We introduce the use of movement-signals (analogously to flight signals and bees waggle) as a mean to transfer messages, enabling the use of distributed algorithms among robots. We propose one-to-one deterministic movement protocols that implement explicit communication among semi-synchronous robots. We first show how the movements of robots can provide implicit acknowledgment in semi-synchronous systems. We use this result to design one-to-one communication among a pair of robots. Then, we propose two one-to-one communication protocols for any system of $n \geq 2$ robots. The former works for robots equipped with observable IDs that agree on a common direction (sense of direction). The latter enables one-to-one communication assuming robots devoid of any observable IDs or sense of direction. All protocols (for either two or any number of robots) assume that no robot remains inactive forever. However, they cannot avoid that the robots move either away or closer to each others, by the way requiring robots with an infinite visibility. In this paper, we also present how to overcome these two disadvantages (some activity of every robot and infinite visibility).

Our protocols enable the use of distributing algorithms based on message exchanges among swarms of stigmergic robots. They also allow robots to be equipped with the means of communication to tolerate faults in their communication devices.

Index terms: Explicit Communication, Mobile Robot Networks, Stigmergy.

1 Introduction

Although research in achieving coordination among teams (or swarms) of mobile robots is challenging, it has great scientific and practical implications. Swarms of mobile robots are currently being utilized and are expected to be employed even more in the future, in various critical situations. Swarms foster the ability to measure properties, collect information, and act in any given (sometimes dangerous) physical environment. Numerous potential applications

*A preliminary version of this work appeared in [11].

†The work of Shlomi Dolev and Michael Segal has been partially supported by US Air Force grant. The work of Yoann Dieudonné and Franck Petit has been partially supported by ANR Project *R-Discover*.

exist for such multi-robot systems: environmental monitoring, large-scale construction, risky area surrounding or surveillance, and the exploration of awkward environments, to name only a few.

In any given environment, the ability of a swarm of robots to succeed in accomplishing the assigned task depends greatly on the capabilities that the robots possess, that is, their moving capacities and sensory organs. Existing technologies allow the consideration of robots equipped with sensory devices for vision (camera, radar, sonar, laser, etc.) and means of communication (wireless devices). Furthermore, means of communication are required to enable classical distributed algorithms and to achieve the completion of several tasks, such as information relay, surveillance, or intelligence activities.

An interesting question is "*What happens if the means of communication are broken or do not exist?*" In this case, the robots can observe the location of other robots but cannot communicate with them. Such robots are called *deaf and dumb*. There are numerous realistic scenarios where there are no means for communication among robots. Such scenarios are easily deciphered, *e.g.*,

- Wireless devices are faulty,
- Robots evolve in zones with blocked wireless communication (hostile environments where communication is scrambled or forbidden), or
- Physical constraints prevent placing wireless devices on robots.

The last case may arise when no physical space is available on the robots or the robots themselves are too small with respect to the size of the wireless device. Such is the case with swarms of nano-robots.

The question of solving distributed tasks with swarms of deaf and dumb robots is not a novel one. This question has been extensively posed in different fields of computer science such as artificial intelligence [20], control theory [17, 22, 7], and recently in the distributed computing field [26, 23]. Some of these approaches are inspired by biological studies of animal behavior, mainly the behavior of social insects [3]. Indeed, these social systems present an intelligent collective behavior, despite being composed of simple individuals with extremely limited capabilities. Solutions to problems "naturally" emerge from the self-organization and indirect communication of these individuals. The capacity to communicate using such *indirect communication* (or, *implicit* communication) is referred to as *stigmergy* in the biological literature [19]. There are numerous examples of such indirect communication in nature, for instance ants and termites communicating using pheromones, or bees communicating by performing waggle dances to find the shortest paths between their nests and food sources. The question of whether the waggle of bees is a language or not is even an issue [29]. Bee waggle dances has been an inspiration source in recent researches in various areas related to the distributed systems, *e.g., swarm intelligence* [13] and *communication technologies* [28, 27].

However, stigmergy leads to the completion of only one given task at a time. Communication is not considered as a task in itself. In other words, the stigmergic phenomenon provides indirect communication; guidance for a specific assignment. Even if stigmergy sometimes allows insects to modify their physical environment, —this phenomenon is sometime referred to as *sematectonic stigmergy* [14]— stigmergy never provides a communication task alone. Stigmergy does not allow tasks such as chatting, intelligence activities, or the sending of information unrelated to a specific task.

In this paper, we investigate avenues for the exchange of information among deaf and dumb mobile robots scattered in the plane. In the sequel, we refer to this task as *explicit communication*—sometimes, also referred to as *direct* communication [20]. Explicit communication enables the use of distributed algorithms among robots. We study the possibility of solving this problem deterministically so that robots communicate only by moving.

Our Contribution. We introduce the use of movement-signals (analogously to flight signals and bee waggles) as a mean to transfer messages between deaf and dumb robots. We propose one-to-one deterministic protocols that implement explicit communication among semi-synchronous robots.

In semi-synchronous settings, each computation step may contain some robots that remain inactive. So, a number of robot movements can go unnoticed and, thus, some messages can be lost as well. As a consequence, each message is required to be acknowledged by the addressees. We first demonstrate how robot movements can provide implicit acknowledgment in semi-synchronous settings. We straightforwardly use this result in the design of semi-synchronous one-to-one communication among a pair of robots.

Next, we propose two one-to-one communication protocols that fit the general case of semi-synchronous systems of $n \geq 2$ robots. The former protocol assumes robots equipped with observable IDs that agree on a common direction (sense of direction). The latter is a routing protocol enabling one-to-one communication among robots that agree on a common handedness (*chirality*) only, *i.e.,* they are *anonymous* and *disoriented*—having no knowledge of any observable IDs and having no sense of direction. Since one-to-one communication must include a technique allowing any robot to send messages to a specific robot, our protocol builds a naming system based on the positions of the robots that allows them to address one another.

