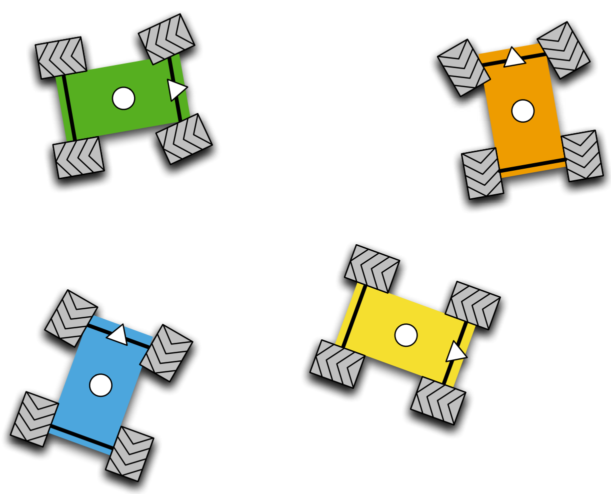
Distributed Control of Robotic Networks

A Mathematical Approach to Motion Coordination Algorithms

Chapter 3: Robotic network models and complexity notions

Francesco Bullo
Jorge Cortés
Sonia Martínez

May 20, 2009



PRINCETON UNIVERSITY PRESS
PRINCETON AND OXFORD

Distributed Control of Robotic Networks, by Francesco Bullo, Jorge Cortés and Sonia Martínez, Applied Mathematics Series, Princeton University Press, 2009, ISBN 978-0-691-14195-4.

Contents

Chapter Three

Robotic network models and complexity notions

This chapter introduces the main subject of study of this book, namely a model for groups of robots that sense their own position, exchange messages according to a geometric communication topology, process information, and control their motion. We refer to such systems as robotic networks. The content of this chapter has evolved from Martínez et al. (2007).

The chapter is organized as follows. The first section contains the formal model. We begin by presenting the physical components of a network, that is, the mobile robots and the communication service connecting them. We then present the notion of control and communication law, and how a law is executed by a robotic network. These notions subsume the notions of synchronous network and distributed algorithm described in Section 1.5. As an example of these notions, we introduce a simple law, called the agree and pursue law, which combines ideas from leader election algorithms and from cyclic pursuit (i.e., a game in which robots chase each other in a circular environment). In the second section, we propose a model of groups of robots that interact through sensing, rather than communication. The third section discusses time, space, and communication complexity notions for robotic networks as extensions of the corresponding notions for distributed algorithms. The complexity notions rely on the basic concept of coordination task and task achievement. The fourth and last section establishes the time, space, and communication required by the agree and pursue law to steer a group of robots to a uniformly spaced rotating configuration. We end the chapter with three sections on, respectively, bibliographical notes, proofs of the results presented in the chapter, and exercises.

3.1 A MODEL FOR SYNCHRONOUS ROBOTIC NETWORKS

Here, we introduce a model for a synchronous robotic network. This model is an extension of the synchronous network model presented in Section 1.5.1. We start by detailing the physical components of the network, which include the robots themselves as well as the communication service among them.

3.1.1 Physical components

We start by providing a basic definition of a robot and a model for how each robot moves in space.

A *mobile robot* is a continuous-time continuous-space dynamical system as defined in Section 1.3, that is, a tuple (X, U, X_0, f), where

(i) X is d-dimensional space chosen among $\mathbb{R}^d$, $\mathbb{S}^d$, and the Cartesian products $\mathbb{R}^{d_1} \times \mathbb{S}^{d_2}$, for some $d_1 + d_2 = d$, called the *state space*;

(ii) U is a compact subset of $\mathbb{R}^m$ containing $\mathbf{0}_m$, called the *input space*;

(iii) X_0 is a subset of X, called the *set of allowable initial states*; and

(iv) $f : X \times U \to \mathbb{R}^d$ is a continuously differentiable control vector field on X, that is, f determines the robot motion $x : \mathbb{R}_{\geq 0} \to X$ via the differential equation, or control system,

$$\dot{x}(t) = f(x(t), u(t)), \tag{3.1.1}$$

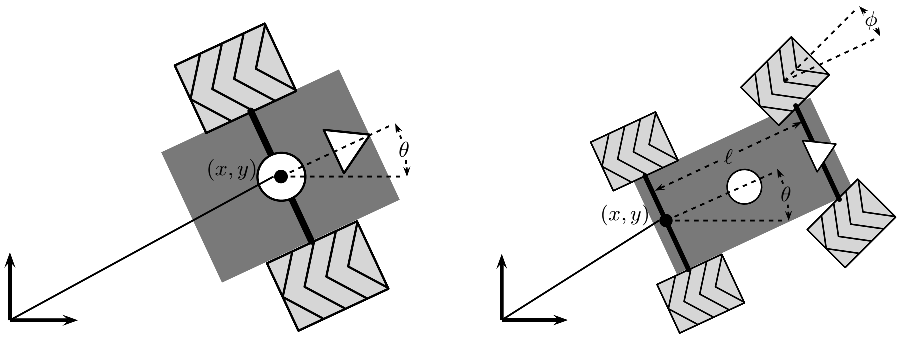
subject to the control $u : \mathbb{R}_{\geq 0} \to U$.

We will use the terms "robot" and "agent" interchangeably. We refer to $x \in X$ and $u \in U$ as a *physical state* and an *input* of the mobile robot, respectively. Most often, the physical state will have the interpretation of a location, or a location and velocity. We will often consider control-affine vector fields. In such a case, we represent f as the ordered family of continuously differentiable vector fields $(f_0, f_1, \dots, f_m)$ on X. In general, the control signal u will not depend only on time but also on x and possible other variables in the system. Note that there is no additional difficulty in modeling mobile robots using dynamical systems defined on manifolds (Bullo and Lewis, 2004), but we avoid it here in the interest of simplicity.

Example 3.1 (Planar vehicle models). The following models of control systems are commonly used in robotics, beginning with the early works of Dubins (1957), and Reeds and Shepp (1990). Figures 3.1(a) and (b) show a two-wheeled vehicle and a four-wheeled vehicle, respectively. The two-wheeled planar vehicle is described by the dynamical system

$$\dot{x} = v \cos\theta, \quad \dot{y} = v \sin\theta, \quad \dot{\theta} = \omega, \tag{3.1.2}$$

with state variables $x \in \mathbb{R}$, $y \in \mathbb{R}$, and $\theta \in \mathbb{S}^1$, describing the planar position and orientation of the vehicle, and with controls v and ω, describing the forward linear velocity and the angular velocity of the vehicle. Depending on which set the controls are restricted to, we define the following models:



(a) (b)

Figure 3.1 A two-wheeled vehicle (a) and four-wheeled vehicle (b). In each case, the orientation of the vehicle is indicated by the small triangle.

The unicycle. The controls v and ω take value in $[-1, 1]$ and $[-1, 1]$, respectively.

The differential drive robot. Set $v = (\omega_{\text{right}} + \omega_{\text{left}})/2$ and $\omega = (\omega_{\text{right}} - \omega_{\text{left}})/2$ and assume that both ω_{right} and ω_{left} take value in $[-1, 1]$.

The Reeds–Shepp car. The control v takes values in $\{-1, 0, 1\}$ and the control ω takes values in $[-1, 1]$.

The Dubins vehicle. The control v is set equal to 1 and the control ω takes value in $[-1, 1]$.

Finally, the four-wheeled planar vehicle, composed of a front and a rear axle separated by a distance ℓ, is described by the same dynamical system (3.1.2) with the following distinctions: $(x, y) \in \mathbb{R}^2$ is the position of the midpoint of the rear axle, $\theta \in \mathbb{S}^1$ is the orientation of the rear axle, the control v is the forward linear velocity of the rear axle, and the angular velocity satisfies $\omega = \frac{v}{\ell} \tan \phi$, where the control ϕ is the steering angle of the vehicle. •

Next, we generalize the notion of synchronous network introduced in Definition 1.38 and introduce a corresponding notion of robotic network.