All our protocols, either for two or n robots, are presented in the *semi-synchronous model* [26] (see also [30]), imposing a certain amount of synchrony among the active robots, *i.e.,* at each time instant, the robots which are activated, observe, compute, and move in one atomic step. However, no other assumption is made on the relative frequency of robot activations with respect to each other, except that each robot is activated infinitely often (uniform fair activation). This lack of synchronization among the robots does not prevent the robots either to move away from or to get closer to each other infinitely often. As a consequence, the robots are required to have an *infinite visibility*. Visibility capability of the robots is an important issue [1, 16]. In this paper, we also show how to overcome these drawbacks by introducing a relaxed form of synchrony. By relaxed, we mean that the robots are not required to be strictly synchronous. A bound $k \geq 1$ is assumed on the maximum activation drift among the robots, *i.e.,* no robot can be activated more than k times between two consecutive activations of any other robot.

Note that our protocols can be easily adapted to efficient implementation of one-to-many or one-to-all explicit communication. Also, in the context of robots (explicitly) interacting by means of communication (*e.g.,* wireless) our solution can serve as a communication backup, *i.e.*, it provides *fault-tolerance* by allowing the robots to communicate without means of communication (wireless devices), since our protocols allow robots to explicitly communicate even if their communication devices are faulty.

Related Work. The issue of handling swarms of robots using deterministic distributed algorithms was first studied in [26]. Beyond supplying formal correctness proofs, the main motivation is to understand the relationship between the capabilities of robots and the solvability of given tasks. For instance, "*Assuming that the robots agree on a common direction (having a compass), which tasks are they able to deterministically achieve*?", or "*What are the minimal conditions to elect a leader deterministically*?" As a matter of fact, the motivation turns out to be the study of the minimum level of ability that the robots are required to have to accomplish basic cooperative tasks in a deterministic way. Examples of such tasks are specific problems, such as *pattern formation*, *line formation*, *gathering*, *spreading*, and *circle formation*—refer for instance to [26, 15, 8, 23, 5, 16, 24, 9, 6] for these problems,— or more "classical" problems in the field of distributed systems, such as *leader election* [23, 15, 10]. To the best of our knowledge, the only work which also addresses the problem of enabling explicit communication in swarms of robots is [4] but it is based on our original results, where we first define and show the way to transmit bits by movement, and therefore execute distributed algorithms by robots with implicit communication.

Paper Organization. In the next section (Section 2), we describe the model and the problem considered in this paper. In Section 3, we show how robot movements can provide implicit acknowledgments in semi-synchronous settings. Here we also propose a straightforward communication protocol for two semi-synchronous robots. Section 4 is devoted to one-to-one communication for any number of robots. The motion containment is discussed in Section 5. Finally, we make some concluding remarks at Section 6.

2 Preliminaries.

In this section, we first define the distributed system considered in this paper. We then state the problem to be solved.

Model. We adopt the model introduced in [26], below referred to as *Semi-Synchronous Model (SSM)*. The distributed system considered in this paper consists of n mobile *robots* (*agents* or *sensors*). Any robot can observe, compute, and move with infinite decimal precision. Each robot r has its own local x-y Cartesian coordinate system with its own unit measure. The robots are equipped with sensors enabling them to instantaneously detect (to take a snapshot) of the position of the other robots in the plane in their own Cartesian coordinate system. Viewed as the points in the Euclidean plane, the robots are mobile and autonomous. There is no kind of explicit communication medium between robots.

Given an x-y Cartesian coordinate system, *handedness* is the way in which the orientation of the y axis (respectively, the x axis) is inferred according to the orientation of the x axis (resp., the y axis). The robots are assumed to have the ability of *chirality*, *i.e.*, the n robots share the same handedness. We consider *non-oblivious* robots, *i.e.*, every robot can remember its previous observations, computations, or motions made in any step.

We assume that the system is either *identified* or *anonymous*. In the former case, each robot r is assumed to have a visible (or observable) identifier denoted id_r such that, for every pair r, r' of distinct robots, $id_r \neq id_{r'}$. In the latter, no robot is assumed to have a visible identifier. In this paper, we will also discuss whether the robots agree on the orientation of their y-axis or not. In the former case, the robots are said to have the *sense of direction*. (Note that since the robots have the ability of chirality, when the robots have the sense of direction, they also agree on their x-axis).

Time is represented as an infinite sequence of time instants $t_0, t_1, \ldots, t_j, \ldots$ Let $P(t_j)$ be the set of the positions in the plane occupied by the n robots at time t_j ($j \geq 0$) and let $p_i(t_j)$ be the position of robot r_i in the plane at time t_j. For every t_j, $P(t_j)$ is called the *configuration* of the distributed system in t_j. $P(t_j)$ expressed in the local coordinate system of any robot r_i is called a *view* of robot r_i. At each time instance t_j ($j \geq 0$), every robot r_i is either *active* or *inactive*. The former means that, during the computation step (t_j, t_{j+1}), using a given algorithm, r_i computes in its local coordinate system a new position $p_i(t_{j+1})$ depending only on the system configuration at t_j, and moves towards $p_i(t_{j+1})$. In the latter case (inactive), r_i does not perform any local computation and remains at the same position during the computation step (t_j, t_{j+1}).

The concurrent activation of robots is modeled by the interleaving model in which the robot activations are driven by a *fair distributed scheduler*, *i.e.*, at every instant, a non-empty subset of robots can be activated (distributed scheduler), and every robot is activated infinitively often (fairness).

In every single activation, the distance traveled by any robot r is bounded by σ_r. So, if the destination point computed by r is farther than σ_r, then r moves towards a point of at most σ_r distance from its current location. The value of σ_r may differ for different robots.

Problem. *Indirect* communication is the result of the observations of other robots. Using indirect communication, we aim to implement *direct* communication that is a purely communicative act, with the sole purpose of transmitting messages [20]. In this paper, we consider direct communication that aims at a particular receiver. Such communication is said to be *one-to-one*, specified as follows: ($Emission$) If a robot r wants to send a message m to a robot r', then r eventually sends m to r'; ($Receipt$) Every robot eventually receives every message which is meant for it.

Note that the above specification induces that r is able to address r'. This implies that any protocol solving the above specification has to develop (1) *routing* mechanism and (2) *naming* mechanism, in the context of anonymous robots.

The specification also induces that the robots are able to communicate explicit messages. Hence, any one-to-one communication protocol in our model should be able (3) to code explicit messages with implicit communication, *i.e.*, with (non-ambiguous) movements.

3 Enabling Acknowledgments

Due to the lack of any communication means, the robots implicitly communicate by observing the position of the other robots in the plane, and by executing a part of their program accordingly. Thus, in our model, each symbol of every message needs to be coded by a non-ambiguous movements. In a synchronous system where every robot is active at each time instant, every movement made by any robot would be seen by all the robots of the system. So, in such settings, there would be no concern with the receipt of each symbol. Therefore, no acknowledgment would be required.

By contrast, in an semi-synchronous system, only fairness is assumed, *i.e.*, in each computation step, some robots can remain inactive. Therefore, some robot movements can go unnoticed, and as a consequence, some symbols (in fact, some messages) can be lost as well. As such, a synchronization mechanism is required, ensuring acknowledgment of each message sent.

In what follows (Subsection 3.1), we first establish general results to implement such a synchronization mechanism. Next, in Subsection 3.2, we show that these results provide a straightforward solution working with two robots.

3.1 Implicit Acknowledgment

Let us first focus on both *Emission* and *Receipt* properties. We first state the following results:

Lemma 3.1. *Let r and r' be two robots. Assume that r always moves in the same direction each time it becomes active. If r observes that the position of r' has changed twice, then r' must have observed that the position of r has changed at least once.*

Proof. By contradiction, assume that at time t_i, r notes that the position of r' has changed twice and r' has not observed that the position of r has changed at least once. Without loss of generality, we assume that t_i is the first time for which r notes that the position of r' has changed twice. So, at time t_i, r knows three distinct positions of r' and $t_i \geq 2$. Let p_j be the last (or the third) position of r' that r has observed, and t_j be the first time instance for which p_j is occupied by r'. Obviously, $t_j < t_i$. We have two cases to consider :

- **case 1 :** $t_j = t_i - 1$. The fact that r knows three distinct positions of r' implies that r became active and has moved at least twice between $t = 0$ and $t = t_i - 1$ and, thus at least once between $t = 0$ and $t = t_i - 2$. Consequently, at time $t_j = t_i - 1$, r' would have noted that r's position has changed at least once. Contradiction.

- **case 2 :** $t_j \leq t_i - 2$. We have two subcases to consider :

  – **subcase a :** r' moves at least once between $t_j + 1$ and $t_i - 1$. In this case, r notes that the position of r' has changed twice before time t_i. This contradicts the fact that t_i is the first time for which r notes that the position of r' has changed twice.

– **subcase b :** r' does not move between $t_j + 1$ and $t_i - 1$. As mentioned above, the fact that r knows three distinct positions of r' implies that r became active and has moved at least twice between $t = 0$ and $t = t_i - 1$. However, r' does not move between $t_j + 1$ and $t_i - 1$. Hence, r has moved at least twice between $t = 0$ and $t = t_j$ and, thus at least once between $t = 0$ and $t = t_j - 1$. Therefore, at time t_j, r' would have noted that r's position has changed at least once. A contradiction.

□

As a consequence of Lemma 3.1, robot r' knows the line over which r has moved, *i.e.*, the line and the direction passing through the first two distinct positions of r that r' has observed. This remark leads to the following corollary:

Corollary 3.2. *Let r and r' be two robots. Assume that r always moves in the same direction on line l as soon as it becomes active. If r observes that the position of r' has changed twice, then r' knows the line l and the direction towards which r moved.*

Note that in [26] the authors made a similar observation without proving it. They used it in the design of a protocol for the gathering problem with two non-oblivious robots. In the next subsection, we show how the above results provide a straightforward protocol for two robots, Protocol $\texttt{Async}_2$.

3.2 One-to-One Communication with Two semi-synchronous Robots

Both robots follow the same scheme. Each time a robot, say r, becomes active, it moves in the opposite direction of the other robot, r'. Let us call this direction $North_r$. Robot r behaves like this while it has nothing to send to r'. As soon as r observes that the position of r' has changed twice, by Corollary 3.2 and due to the fairness, r is guaranteed that r' knows the line H and the direction that r has moved. Let us call the line H the *horizon line*. Note that since the two robots follow the same behavior, H is common to both of them, and their respective Norths are oriented in the opposite direction.