Definition 3.2 (Robotic network). The physical components of a *robotic network* $\mathcal{S}$ consist of a tuple $(I, \mathcal{R}, E_{\text{cmm}})$, where

(i) $I = \{1, \dots, n\}$, I is called the *set of unique identifiers (UIDs)*;

(ii) $\mathcal{R} = \{R^{[i]}\}_{i \in I} = \{(X^{[i]}, U^{[i]}, X_0^{[i]}, f^{[i]})\}_{i \in I}$ is a set of mobile robots;

(iii) E_{cmm} is a map from $\prod_{i \in I} X^{[i]}$ to the subsets of $I \times I$—this map is called the *communication edge map*.

Additionally, if all mobile robots are identical, that is, if $R^{[i]} = (X, U, X_0, f)$ for all $i \in \{1, \dots, n\}$, then the robotic network is *uniform*. •

Remarks 3.3 (Notational conventions and meaning of the communication edge map).

(i) Following the convention established in Section 1.5, we let the superscript $[i]$ denote the variables and spaces which correspond to the robot with unique identifier i; for instance, $x^{[i]} \in X^{[i]}$ and $x_0^{[i]} \in X_0^{[i]}$ denote the physical state and the initial physical state of robot $R^{[i]}$, respectively. We refer to $x = (x^{[1]}, \dots, x^{[n]}) \in \prod_{i \in I} X^{[i]}$ as a *state* of the network.

(ii) The map $x \mapsto (I, E_{\text{cmm}}(x))$ models the topology of the communication service among the robots: at a physical state $x = (x^{[1]}, \dots, x^{[n]})$, two robots at locations $x^{[i]}$ and $x^{[j]}$ can communicate if and only if the pair (i, j) is an edge in $E_{\text{cmm}}(x) = E_{\text{cmm}}(x^{[1]}, \dots, x^{[n]})$. Accordingly, we refer to $(I, E_{\text{cmm}}(x))$ as the *communication graph* at x. When and which robots communicate is discussed in Section 3.1.2. As communication graphs, we will often adopt one of the proximity graphs discussed in Section 2.2, and in particular the (undirected) disk graph. •

To make things concrete, let us present some examples of robotic networks that will be commonly used later.

Example 3.4 (First-order robots with range-limited communication). Consider a group of robots moving in $\mathbb{R}^d$, $d \geq 1$. As in Chapter 2, we let p denote a point in $\mathbb{R}^d$ and we let $\{p^{[1]}, \dots, p^{[n]}\}$ denote the robot locations. Assume that the robots move according to

$$\dot{p}^{[i]}(t) = u^{[i]}(t), \tag{3.1.3}$$

with $u^{[i]} \in [-u_{\max}, u_{\max}]^d$; for an illustration, see Figure 3.2. According to our mobile robot notation, these are identical robots of the form

$$(\mathbb{R}^d, [-u_{\max}, u_{\max}]^d, \mathbb{R}^d, (\mathbf{0}_d, \boldsymbol{e}_1, \dots, \boldsymbol{e}_d)).$$

We assume that each robot can sense its own position and can communicate with any other robot within distance r, that is, we adopt the r-disk graph $\mathcal{G}_{\text{disk}}(r)$ defined in Section 2.2 as communication graph. These data define the uniform robotic network $\mathcal{S}_{\text{disk}}$.

It will also be interesting to consider first-order robots with communication graphs other than the disk graph; important examples include the Delaunay graph $\mathcal{G}_{\text{D}}$, the limited Delaunay graph $\mathcal{G}_{\text{LD}}(r)$, and the ∞-disk

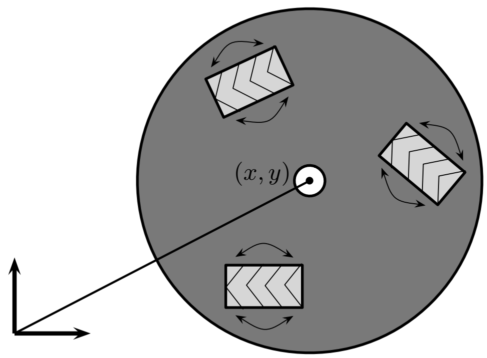


Figure 3.2 An omnidirectional vehicle. In addition to controlling the rotation speed of the wheels, the vehicle can also actuate the direction in which they point. This allows the vehicle to move in any direction according to the first-order dynamics (3.1.3).

graph $\mathcal{G}_{\infty\text{-disk}}(r)$, discussed in Section 2.2. These three graphs, adopted as communication models, give rise to three robotic networks denoted $\mathcal{S}_{\mathrm{D}}$, and $\mathcal{S}_{\mathrm{LD}}$, $\mathcal{S}_{\infty\text{-disk}}$, respectively. •

Example 3.5 (Planar vehicle robots with Delaunay communication). We consider a group of vehicle robots moving in an allowable environment $Q \subset \mathbb{R}^2$ according to the planar vehicle dynamics introduced in Example 3.1. We let $\{(p^{[1]}, \theta^{[1]}), \dots, (p^{[n]}, \theta^{[n]})\}$ denote the robot physical states, where $p^{[i]} = (x^{[i]}, y^{[i]}) \in Q$ corresponds to the position and $\theta^{[i]} \in \mathbb{S}^1$ corresponds to the orientation of the robot $i \in I$. As the communication graph, we adopt the Delaunay graph $\mathcal{G}_{\mathrm{D}}$ on Q introduced in Section 2.2. These data define the uniform robotic network $\mathcal{S}_{\mathrm{vehicles}}$. •

Example 3.6 (Robots with line-of-sight communication). We consider a group of robots moving in an allowable environment $Q \subset \mathbb{R}^2$. As in Example 3.4, we let $\{p^{[1]}, \dots, p^{[n]}\}$ denote the robot locations and we assume that the robots move according to the motion model (3.1.3). Each robot can sense its own position and the boundary of ∂Q, and can communicate with any other robot within distance r and within line of sight, that is, we adopt the range-limited visibility graph $\mathcal{G}_{\text{vis-disk},Q}$ in Q defined in Section 2.2 as the communication graph. These data define the uniform robotic network $\mathcal{S}_{\text{vis-disk}}$. •

Example 3.7 (First-order robots in $\mathbb{S}^1$). Consider a group of n robots $\{\theta^{[1]}, \dots, \theta^{[n]}\}$ in $\mathbb{S}^1$, moving along on the unit circle with an angular velocity equal to the control input. Each identical robot is described by the tuple $(\mathbb{S}^1, [-u_{\max}, u_{\max}], \mathbb{S}^1, (0, \boldsymbol{e}))$, where $\boldsymbol{e}$ is the vector field on $\mathbb{S}^1$ describing unit-speed counterclockwise rotation. As in the previous examples, we assume that each robot can sense its own position and can communicate with any other robot within distance r along the circle, that is, we adopt the r-disk graph $\mathcal{G}_{\mathrm{disk}}(r)$ on $\mathbb{S}^1$ defined in Section 2.2 as the communication graph. These data define the uniform robotic network $\mathcal{S}_{\mathrm{circle}}$. •

We conclude this section with a remark.

Remark 3.8 (Congestion models in robotic networks). The behavior of a robotic network might be affected by communication and physical congestion problems.

Communication congestion: Omnidirectional wireless transmissions interfere. Clear reception of a signal requires that no other signals are present at the same point in time and space. In an *ad hoc* network, node i receives a message transmitted by node j only if all other neighbors of i are silent. In other words, the transmission medium is shared among the agents. As the density of agents increases, so does wireless communication congestion. The following asymptotic and optimization results are known.

First, for *ad hoc* networks with n uniformly randomly placed nodes, it is known (Gupta and Kumar, 2000) that the maximum-throughput communication range $r(n)$ of each node decreases as the density of nodes increases; in d dimensions, the appropriate scaling law is $r(n) \in \Theta\big((\log(n)/n)^{1/d}\big)$. This is referred to as the *connectivity regime* in percolation theory and statistical mechanics. Using the k-nearest-neighbor graph over uniformly placed nodes, the analysis in Xue and Kumar (2004) suggests that the minimal number of neighbors in a connected network grows with $\log(n)$.