From this moment on, r can start to send messages to r'. When r wants to send a bit "0" ("1", respectively) to r', r moves along a line perpendicular to on East side (West side, resp.) of H with respect to $North_r$. It then moves in the same direction each time it becomes active until it observes that the position of r' has changed twice. From this moment on, from Lemma 3.1, r knows that r' has seen it on its East side. Then, r comes back to H. Once r is located at H, it starts to move again towards the $North_r$ direction until it observes that r' has moved twice. This way, if Robot r wants to communicate another bit (following the same scheme), it is allowed to move on its East or West side again. Thus, the new bit and the previous bit are well distinguished by robot r' even if they have the same value. An example of our scheme (we call it $\texttt{Async}_2$) is shown in Figure 1.

Note that the lack of synchrony does not prevent the robots to move infinitely often, even having no message to send, as well as it does not allow to predict the traveled distance on each segment. This issues are addressed in Section 5. By Lemma 3.1, Protocol $\texttt{Async}_2$ ensures the *Receipt* property provided the following condition: r observed

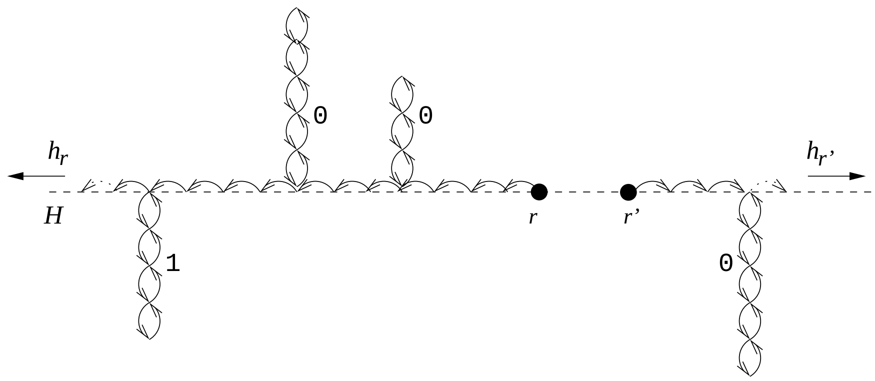


Figure 1: semi-synchronous communication for 2 robots. Robot r sends "$001\ldots$", Robot r' sends "$0\ldots$"

that the position of r' changed twice before any direction change. We now show that Protocol $\texttt{Async}_2$ ensures this condition, noticing the following Remark.

Remark 3.3. *Whenever any robot is activated it moves.*

Lemma 3.4. *Let r and r' be two robots. In every execution of Protocol* $\texttt{Async}_2$*, r observes that the position of r' changes infinitely often.*

Proof. Assume by contradiction that there exists some execution of Protocol $\texttt{Async}_2$ such that, eventually, r observes that the position of r' remains unchanged. Consider the suffix of such an execution where r observes that the position of r' remains unchanged. Assume that r' is eventually motionless. By fairness and Remark 3.3, this case is impossible. So, r' moves infinitely often. Thus, each time that r observes r', r' is at the same position. There are two cases to consider:

1. Robot r' eventually sends no bits. In that case, by executing Protocol $\texttt{Async}_2$, r' moves infinitely often in the same direction on H. This contradicts that each time r observes r', r' is in the same position.

2. Robot r' sends bits infinitely often. Since r' is at the same position each time r observes it, r' goes in a direction and comes back at the same position infinitely often. From Protocol $\texttt{Async}_2$, a robot can change its direction only when it observed that the position of the other robot changed twice. Since r' changes its direction infinitely often, r' observed that the position of r changed twice infinitely often. Therefore, from Lemma 3.1, while moving in the same direction, each time r' observes that the position of r changed twice, r observes that the position of r' has changed at least once. A contradiction.

□

Lemmas 3.1 and 3.4 prove that Protocol $\texttt{Async}_2$ ensures the *Receipt* property. Furthermore, Lemma 3.4 guarantees that no robot is starved sending a bit (*i.e.*, it can change its direction infinitely often). So, Property *Emission* is guaranteed by Protocol $\texttt{Async}_2$. This leads to the following theorem:

Theorem 3.5. *Protocol* $\texttt{Async}_2$ *implements one-to-one explicit communication for two robots.*

Note that if each robot r knows the maximum distance $\sigma_{r'}$ that the other robot r' can cover in one step, then, the protocol can be easily adapted to reduce the number of moves made by the robots to send bytes (instead of "bits"). In that case, the total distance $2\sigma_{r'}$ made by r' on its right and its left can be divided by the number of possible bytes sent by the robots. Then, r' moves on its right or on its left of a distance corresponding proportionally to the byte to be sent.

4 One-to-One Communication With Any Number Of Semi-Synchronous Robots

In this section, we adapt the previous results in the design of two protocols working with any number of semi-synchronous robots. In Subsection 4.1, we present our main routing scheme with Protocol $\texttt{Async}_n^I$. It works with the strongest assumptions, *i.e.*, robots equipped with observable IDs and sense of direction. In Subsection 4.2, Protocol $\texttt{Async}_n^A$ provides one-to-one routing for anonymous robots lacking any sense of direction. In the sequel, we omit the upperscript (either I or A) to refer to any of the two protocols ($\texttt{Async}_n^I$ or $\texttt{Async}_n^A$).

4.1 Routing With Identified Robots Having Sense of Direction

First, each robot being *a priori* surrounded by several robots, our method requires the inclusion of a mechanism for avoiding collisions. Next, it must include a technique allowing any robot to send messages to a specific robot. In order to deal with collision avoidance, we use the following concept, *Voronoi diagram*, in the design of our method.