Second, a growing body of literature (Santi, 2005; Lloyd et al., 2005) is available on *topology control*, that is, on how to compute transmission power values in an *ad hoc* network so as to minimize energy consumption and interference (due to multiple sources), while achieving various graph topological properties, such as connectivity or low network diameter.

Physical congestion. Robots can collide: it is clearly important to avoid "simultaneous access to the same physical area" by multiple robots. It is reasonable to assume that, as the number of robots increases, so should the area available for their motion. A convenient alternative approach is the one taken by Sharma et al. (2007), where robots' safety zones decrease with decreasing robot speed. This suggests that, in a fixed environment, individual nodes of a large ensemble have to move at a speed decreasing with n, and in particular, at a speed proportional to $n^{-1/d}$. Roughly speaking, if the overall volume V in which the groups of agents move is constant, and there are n robots, then the speed v at which they can move goes approximately as $v^d \approx \frac{V}{n}$.

In summary, one way to incorporate congestion effects into the robotic network model is to assume that the parameters of the network physical components depend upon the number of robots n. In the limit as $n \to +\infty$, we will sometimes assume that r and $u_{\max}$, the communication range and the velocity upper bound in Examples 3.4 and 3.7, are of order $n^{-1/d}$. •

3.1.2 Control and communication laws

Here, we present a discrete-time communication, continuous-time motion model for the evolution of a robotic network subject to a communication and control law. In our model, each robot evolves in the physical domain in continuous time, senses its position in continuous time, and, in discrete time, exchanges information with other robots and executes a state machine, which we shall refer to as a processor. The following definition is a generalization of the concept of distributed algorithm introduced in Definition 1.39 and of the classical notion of dynamical feedback controller.

Definition 3.9 (Control and communication law). A *control and communication law* $\mathcal{CC}$ for a robotic network $\mathcal{S}$ consists of the sets:

(i) $\mathbb{A}$, a set containing the `null` element, called the *communication alphabet*—elements of $\mathbb{A}$ are called *messages*;

(ii) $W^{[i]}$, $i \in I$, called the *processor state sets*; and

(iii) $W_0^{[i]} \subseteq W^{[i]}$, $i \in I$, sets of *allowable initial values*;

and of the following maps:

(i) $\text{msg}^{[i]} : X^{[i]} \times W^{[i]} \times I \to \mathbb{A}$, $i \in I$, called *message-generation functions*;

(ii) $\text{stf}^{[i]} : X^{[i]} \times W^{[i]} \times \mathbb{A}^n \to W^{[i]}$, $i \in I$, called *(processor) state-transition functions*; and

(iii) $\text{ctl}^{[i]} : X^{[i]} \times X^{[i]} \times W^{[i]} \times \mathbb{A}^n \to U^{[i]}$, $i \in I$, called *(motion) control functions.*

If $\mathcal{S}$ is uniform and if $W^{[i]} = W$, $\text{msg}^{[i]} = \text{msg}$, $\text{stf}^{[i]} = \text{stf}$, and $\text{ctl}^{[i]} = \text{ctl}$, for all $i \in I$, then $\mathcal{CC}$ is said to be *uniform* and is described by a tuple $(\mathbb{A}, W, \{W_0^{[i]}\}_{i \in I}, \text{msg}, \text{stf}, \text{ctl})$. •

We will sometimes refer to a control and communication law as a *distributed motion coordination algorithm*. Roughly speaking, the rationale behind Definition 3.9 is as follows (see Figure 3.3). The state of robot i in-

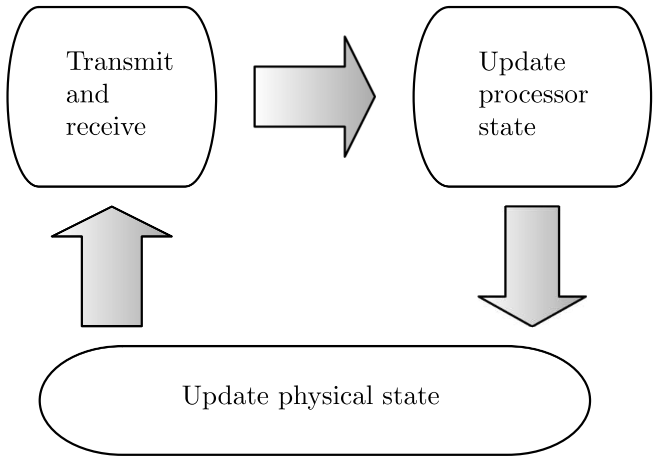


Figure 3.3 The execution of a control and communication law by a robotic network.

cludes both the physical state $x^{[i]} \in X^{[i]}$ and the *processor state* $w^{[i]} \in W^{[i]}$ of the state machine that robot i implements. These states are initialized with values in their corresponding allowable initial sets $X_0^{[i]}$ and $W_0^{[i]}$. We assume that the robot can sense it own physical position $x^{[i]}$. At each time instant $\ell \in \mathbb{Z}_{\geq 0}$, robot i sends to each of its out-neighbors j in the communication digraph $(I, E_{\text{cmm}}(x))$ a message (possibly the `null` message) computed by applying the message-generation function $\text{msg}^{[i]}$ to the current values of its physical state $x^{[i]}$ and processor state $w^{[i]}$, and to the identity j. Subsequently, but still at the time instant $\ell \in \mathbb{Z}_{\geq 0}$, robot i updates the value of its processor state $w^{[i]}$ by applying the state-transition function $\text{stf}^{[i]}$ to the current value of its physical state $x^{[i]}$, processor state $w^{[i]}$ and to the messages it receives from its in-neighbors. Between communication instants, that is, for $t \in [\ell, \ell+1)$ for some $\ell \in \mathbb{Z}_{\geq 0}$, the motion of the ith robot is determined by applying the control function to the current value of $x^{[i]}$, the value of $x^{[i]}$ at time ℓ, the current value of $w^{[i]}$, and the messages received at time ℓ. This evolution model is very similar to the one that we introduced for synchronous networks in Definition 1.40: in each communication round, the first step is transmission and the second one is computation and, except for the dependence on the physical state x, the communication and state transition processes are identical.

These ideas are formalized in the following definition.

Definition 3.10 (Evolution of a robotic network). Let $\mathcal{CC}$ be a control and communication law for the robotic network $\mathcal{S}$. The *evolution* of $(\mathcal{S}, \mathcal{CC})$ from initial conditions $x_0^{[i]} \in X_0^{[i]}$ and $w_0^{[i]} \in W_0^{[i]}$, $i \in I$, is the collection of curves $x^{[i]} : \mathbb{R}_{\geq 0} \to X^{[i]}$ and $w^{[i]} : \mathbb{Z}_{\geq 0} \to W^{[i]}$, $i \in I$, defined by

$$\dot{x}^{[i]}(t) = f\Big(x^{[i]}(t), \operatorname{ctl}^{[i]}\big(x^{[i]}(t), x^{[i]}(\lfloor t \rfloor), w^{[i]}(\lfloor t \rfloor), y^{[i]}(\lfloor t \rfloor)\big)\Big),$$

where $\lfloor t \rfloor = \max\{\ell \in \mathbb{Z}_{\geq 0} \mid \ell < t\}$, and

$$w^{[i]}(\ell) = \operatorname{stf}^{[i]}(x^{[i]}(\ell), w^{[i]}(\ell - 1), y^{[i]}(\ell)),$$

with $x^{[i]}(0) = x_0^{[i]}$ and $w^{[i]}(-1) = w_0^{[i]}$, $i \in I$. In the previous equations, $y^{[i]} : \mathbb{Z}_{\geq 0} \to \mathbb{A}^n$ (describing the messages received by processor i) has components $y_j^{[i]}(\ell)$, for $j \in I$, defined by

$$y_j^{[i]}(\ell) = \begin{cases} \operatorname{msg}^{[j]}(x^{[j]}(\ell), w^{[j]}(\ell-1), i), & \text{if } (j, i) \in E_{\mathrm{cmm}}\big(x^{[1]}(\ell), \dots, x^{[n]}(\ell)\big), \\ \texttt{null}, & \text{otherwise.} \end{cases}$$

•

For convenience, we define $w(t) = w(\lfloor t \rfloor)$ for all $t \in \mathbb{R}_{\geq 0}$, and let $\mathbb{R}_{\geq 0} \ni t \mapsto (x(t), w(t))$ denote the curves $x^{[i]}$ and $w^{[i]}$, for $i \in \{1, \dots, n\}$.