Definition 4.1 (Voronoi diagram)**.** *[2] The Voronoi diagram of a set of points* $P = \{p_1, p_2, \cdots, p_n\}$ *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 *if and only if for any other point* $p_j \in P$, $dist(q, p_i) < dist(q, p_j)$ *where* $dist(p, q)$ *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.*

We assume that the robots know $P(t_0)$, *i.e.*, either the positions of the robots are known by every robot in t_0 or all the robots are awake in t_0. Note that given $P(t_0)$, the Voronoi diagram can be computed in $O(n \log n)$ time [2]. Using $P(t_0)$, when a robot wakes for the time (possibly after t_0), it computes the two following preprocessing steps:

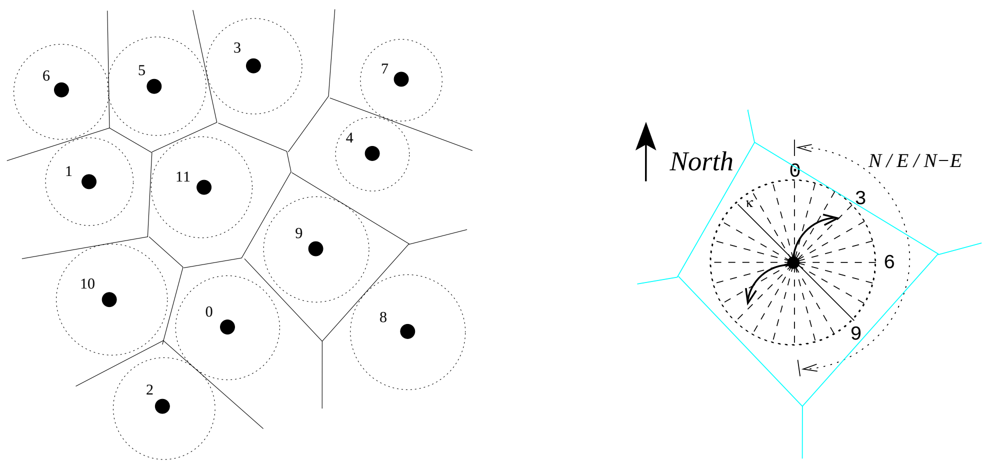
1. Each robot computes the Voronoi Diagram, each Voronoi cell being centered on a robot position—refer to Case (a) in Figure 2, the solid lines show the boundary of the Voronoi cells. Every robot is allowed to move within its Voronoi cell only, ensuring collision avoidance.

2. For each associated Voronoi cell c_r of robot r, each robot r computes the corresponding *granular* g_r, the largest disc of radius R_r centered on r and enclosed in c_r—Case (a) in Figure 2, the dotted lines. Notice that the radii of

different disks might vary. Each granular is sliced into $2n$ slices, *i.e.*, the angle between two adjacent diameters is equal to $\frac{\pi}{n}$. Each diameter is labeled from 0 to $n-1$, the diameter labeled by 0 being aligned to the North, the other numbered in a natural order, progressing the clockwise direction—Case (b) in Figure 2.

Since the robots share a common handedness (chirality), they all agree on the same clockwise direction. Having a common sense of direction, they all agree on the same granular and diameter numbering.

In the sequel, we assume that no robot transmits bits to itself—otherwise, an extra slice would be necessary. For every robot r, let us refer to the diameter labeled with r's ID as κ. In our method, κ plays the role of the horizon line H as for the case with two robots. That is, each robot moves on κ to indicate that it has no bit to transmit.



(a) An example showing a system with 12 robots after the two preprocessing phases.

(b) Robot 9 sends either "0" or "1" to Robot 3.

Figure 2: One-to-one communication with n identified robots with the sense of direction.

We now informally describe Protocol $\texttt{Async}_n^I$ for every robot r. The general scheme is as follows: While $r = r_i$ has no bit to send, r keeps moving on κ in both directions. When r wants to send a bit to a particular robot r', then after it comes back at $p_i(t_0)$ (the center of its granular g_r), r moves on the diameter labeled with r' in either the Northern/Eastern/North-Eastern or the Southern/Western/South-Western direction with respect to H_r and the bit r wants to send to r', either 0 or 1—refer to Case (b) in Figure 2.

As for the case with two robots only, our scheme needs to deal with the asynchronism. That is, by sending either no bit (by moving on κ) or a bit to a particular r' (by moving on the respective diameter of r'), r must make sure that all the robots observed its movements before it changes its direction. With respect to Lemma 3.1, r must move in the same direction until it observes that the position of every robot has changed twice. In order to satisfy this constraint, each time r leaves $p_i(t_0)$ towards the border of g_r in a particular direction, it first moves at a distance d_1 from the border of its granular g_r equal to $R_r - \min(\sigma_r, \frac{R_r}{x})$, with R_r being equal to the radius of g_r and x being a positive real greater than or equal to 2. Next, for each of its k-th movements in the same direction, r moves a distance equal to

$\min(\sigma_r, \frac{d_{k-1}}{x})$, where d_{k-1} is the remaining distance to cover from the current position of r to the border of g_r in that direction.

By applying the same reasoning as in Subsection 3.2 for every pair of robots, we can claim:

Theorem 4.2. *Protocol* $\texttt{Async}_n^I$ *implements one-to-one explicit communication for any number* $n \geq 1$ *of identified robots having the sense of direction.*

4.2 Routing With Anonymous Robots Having No Sense of Direction

With robots devoid of observable IDs, it may seem difficult to send a message to a specific robot. However, it is shown in [15] that if the robots have a sense of direction and chirality, then they can agree on a total order over the robots. This is simply obtained as follows: Each robot r labels every observed robot with its local $x-y$ coordinate in the local coordinate system of r. Even if the robots do not agree on their metric system, by sharing the same x- and y-axes, they agree on the same order.

By contrast, with the lack of sense of direction, due to the symmetry of some configurations, the robots may be unable to deterministically agree on a common labeling for the cohort.