Remarks 3.11 (Simplifications of control and communication laws).

(i) A control and communication law $\mathcal{CC}$ is *static* if the processor state set $W^{[i]}$ is a singleton for all $i \in I$. This means that there is no meaningful evolution of the processor state. In this case, $\mathcal{CC}$ can be described by a tuple $(\mathbb{A}, \{\operatorname{msg}^{[i]}\}_{i \in I}, \{\operatorname{ctl}^{[i]}\}_{i \in I})$, with $\operatorname{msg}^{[i]} : X^{[i]} \times I \to \mathbb{A}$, and $\operatorname{ctl}^{[i]} : X^{[i]} \times X^{[i]} \times \mathbb{A}^n \to U^{[i]}$, for $i \in I$.

(ii) A control and communication law $\mathcal{CC}$ is *data-sampled* if the control functions are independent of the current position of the robot and depend only upon the robot's position at the last sample time. Specifically, the control functions have the following property: given a processor state $w^{[i]} \in W^{[i]}$, an array of messages $y^{[i]} \in \mathbb{A}^n$, a current state $x^{[i]}$, and a state at last sample time $x_{\mathrm{smpld}}^{[i]}$, the control input $\operatorname{ctl}^{[i]}(x^{[i]}, x_{\mathrm{smpld}}^{[i]}, w^{[i]}, y^{[i]})$ is independent of $x^{[i]}$, for all $i \in I$. In this case, the control functions can be described by maps of the form $\operatorname{ctl}^{[i]} : X^{[i]} \times W^{[i]} \times \mathbb{A}^n \to U^{[i]}$, for $i \in I$.

(iii) In many control and communication laws, the robots exchange full information about their states, including both their processor and their physical states. For such laws, we identify the communication alphabet with $\mathbb{A} = (X \times W) \cup \{\texttt{null}\}$ and we refer to the corresponding message-generation function $\operatorname{msg}_{\mathrm{std}}(x, w, j) = (x, w)$ as the *standard message-generation function*.

Note that we allow the processor state set and the communication alphabet to contain an infinite number of symbols. In other words, we assume that a robot can store and transmit a (finite number of) integer and real numbers, among other things. This is equivalent to assuming that we neglect any inaccuracies due to quantization, as we did in Section 1.6. •

Remark 3.12 (Extensions of control and communication laws). Here, we briefly discuss alternative models and extensions of the proposed models.

Asynchronous sensor-based interactions. In the early network model proposed by Suzuki and Yamashita (1999), robots are referred to as "anonymous" and "oblivious" in precisely the same way in which we defined control and communication laws to be uniform and static, respectively. As compared with our notion of robotic network, the model in Suzuki and Yamashita (1999) is more general in that the robots' activation schedules do not necessarily coincide (i.e., this model is asynchronous), and at the same time it is less general in that (1) robots cannot communicate any information other than their respective positions, and (2) each robot observes every other robot's position (i.e., the complete communication graph is adopted). In the Section 3.2 below, we present a model in which robots rely on sensing rather than communication for their interaction.

Discrete-time motion models. For some algorithms in later chapters, it will be convenient to consider discrete-time motion models; for example, we present discrete-time motion models for first-order agents in Section 4.1. In some other cases, it will be convenient to consider dynamical interactions between agents taking place in continuous time.

Stochastic link models. Although we do not present any results on this topic in this notes, it is possible to develop robotic networks models over random graphs and random geometric graphs, as studied by Bollobás (2001) and Penrose (2003). Furthermore, it is of interest to consider communication links with time-varying rates. •

3.1.3 The agree and pursue control and communication law

We conclude this section with an example of a dynamic control and communication law. The problem is described as follows: a collection of robots with range-limited communication are placed on the unit circle; the robots move and communicate with the objectives of (1) agreeing on a direction of motion (clockwise or counterclockwise) and (2) achieving an equidistant configuration where all robots are equally angularly spaced. To achieve these two objectives, we combine ideas from leader election algorithms for synchronous networks (see Section 1.5.4) and from cyclic pursuit problems (see Exercise E1.30): the robots move a distance proportional to an appropriate inter-robot separation, and they repeatedly compare their identifiers to discover the direction of motion of the robot with the largest identifier. In other words, the robots run a leader election task in their processor states and a uniform robotic deployment task in their physical state—these are among the most basic tasks in distributed algorithms and cooperative control. We present the algorithm here and characterize its correctness and performance later in the chapter.

From Example 3.7, we consider the uniform network $\mathcal{S}_{\text{circle}}$ of locally connected first-order robots on $\mathbb{S}^1$. For $r, u_{\max}, k_{\text{prop}} \in \,]0, \frac{1}{2}[$ with $k_{\text{prop}} r \leq u_{\max}$, we define the AGREE & PURSUE law, denoted by $\mathcal{CC}_{\text{AGREE \& PURSUE}}$, as the *uniform data-sampled* law loosely described as follows:

> *[Informal description]* The processor state consists of `dir` (the robot's direction of motion) taking values in $\{\texttt{c}, \texttt{cc}\}$ (meaning clockwise and counterclockwise) and `max-id` (the largest UID received by the robot, initially set to the robot's UID) taking values in I. In each communication round, each robot transmits its position and its processor state. Among the messages received from agents moving toward its position, each agent picks the message with the largest value of `max-id`. If this value is larger than its own value, the agent resets its processor state with the selected message. Between communication rounds, each robot moves in the clockwise or counterclockwise direction depending on whether its processor state `dir` is `c` or `cc`. Each robot moves k_{prop} times the distance to the immediately next neighbor in the chosen direction, or, if no neighbors are detected, k_{prop} times the communication range r.

Note that the processor state with the largest UID will propagate throughout the network as in the FLOODMAX ALGORITHM for leader election. Also, note that the assumption $k_{\text{prop}} r \leq u_{\max}$ guarantees that the desired control is always within the allowable range $[-u_{\max}, u_{\max}]$. Next, we define the law formally:

`Robotic Network:` $\mathcal{S}_{\text{circle}}$, first-order agents in $\mathbb{S}^1$
with absolute sensing of own position, and
with communication range r

`Distributed Algorithm:` AGREE & PURSUE

`Alphabet:` $\mathbb{A} = \mathbb{S}^1 \times \{\texttt{c}, \texttt{cc}\} \times I \cup \{\texttt{null}\}$

`Processor State`: $w = (\texttt{dir}, \texttt{max-id})$, where
  $\texttt{dir} \in \{\texttt{c}, \texttt{cc}\}$, initially: $\texttt{dir}^{[i]}$ unspecified
  $\texttt{max-id} \in I$, initially: $\texttt{max-id}^{[i]} = i$ for all i

% Standard message-generation function
`function` $\mathrm{msg}(\theta, w, i)$
  return (θ, w)

`function` $\mathrm{stf}(\theta, w, y)$
  for each non-`null` message $(\theta_{\mathrm{rcvd}}, (\texttt{dir}_{\mathrm{rcvd}}, \texttt{max-id}_{\mathrm{rcvd}}))$ in y **do**
    if $(\texttt{max-id}_{\mathrm{rcvd}} > \texttt{max-id})$ AND $(\mathrm{dist}_{\texttt{cc}}(\theta, \theta_{\mathrm{rcvd}}) \leq r$ AND $\texttt{dir}_{\mathrm{rcvd}} = \texttt{c})$ OR $(\mathrm{dist}_{\texttt{c}}(\theta, \theta_{\mathrm{rcvd}}) \leq r$ AND $\texttt{dir}_{\mathrm{rcvd}} = \texttt{cc})$ **then**
      $\texttt{new-dir} := \texttt{dir}_{\mathrm{rcvd}}$
      $\texttt{new-id} := \texttt{max-id}_{\mathrm{rcvd}}$
  return $(\texttt{new-dir}, \texttt{new-id})$

`function` $\mathrm{ctl}(\theta_{\mathrm{smpld}}, w, y)$
  $d_{\mathrm{tmp}} := r$
  for each non-`null` message $(\theta_{\mathrm{rcvd}}, (\texttt{dir}_{\mathrm{rcvd}}, \texttt{max-id}_{\mathrm{rcvd}}))$ in y **do**
    if $(\texttt{dir} = \texttt{cc})$ AND $(\mathrm{dist}_{\texttt{cc}}(\theta_{\mathrm{smpld}}, \theta_{\mathrm{rcvd}}) < d_{\mathrm{tmp}})$ **then**
      $d_{\mathrm{tmp}} := \mathrm{dist}_{\texttt{cc}}(\theta_{\mathrm{smpld}}, \theta_{\mathrm{rcvd}})$
      $u_{\mathrm{tmp}} := k_{\mathrm{prop}} d_{\mathrm{tmp}}$
    if $(\texttt{dir} = \texttt{c})$ AND $(\mathrm{dist}_{\texttt{c}}(\theta_{\mathrm{smpld}}, \theta_{\mathrm{rcvd}}) < d_{\mathrm{tmp}})$ **then**
      $d_{\mathrm{tmp}} := \mathrm{dist}_{\texttt{c}}(\theta_{\mathrm{smpld}}, \theta_{\mathrm{rcvd}})$
      $u_{\mathrm{tmp}} := -k_{\mathrm{prop}} d_{\mathrm{tmp}}$
  return u_{tmp}

An implementation of this control and communication law is shown in Figure 3.4. As parameters, we select $n = 45$, $r = 2\pi/40$, $u_{\max} = 1/4$ and $k_{\mathrm{prop}} = 7/16$. Along the evolution, all robots agree upon a common direction of motion and, after a suitable time, they reach a uniform distribution.

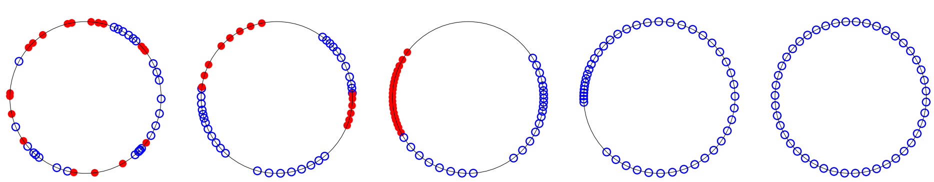


Figure 3.4 The AGREE & PURSUE law. Red-colored disks and blue-colored circles correspond to robots moving counterclockwise and clockwise, respectively. The initial positions and the initial directions of motion are randomly generated. The five diagrams depict the state of the network at times 0, 9, 20, 100, and 800.

3.2 ROBOTIC NETWORKS WITH RELATIVE SENSING

The model presented above assumes the ability of each robot to know its own absolute position. Here, we treat the alternative setting in which the robots do not communicate amongst themselves, but instead detect and measure each other's relative position through appropriate sensors. Additionally, we assume that the robots will perform measurements of the environment without having any *a priori* knowledge of it. We assume that robots do not have the ability to perform measurements expressed in a common reference frame. An early reference in which relative information is adopted is Lin et al. (2005).

3.2.1 Kinematics notions

Because the robots do not have a common reference frame, all the measurements generated by their on-board sensors are expressed in a local reference frame. To formalize this fact, it is useful to review some basic kinematics conventions. We let $\Sigma^{\text{fixed}} = (p^{\text{fixed}}, \{\boldsymbol{x}^{\text{fixed}}, \boldsymbol{y}^{\text{fixed}}, \boldsymbol{z}^{\text{fixed}}\})$ be a fixed reference frame in $\mathbb{R}^3$. A point q, a vector v, and a set of points S expressed with respect to the frame Σ^{fixed} are denoted by q_{fixed}, v_{fixed} and S_{fixed}, respectively. Next, let $\Sigma^{\text{b}} = (p^{\text{b}}, \{\boldsymbol{x}^{\text{b}}, \boldsymbol{y}^{\text{b}}, \boldsymbol{z}^{\text{b}}\})$ be a reference frame fixed to

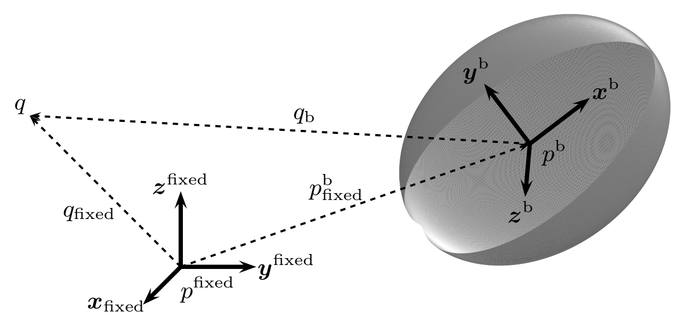


Figure 3.5 Inertially fixed and body-fixed frames in $\mathbb{R}^3$.

a moving body. The origin of Σ^{b} is the point p^{b}, denoted by $p^{\text{b}}_{\text{fixed}}$ when expressed with respect to Σ^{fixed}. The orientation of Σ^{b} is characterized by the d-dimensional rotation matrix $R^{\text{b}}_{\text{fixed}}$, whose columns are the frame vectors $\{\boldsymbol{x}^{\text{b}}, \boldsymbol{y}^{\text{b}}, \boldsymbol{z}^{\text{b}}\}$ of Σ^{b} expressed with respect to Σ^{fixed}. We recall here the definition of the group of rotation matrices in d-dimensions:

$$\text{SO}(d) = \{R \in \mathbb{R}^{d\times d} \mid RR^T = I_d,\ \det(R) = +1\}.$$

With these notations, changes of reference frames are described by

$$\begin{aligned} q_{\text{fixed}} &= R^{\text{b}}_{\text{fixed}} q_{\text{b}} + p^{\text{b}}_{\text{fixed}}, \\ v_{\text{fixed}} &= R^{\text{b}}_{\text{fixed}} v_{\text{b}}, \\ S_{\text{fixed}} &= R^{\text{b}}_{\text{fixed}} S_{\text{b}} + p^{\text{b}}_{\text{fixed}}. \end{aligned} \tag{3.2.1}$$

Note that these change-of-frames formulas also hold in the planar case with the corresponding definition of the rotation matrix in $\mathrm{SO}(2)$.

Remark 3.13 (Comparison with literature). In our notation, the subscript denotes the frame with respect to which the quantity is expressed. Other references in the literature sometimes adopt the opposite convention, in which the superscript denotes the frame with respect to which the quantity is expressed. •

3.2.2 Physical components

In what follows, we describe our notion of *mobile robots equipped with relative sensors*. We consider a group of n robots moving in an allowable environment $Q \subset \mathbb{R}^d$, for $d \in \{2, 3\}$, and we assume that a reference frame $\Sigma^{[i]}$, for $i \in \{1, \dots, n\}$, is attached to each robot (see Figure 3.6). Expressed with respect to the fixed frame Σ^{fixed}, the ith frame $\Sigma^{[i]}$ is described by a position $p^{[i]}_{\text{fixed}} \in \mathbb{R}^d$ and an orientation $R^{[i]}_{\text{fixed}} \in \mathrm{SO}(d)$. The continuous-time motion and discrete-time sensing models are described as follows.