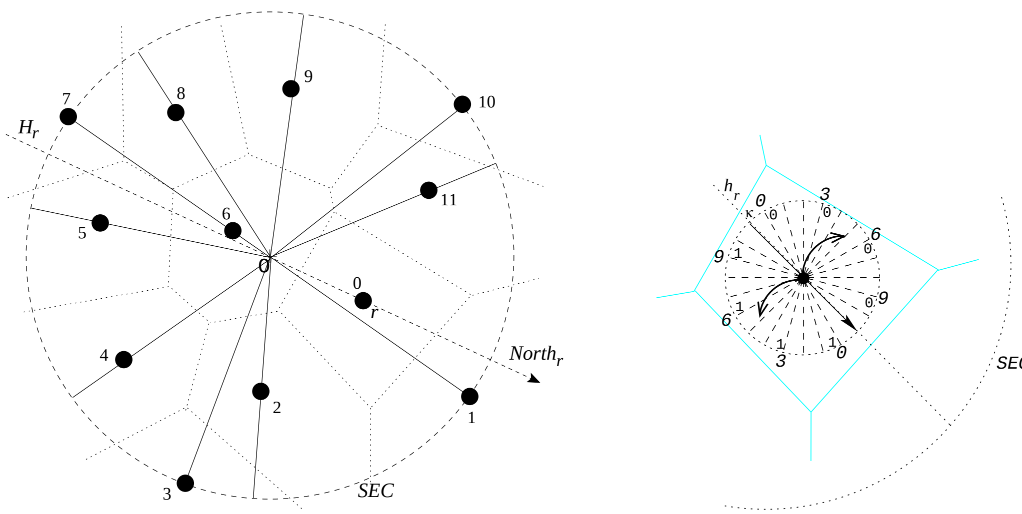
We now describe our method, Protocol $\texttt{Async}_n^A$, in the design of a relative naming (w.r.t. each robot) allowing the implementation of one-to-one communication for anonymous robots with no common sense of direction. We refer to Figure 3 to explain our scheme.

Our method starts (at t_0) with the two preprocessing steps described in Subsection 4.1. At the end we have the Voronoi Diagram and the sliced granulars—to avoid useless overload in the figure, the latter is omitted in Figure 3, Case (a). Then, still at time t_0, each robot r computes the *smallest enclosing circle*, denoted by SEC, of the robot positions. Note that since the robots have the ability of chirality, they can agree on a common clockwise direction of SEC. Also, SEC is unique and can be computed in linear time [21].

Next, if there exists one robot r at $\texttt{O}$, the center of SEC, then r diverges from $\texttt{O}$ by moving on an arbitrary position located on the circumference of a circle centered at $\texttt{O}$ and so that its radius is (strictly) smaller than to the radius of the circle centered at $\texttt{O}$ passing through the closest robots from $\texttt{O}$.

Then, r considers the "*horizon line*", denoted by H_r, as the line passing through itself and $\texttt{O}$. Given H_r, r considers each radius of SEC passing through a robot. The robots are numbered in increasing order following the radii in the clockwise direction starting from H_r. When several robots are located on the same radius, they are numbered in increasing order starting from $\texttt{O}$. Note that this means that r is not necessary labeled by 0 if some robots are located between itself and $\texttt{O}$ on its radius. The robots being devoid of any sense of direction cannot agree on a common North direction. Henceforth, North is given relatively to each robot by its position with respect to $\texttt{O}$. In other words, there exists a "North" relatively to each robot. An example of this preprocessing phase is shown in Case (a) of Figure 3 for a given robot r.

The method to send messages to a given robot is similar to the previous case. Every robot r slices its granular



(a) An example showing how the relative naming is built with respect to r.

(b) An example showing how the granular is sliced with respect to r.

Figure 3: One-to-one communication with n anonymous robots having no sense of direction.

according to H_r into $n+1$ slices. The diameter corresponding to H_r being labeled by 0 and so on in the clockwise direction—refer to Case (b) in Figure 3. Consider the extra slice, corresponding to H_r (the diameter being on radius of SEC passing through r) is not assigned to a particular robot. Let us call this slice κ. Again, κ plays the role of the horizon line H on which every robot moves on to indicate that it has no bit to transmit. The sending of a bit is made following the same scheme as above, the Northern being given by the direction of H_r and the Eastern following the clockwise direction. Each robot addresses bits according to its relative labeling. By construction, the labeling is specific to each robot. However, every robot, r, is able to compute the labeling with respect to each robot of the system. Therefore, by observing each movement made by any robot r', r is able to know to whom a bit is addressed, and in particular, when it is addressed to itself. Every robot is able to compute the message address, by being able to compute the relative naming of all the robots. Also notice that in this case the protocol require the robots to have visibility of at most the diameter of the SEC.

Theorem 4.3. *Protocol* $\mathtt{Async}_n^A$ *implements one-to-one explicit communication for any number* $n \geq 1$ *of anonymous robots having no sense of direction.*

5 Motion Containment and Limited Visibility

The above schemes have the drawback of either making the robots moving away from each other infinitely often (Protocol $\mathtt{Async}_2$) or requiring that the robots are able to move (and to observe) an infinitesimally small distance (both Protocols $\mathtt{Async}_n$). Note that this is due to the extreme weakness of the system considered in this paper. Indeed, the only assumption made on the concurrent activation of robots is uniform fairness. This means that no assumption models the "*interleaving*" of activations of each robot with respect to the others. In other words, there is no bound between two consecutive actions of a robot with respect to the activation of others. We now introduce a certain amount of synchrony among the robots called interleaving degree.