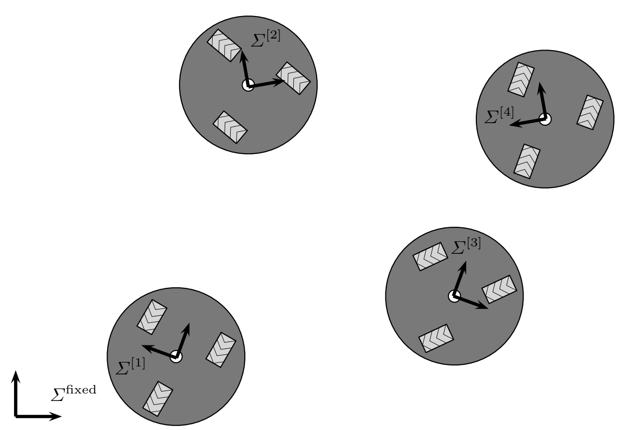


Figure 3.6 A robotic network with relative sensing. A group of four robots moves in $\mathbb{R}^2$. Each robot $i \in \{1, \dots, 4\}$ has its own reference frame $\Sigma^{[i]}$.

Motion model: We select a simple motion model: for all $t \in \mathbb{R}_{\geq 0}$, the orientation $R_{\text{fixed}}^{[i]}$ is constant in time and robot i translates according to

$$\dot{p}_{\text{fixed}}^{[i]}(t) = R_{\text{fixed}}^{[i]} u_i^{[i]}, \tag{3.2.2}$$

that is, the ith control input $u_i^{[i]}$ is known and applied in the robot frame. Each control input $u_i^{[i]}$, $i \in \{1, \dots, n\}$, takes values in a compact input space U. Clearly, it would be possible to consider a motion model with time-varying orientation and we refer the reader to Exercise E3.1, where we do so.

Sensing model: At each discrete time instant, robot i activates a sensor that detects the presence and returns a measurement about the relative position of any object (robots or environment boundary) inside a given "sensor footprint." We describe the model in two steps. First, each robot measures

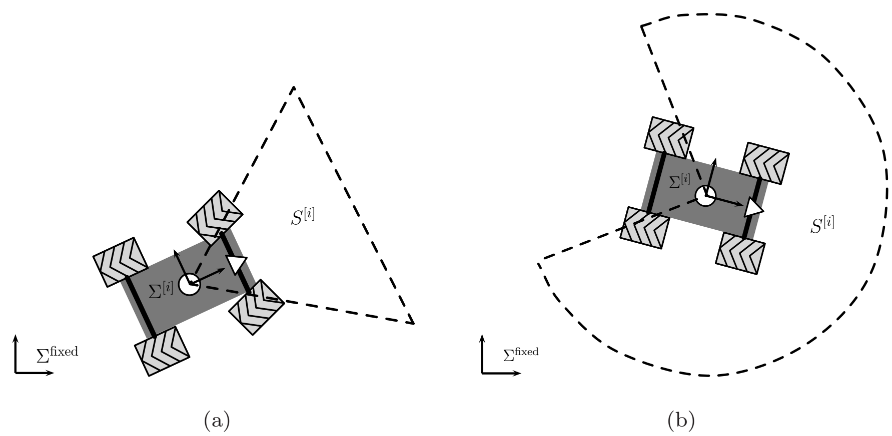


Figure 3.7 Examples of sensor footprints. (a) The cone-shaped sensor footprint of a vehicle equipped with a camera. (b) The 270-degree wedge-shaped sensor footprint of a vehicle equipped with a laser scanner.

other robots' positions and the environment as follows.

Sensing other robots' positions. There exists a set $\mathbb{A}_{\text{rbt}}$ containing the `null` element, called the *sensing alphabet*, and a map rbt-sns : $\mathbb{R}^d \to \mathbb{A}_{\text{rbt}}$, called the *sensing function*, with the interpretation that robot i acquires the symbol rbt-sns$(p_i^{[j]}) \in \mathbb{A}_{\text{rbt}}$ for each robot $j \in \{1, \dots, n\} \setminus \{i\}$.

Sensing the environment. There exists a set $\mathbb{A}_{\text{env}}$ containing the `null` element, called the *environment sensing alphabet*, and a map env-sns : $\mathbb{P}(\mathbb{R}^d) \to \mathbb{A}_{\text{env}}$, called the *environment sensing function*, with the interpretation that robot i acquires the symbol env-sns$(Q_i) \in \mathbb{A}_{\text{env}}$.

Second, we let $S^{[i]} \subset \mathbb{R}^d$ be the *sensor footprint* of robot i and we let $S_i^{[i]}$ be its expression in the frame $\Sigma^{[i]}$ (see Figure 3.7). For simplicity, we assume that all robot sensors are equal, so that we can write $S_i^{[i]} = S$. We require both sensing functions to provide no information about robots and boundaries that are outside S in the following two meanings: (i) if p is any point outside S, then $\text{rbt-sns}(p) = \texttt{null}$; and (ii) if W is any subset of $\mathbb{R}^d$, $\text{env-sns}(W) = \text{env-sns}(W \cap S)$.

We summarize this discussion with the following definition.

Definition 3.14 (Network with relative sensing). The physical components of a *network with relative sensing* consist of n mobile robots with identifiers $\{1, \dots, n\}$, with configurations in $Q \times \mathrm{SO}(d)$, for an allowable environment $Q \subset \mathbb{R}^d$, with dynamics described by equation (3.2.2), and with relative sensors described by the sensor footprint S, sensing alphabets $\mathbb{A}_{\text{rbt}}$ and $\mathbb{A}_{\text{env}}$, and sensing functions rbt-sns and env-sns. •

To make things concrete, let us present two examples of robotic networks with relative sensing that are analogs of the "communication-based" robotic networks $\mathcal{S}_{\text{disk}}$ and $\mathcal{S}_{\text{vis-disk}}$ in Examples 3.4 and 3.6.

Example 3.15 (Disk sensor and corresponding relative-sensing network). Given a *sensing range* $r \in \mathbb{R}_{>0}$, the *disk sensor* has sensor footprint $\overline{B}(\mathbf{0}_d, r)$, that is, a disk sensor measures any object (robot and environment boundary) within distance r. Regarding sensing of other robots, we assume that the alphabet is $\mathbb{A}_{\text{rbt}} = \mathbb{R}^d \cup \{\texttt{null}\}$ and that the sensing function is $\text{rbt-sns}(p_i^{[j]}) = p_i^{[j]}$ for each robot $j \in \{1, \dots, n\} \setminus \{i\}$, inside the sensor footprint $\overline{B}(\mathbf{0}_d, r)$, and $\text{rbt-sns}(p_i^{[j]}) = \texttt{null}$, otherwise. Regarding sensing of the environment, we assume that the alphabet is $\mathbb{A}_{\text{env}} = \mathbb{P}(\mathbb{R}^d)$ and that the sensing function is $\text{env-sns}(Q_i) = Q_i \cap \overline{B}(\mathbf{0}_d, r)$. A group of robots with disk sensors defines the robotic network with relative sensing $\mathcal{S}_{\text{disk}}^{\text{rs}}$. •

Example 3.16 (Range-limited visibility sensor and corresponding relative-sensing network). Given a *sensing range* $r \in \mathbb{R}_{>0}$, the *range-limited visibility sensor* has sensor footprint $\overline{B}(\mathbf{0}_d, r)$ and performs measurements only of objects within unobstructed line of sight. Regarding sensing of other robots, we assume that the alphabet is $\mathbb{A}_{\text{rbt}} = \mathbb{R}^d \cup \{\texttt{null}\}$ and that the sensing function is $\text{rbt-sns}(p_i^{[j]}) = p_i^{[j]}$ for each robot $j \in \{1, \dots, n\} \setminus \{i\}$, inside the range-limited visibility set $\mathrm{Vi}_{\text{disk}}(\mathbf{0}_2; Q_i)$, and $\text{rbt-sns}(p_i^{[j]}) = \texttt{null}$, otherwise. Regarding sensing of the environment, we assume[1] that the alphabet is $\mathbb{A}_{\text{env}} = \mathbb{P}(\mathbb{R}^d)$ and that the environment sensor measures the range-