Definition 5.1 (Interleaving Degree)**.** *Let $k > 0$ be the interleaving degree such that, for every pair of distinct robots r and r', for every suffix of computation in which r is activated k times, r' is activated at least once.*

Note that if $k = 1$, then the system is (fully) *synchronous*, *i.e.,* every robot is activated at each time instant. Assuming that $k > 1$, there is a certain amount of asynchrony ensuring that every k moves of any robot r, every robot r' ($\neq r$) observed at least one move made by r. The following lemma is straightforward:

Lemma 5.2. *Let r and r' be two robots. Assuming an interleaving degree of $k \geq 1$, every k moves of r, then r' have observed that the position of r has changed at least once.*

Lemma 5.2 plays the same role as Lemma 3.1. So, assuming an interleaving degree of $k \geq 1$, for any pair of robots r and r', every $2k$ activations of a robot r, r sees r' moving at least twice and r' sees r moving at least once.

We directly use this result to modify Protocol $\texttt{Async}_2$ as follows: Instead of moving infinitely often in the opposite direction of r' on H, r comes back on its initial position after each bit sent, or byte sent if r knows $\sigma_{r'}$, see Section 3.2. This ensures that each robot r does not move farther than a distance d_r equal to $k\sigma_r$.

Note that by making this modification, both robots are no longer required to have an infinite visibility. If each robot knows k and $\sigma_{r'}$, then the visibility of r can be bounded by $k(\sigma_r + \sigma_{r'}) + \delta$, with δ being equal to the distance between the initial positions of both r and r'. Otherwise, the visibility can be finite but cannot be bounded.

Similarly, if for robot r', every robot r is able to observe that r' moved a distance equal or less than $d_{r'} = \min(\sigma_{r'}, \frac{R_{r'}}{k})$, then Protocol $\texttt{Async}_n$ works assuming that each robot r moves a distance equal to d_r. As for the case with two robots, the visibility of the robots is no longer required to be infinite, and can be bounded to the radius of the smallest enclosing circle.

Note that this latter bound can be reduced in the case when the robots have observable IDs. Indeed, Protocol $\texttt{Async}_n^I$ assumes that the observable IDs of the robots can be mapped to the range $1 \ldots n$. Thus, combined with a "classical" algorithm to maintain a routing table [25], the same scheme can be easily used to implement one-to-one communication among robots with a visibility limited to its neighbors only, provided that (i) every robot knows n, and (ii) no robot movement breaks the graph of observability [16]. Each robot adds the ID of the addressee and (directly) communicates with its neighbors using our protocol. The message is then routed towards its destination using any "classical" routing algorithm.

6 Concluding Remarks, Extensions, and Open Problems

In this work we proposed (deterministic) movement protocols that implement explicit communication, and therefore allow the application of (existing) distributed algorithms that use message exchanges. Movements-signals are introduced as a means to transfer messages between deaf and dumb robots. The movement protocols can serve as a backup to other means of (e.g., wireless) communication. Several protocols and enhancements have been proposed that implement one-to-one communication in various semi-synchronous environments. Note that our solutions allow every robot to read each message sent by any robot r to any robot r'. This provides fault-tolerance by redundancy, any robot being able to send any message again. This also enables one-to-many or one-to-all communication. For instance, one-to-all communication can be implemented in Protocol $\texttt{Async}_n$ by adding an extra slice labeled by $n+1$ intended to communicate a message to every robot.

We call a communication protocol *silent* when a robot eventually moves only if it has some message to transmit. Note that this desirable property would help to save energy resources of the robots. The proposed semi-synchronous protocols are not silent (Remark 3.3). The question of whether the design of silent semi-synchronous algorithms is possible or not remains open.

A related issue concerns the distance (eventually) covered by the robots. In this paper, we provided a solution to overcome the problem of limited moves number by introducing a certain amount of synchrony among the robots, the degree of interleaving. As a matter of fact, we believe that the lack of a bound on the degree of interleaving implies an impossibility for communication by a finite number of moves. Intuitively, this arises from the fact that a semi-synchronous robot that sequentially observes another robot at the same place cannot determine whether the robot moved and returned to the same position or did not move at all.

Computations with an *infinite decimal precision* are different and, in a way, represent a weaker assumption than infinitely small movements. Indeed, one can assume infinite decimal precision with the "reasonable" assumption of finite movements, *i.e.*, with a *minimal* and *maximal* distance covered in one atomic step, or even step over a grid. In this paper, we assumed a maximal covered distance (σ_r), but not a minimal covered distance. This would be the case by assuming that the plane is either a grid or a hexagonal pavement [18]. For instance, with such assumption, the robots could be prone to make computation errors due to round off, and, therefore, face a situation where robots are not able to identify all of possible $2n$ directions obtained by slices inside of disks and are limited to recognize only a certain number of directions. This case could be solved by avoiding the use of $2n$ slices of granular by transmitting the index of the robot to whom the message intended following the message itself. For this we would need only $k+1$, $1 \leq k < 2n$ segments (or $2k+1$ slices). In particular, we would use one segment for message transmission (as in the case of two robots); using the other k segments the robot who wants to transmit a message allows to transmit the index of the robot to whom the message is designated. Definitely, such index can be represented by $\frac{\log n}{\log k} = \log_k n$ symbols. Notice, this strategy would slow down the algorithm and increase the number of steps required to transmit a message. More precisely, the number of steps required in this method to identify the designated robot is $\log_k n$. For example, by taking $O(\log n)$ slices instead of $O(n)$, the number of steps to transmit a message would increase by $O(\frac{\log n}{\log \log n})$.

In the continuation of the above discussion, the other important feature in the field of mobile robots is the weakness/strength of the model. For instance, in this paper, we used the semi-synchronous model (SSM). It would be interesting to relax synchrony among the robots in order to reach solutions for a fully asynchronous model (*e.g.*, CORDA [23]).

Finally, *stabilization* [12] would be a very desirable property to enable. It seems that, in our case, stabilization could be achieved assuming an interleaving degree equal to 1 (*i.e.*, synchronous settings) by carefully adapting Protocol $\texttt{Async}_n$, say by assuming a global clock (using GPS input) returning to the initial location and (re)computing the preprocessing phase every round timestamp. The self-stabilization property assuming no interleaving degree (*i.e.*, the semi-synchronous case) requires further study.

References

[1] H Ando, Y Oasa, I Suzuki, and M Yamashita. A distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Transaction on Robotics and Automation*, 15(5):818–828, 1999.

[2] F Aurenhammer. Voronoi diagrams- a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, 1991.

[3] R. Beckers, O.E. Holland, and J.L. Deneubourg. From local actions to global tasks: Stigmergy and collective robotics. *Artificial Life, 4th Int. Worksh. on the Synth. and Simul. of Living Sys.*, 4:173–202, 1994.

[4] Zohir Bouzid, Shlomi Dolev, Maria Potop-Butucaru, and Sébastien Tixeuil. Robocast: Asynchronous communication in robot networks. In *OPODIS*, pages 16–31, 2010.

[5] M Cieliebak, P Flocchini, G Prencipe, and N Santoro. Solving the robots gathering problem. In *Proceedings of the 30th International Colloquium on Automata, Languages and Programming (ICALP 2003)*, pages 1181–1196, 2003.

[6] R Cohen and D Peleg. Local spreading algorithms for autonomous robot systems. *Theor. Comput. Sci.*, 399(1-2):71–82, 2008.

[7] C.R.Kube. Task modelling in collective robotics. *Auton. Robots*, 4(1):53–72, 1997.

[8] X Defago and A Konagaya. Circle formation for oblivious anonymous mobile robots with no common sense of orientation. In *2nd ACM International Annual Workshop on Principles of Mobile Computing (POMC 2002)*, pages 97–104, 2002.

[9] Y Dieudonné, O Labbani-Igbida, and F Petit. Circle formation of weak mobile robots. *ACM Transactions on Autonomous and Adaptive Systems*, 3(4), 2008.

[10] Y Dieudonné and F Petit. Deterministic leader election in anonymous sensor networks without common coodinated system. In *11th International Conference On Principles of Distributed Systems (OPODIS 2007)*, volume 4878 of *Lecture Notes in Computer Science, Springer*, pages 132–142, 2007.

[11] Yoann Dieudonné, Shlomi Dolev, Franck Petit, and Michael Segal. Deaf, dumb, and chatting asynchronous robots. In *OPODIS*, pages 71–85, 2009.

[12] S. Dolev. *Self-Stabilization*. The MIT Press, 2000.

[13] Andries P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons, 2006.

[14] E.O.Wilson. *Sociobiology*. Belknap Press of Harward University Press, 1975.

[15] 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 *10th Annual International Symposium on Algorithms and Computation (ISAAC 99)*, pages 93–102, 1999.

[16] 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.

[17] T. Fukuda and S. Nakagawa. Approach to the dynamically reconfigurable robotic sytem. *Journal of Intelligent and Robotic System*, 1:55–72, 1988.

[18] N Gordon, I A. Wagner., and A M. Brucks. Discrete bee dance algorithms for pattern formation on a grid. In *IAT '03: Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology*, pages 545–549. IEEE Computer Society, 2003.

[19] P-P Grassé. La reconstruction du nid et les coordinations inter-individuelles chez bellicosi-termes natalensis et cubitermes sp. la theorie de la stigmergie: Essai dinterpretation des termites constructeurs. *Insectes Sociaux*, 6:41–83, 1959.

[20] M.J. Matarić. Issues and approaches in the design of collective autonomous agents. *Robotics and Autonomous Systems*, 16(2-4):321–331, 1995.

[21] N Megiddo. Linear-time algorithms for linear programming in R^3 and related problems. *SIAM Journal on Computing*, 12(4):759–776, 1983.

[22] F.R. Noreils. An architecture for cooperative and autonomous mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 2703–2710, 1992.

[23] G Prencipe. *Distributed Coordination of a Set of Autonomous Mobile Robots.* PhD thesis, Dipartimento di Informatica, University of Pisa, 2002.

[24] G Prencipe and N Santoro. Distributed algorithms for autonomous mobile robots. In *The 2006 IFIP International Conference on Embedded And Ubiquitous Computing (EUC 2006)*, 2006.

[25] N Santoro. *Design and Analysis of Distributed Algorithms*. John Wiley & Sons, Inc., 2007.

[26] I Suzuki and M Yamashita. Distributed anonymous mobile robots - formation of geometric patterns. *SIAM Journal of Computing*, 28(4):1347–1363, 1999.

[27] C Tovey. The honey bee algorithm: A biological inspired approach to internet server optimization. *Engineering Enterprise, the Alumni Magazine for ISyE at Georgia Institute of Technology*, Spring 2004:13–15, 2004.

[28] H.F Wedde and M Farooq. A comprehensive review of nature inspired routing algorithms for fixed telecommunication networks. *J. Syst. Archit.*, 52(8):461–484, 2006.

[29] P.H Wenner and A.M Wells. *Anatomy of a Controversy: the Question of a "Language" Among Bees*. Columbia University Press, New-York, 1990.

[30] Masafumi Yamashita and Ichiro Suzuki. Characterizing geometric patterns formable by oblivious anonymous mobile robots. *Theor. Comput. Sci.*, 411(26-28):2433–2453, 2010.