[1] It would be equivalent to assume that the robot can sense every portion of ∂Q that is within distance r and that is visible from the robot's position.

limited visibility set $\mathrm{Vi}_{\mathrm{disk}}(p^{[i]}_{\mathrm{fixed}}; Q)$ expressed with respect to the frame $\Sigma^{[i]}$; for the definition of range-limited visibility set, see Section 2.1.2. In other words, the environment sensing function is $\text{env-sns}(Q_i) = \mathrm{Vi}_{\mathrm{disk}}(\mathbf{0}_2; Q_i)$. This is illustrated in Figure 3.8. A group of robots with range-limited visi-

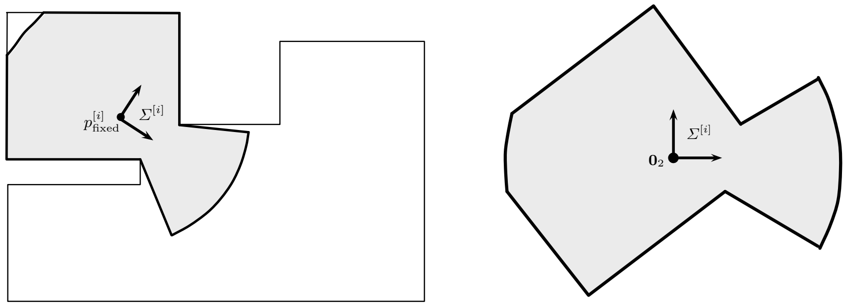


Figure 3.8 The left-hand plot depicts the range-limited visibility set $\mathrm{Vi}_{\mathrm{disk}}(p^{[i]}_{\mathrm{fixed}}; Q)$ expressed with respect to an inertially fixed frame. The right-hand plot depicts the range-limited visibility set expressed with respect to the body-fixed frame $\Sigma^{[i]}$, that is, $\mathrm{Vi}_{\mathrm{disk}}(\mathbf{0}_2; Q_i)$.

bility sensors defines the robotic network with relative sensing $\mathcal{S}^{\mathrm{rs}}_{\text{vis-disk}}$. •

Remark 3.17 (Sensing model consequences). The proposed sensing model has the following two consequences:

(i) Robots have no information about the absolute position and orientation of themselves, the other robots or any part of the environment.

(ii) The relative sensing capacity of the robots gives rise to a proximity graph, called the *sensing graph*, whose edges are the collection of robot pairs that are within sensing range. For example, in the network $\mathcal{S}^{\mathrm{rs}}_{\mathrm{disk}}$, the sensing graph is the disk graph $\mathcal{G}_{\mathrm{disk}}(r)$. In general, sensing graphs are directed. •

3.2.3 Relative-sensing control laws

As we did for robotic networks with interactions based on communication, we define here control laws based on relative sensing and we describe the closed-loop evolution of robotic networks with relative sensing.

First, we consider a robotic network with relative sensing $\mathcal{S}^{\mathrm{rs}}$ characterized by: identifiers $\{1, \dots, n\}$, configurations in $Q \times \mathrm{SO}(d)$, for an allowable environment $Q \subset \mathbb{R}^d$, dynamics described by equation (3.2.2), and relative sensors described by the sensor footprint S, sensing alphabets $\mathbb{A}_{\mathrm{rbt}}$ and

$\mathbb{A}_{\text{env}}$, and sensing functions rbt-sns and env-sns. A *relative-sensing control law* $\mathcal{RSC}$ for the robotic network with relative sensing $\mathcal{S}^{\text{rs}}$ consists of the following tuple:

(i) W, called the *processor state set*, with a corresponding set of *allowable initial values* $W_0 \subseteq W$;

(ii) $\text{stf} : W \times \mathbb{A}_{\text{rbt}}^n \times \mathbb{A}_{\text{env}} \to W$, called the *(processor) state-transition function*; and

(iii) $\text{ctl} : W \times \mathbb{A}_{\text{rbt}}^n \times \mathbb{A}_{\text{env}} \to U$, called the *(motion) control function*.

As for robotic networks, we say that $\mathcal{RSC}$ is *static* if W is a singleton for all $i \in \{1, \dots, n\}$; in this case, $\mathcal{RSC}$ can be described by a motion control function $\text{ctl} : \mathbb{A}_{\text{rbt}}^n \times \mathbb{A}_{\text{env}} \to U$. Additionally, if the environment $Q = \mathbb{R}^d$, then $\mathcal{RSC}$ can be described by a motion control function $\text{ctl} : W \times \mathbb{A}_{\text{rbt}}^n \to U$.

Second, the *evolution* of $(\mathcal{S}^{\text{rs}}, \mathcal{RSC})$ from initial conditions $(p_0^{[i]}, R_{\text{fixed}}^{[i]}) \in \mathbb{R}^d \times \text{SO}(d)$ and $w_0^{[i]} \in W_0$, $i \in \{1, \dots, n\}$ is the collection of curves $p_{\text{fixed}}^{[i]} : \mathbb{R}_{\geq 0} \to \mathbb{R}^d$ and $w^{[i]} : \mathbb{Z}_{\geq 0} \to W$, $i \in \{1, \dots, n\}$, defined by

$$\begin{aligned} \dot{p}_{\text{fixed}}^{[i]}(t) &= R_{\text{fixed}}^{[i]} \, \text{ctl}\big(w^{[i]}(\lfloor t \rfloor), y^{[i]}(\lfloor t \rfloor), y_{\text{env}}^{[i]}(\lfloor t \rfloor)\big), \\ w^{[i]}(\ell) &= \text{stf}(w^{[i]}(\ell - 1), y^{[i]}(\ell), y_{\text{env}}^{[i]}(\ell)), \end{aligned}$$

with $p_{\text{fixed}}^{[i]}(0) = p_0^{[i]}$ and $w^{[i]}(-1) = w_0^{[i]}$, $i \in \{1, \dots, n\}$. In the previous equations, $y^{[i]} : \mathbb{Z}_{\geq 0} \to \mathbb{A}_{\text{rbt}}^n$ (describing the robot measurements taken by sensor i) with components $y_j^{[i]}(\ell)$, for $j \in \{1, \dots, n\}$, and $y_{\text{env}}^{[i]} : \mathbb{Z}_{\geq 0} \to \mathbb{A}_{\text{env}}$ (describing the environment measurements taken by sensor i) are defined by

$$y_j^{[i]}(\ell) = \text{rbt-sns}(p_i^{[j]}(\ell)), \quad y_{\text{env}}^{[i]}(\ell) = \text{env-sns}(Q_i(\ell)).$$

In the last equation, $p_i^{[j]}$ and $Q_i(\ell)$ denote the position of the j-th robot and the environment Q as expressed with respect to the moving frame $\Sigma^{[i]}$.

3.2.4 Equivalence between control and communication laws and relative-sensing control laws

Consider a "communication-based" robotic network $\mathcal{S}_1$ with a control and communication law $\mathcal{CC}_1$ with the following properties:

(i) Regarding $\mathcal{S}_1$: the network is uniform, the state space is $X = \mathbb{R}^d$ with states denoted by $x^{[i]} = p^{[i]}$, the communication graph is the r-disk graph, and the robot dynamics are $\dot{p}^{[i]} = u^{[i]}$.

(ii) Regarding $\mathcal{CC}_1$: the control and communication law is uniform and data-sampled, the communication alphabet is $\mathbb{A} = \mathbb{R}^d \cup \{\texttt{null}\}$, and the message-generation function is $\text{msg}(p, w, j) = p$.

Given a network and a law $(\mathcal{S}_1, \mathcal{CC}_1)$ satisfying (i) and (ii), the control and communication law $\mathcal{CC}_1$ is *invariant* if its state transition and control maps satisfy, for all $p \in \mathbb{R}^d$, $w \in W$, $y \in \mathbb{A}^n$, and $R \in \text{SO}(d)$,

$$\begin{aligned}\text{stf}(p, w, y) &= \text{stf}\big(\mathbf{0}_d, w, R(y - p)\big),\\ \text{ctl}(p, w, y) &= R^T \text{ctl}\big(\mathbf{0}_d, w, R(y - p)\big),\end{aligned}$$

where the ith component of $R(y - p) \in \mathbb{A}^n$ is $R(y_i - p)$ if $y_i \in \mathbb{R}^d$, or $\texttt{null}$ if $y_i = \texttt{null}$.

Next, consider a relative-sensing network $\mathcal{S}_2$ with disk sensors as in Example 3.15, that is, assume that the sensing footprint is $\overline{B}(\mathbf{0}_d, r)$, the sensing alphabet is $\mathbb{A}_{\text{rbt}} = \mathbb{R}^d \cup \{\texttt{null}\}$, and the sensing function equals the identity function in $\overline{B}(\mathbf{0}_d, r)$. We assume no environment sensing as we set $Q = \mathbb{R}^d$. The communication and control law $\mathcal{CC}_1$ and the relative-sensing control law $\mathcal{RSC}_2$ for network $\mathcal{S}_2$ are *equivalent* if their processor state sets identical, for example, denoting both by W, and their state transition and control maps satisfy, for all $w \in W$ and $y \in \mathbb{R}^d \cup \{\texttt{null}\} = \mathbb{A}^n = \mathbb{A}^n_{\text{rbt}}$,

$$\text{stf}_1(\mathbf{0}_d, w, y) = \text{stf}_2(w, y), \qquad \text{and} \qquad \text{ctl}_1(\mathbf{0}_d, w, y) = \text{ctl}_2(w, y).$$

Proposition 3.18 (Evolution equivalence). *If $\mathcal{CC}_1$ is invariant and if $\mathcal{CC}_1$ and $\mathcal{RSC}_2$ are equivalent, then the evolutions of the control and communication laws $(\mathcal{S}_1, \mathcal{CC}_1)$ and $(\mathcal{S}_2, \mathcal{RSC}_2)$ from identical initial conditions are identical.*

Proof. Assume that the messages and measurements array $y^{[i]}(t)$ received by the i-th robot at time t in the communication-based network and in the relative-sensing networks are equal to, respectively:

$$p^{[j_1]}_{\text{fixed}}, \dots, p^{[j_k]}_{\text{fixed}}, \qquad \text{and} \qquad p^{[j_1]}_i, \dots, p^{[j_k]}_i.$$

Under this assumption, the evolutions of the communication-based network and of the relative-sensing networks are written, respectively, as,

$$\begin{aligned}\dot{p}^{[i]}_{\text{fixed}} &= \text{ctl}_1(p^{[i]}_{\text{fixed}}, w^{[i]}, p^{[j_1]}_{\text{fixed}}, \dots, p^{[j_k]}_{\text{fixed}}),\\ \dot{p}^{[i]}_{\text{fixed}} &= R^{[i]}_{\text{fixed}} \text{ctl}_2(w^{[i]}, p^{[j_1]}_i, \dots, p^{[j_k]}_i).\end{aligned}$$

From equation (3.2.1), we know that, for all $j \in \{j_1, \dots, j_k\}$,

$$p^{[j]}_{\text{fixed}} = R^{[i]}_{\text{fixed}} p^{[j]}_i + p^{[i]}_{\text{fixed}} \qquad \Longrightarrow \qquad p^{[j]}_i = (R^{[i]}_{\text{fixed}})^T (p^{[j]}_{\text{fixed}} - p^{[i]}_{\text{fixed}}).$$

From this equality and from the fact that $\mathcal{CC}_1$ is invariant, we observe that

$$\mathrm{ctl}_1(p^{[i]}_{\mathrm{fixed}}, w^{[i]}, p^{[j_1]}_{\mathrm{fixed}}, \dots, p^{[j_k]}_{\mathrm{fixed}}) = R^{[i]}_{\mathrm{fixed}} \mathrm{ctl}_1(\mathbf{0}_d, w^{[i]}, p^{[j_1]}_i, \dots, p^{[j_k]}_i).$$

Since $\mathcal{CC}_1$ and $\mathcal{RSC}_2$ are equivalent, the two evolution equations coincide. A similar reasoning also shows that the evolutions of the processor states are identical. ■

Remark 3.19 (Communication-based laws on relative-sensing networks). Proposition 3.18 implies the following fact. An invariant control and communication law for a robotic network satisfying appropriate properties can be implemented on an appropriate relative-sensing network as a relative-sensing control law. •

3.3 COORDINATION TASKS AND COMPLEXITY NOTIONS

In this section, we introduce concepts and tools that are useful analyzing a communication and control law in a robotic network; our treatment is directly generalized to relative-sensing networks. We address the following questions: What is a coordination task for a robotic network? When does a control and communication law achieve a task? And with what time, space, and communication complexity?

3.3.1 Coordination tasks

Our first analysis step is to characterize the correctness properties of a communication and control law. We do so by defining the notions of task and of task achievement by a robotic network.

Definition 3.20 (Coordination task). Let $\mathcal{S}$ be a robotic network and let $\mathcal{W}$ be a set.

(i) A *coordination task* is a map $\mathcal{T}\colon \prod_{i\in I} X^{[i]} \times \mathcal{W}^n \to \{\texttt{true}, \texttt{false}\}$.

(ii) If $\mathcal{W}$ is a singleton, then the coordination task is said to be *static* and can be described by a map $\mathcal{T}\colon \prod_{i\in I} X^{[i]} \to \{\texttt{true}, \texttt{false}\}$.

Additionally, let $\mathcal{CC}$ be a control and communication law for $\mathcal{S}$:

(i) The law $\mathcal{CC}$ is *compatible* with the task $\mathcal{T} : \prod_{i\in I} X^{[i]} \times \mathcal{W}^n \to \{\texttt{true}, \texttt{false}\}$ if its processor state takes values in $\mathcal{W}$, that is, if $W^{[i]} = \mathcal{W}$, for all $i \in I$.

(ii) The law $\mathcal{CC}$ *achieves* the task $\mathcal{T}$ if it is compatible with it and if, for all initial conditions $x^{[i]}_0 \in X^{[i]}_0$ and $w^{[i]}_0 \in W^{[i]}_0$, $i \in I$, there exists

$T \in \mathbb{R}_{>0}$ such that the network evolution $t \mapsto (x(t), w(t))$ has the property that $\mathcal{T}(x(t), w(t)) = \texttt{true}$ for all $t \geq T$. •

Remark 3.21 (Temporal logic). Loosely speaking, the phrase "a law achieves a task" means that the network evolutions reach (and remain at) a specified pattern in the robot physical or processor state. In other words, the task is achieved if *at some time* and *for all subsequent times*, the predicate evaluates to `true` along system trajectories. It is possible to consider more general tasks based on more expressive predicates on trajectories. Such predicates can be defined through various forms of temporal and propositional logic, (see, e.g., Emerson, 1994). In particular, (linear) temporal logic contains certain constructs that allow reasoning in terms of time and is hence appropriate for robotic applications—as argued, for example, by Fainekos et al. (2005). Network tasks such as periodically visiting a desired set of configurations can be encoded with temporal logic statements. •

Example 3.22 (Direction agreement and equidistance tasks). From Example 3.7, consider the uniform network $\mathcal{S}_{\text{circle}}$ of locally connected first-order agents in $\mathbb{S}^1$. From Section 3.1.3, recall the AGREE & PURSUE control and communication law $\mathcal{CC}_{\text{AGREE \& PURSUE}}$ with processor state taking values in $W = \{\texttt{cc}, \texttt{c}\} \times I$. There are two tasks of interest. First, we define the *direction agreement task* $\mathcal{T}_{\texttt{dir}} : (\mathbb{S}^1)^n \times W^n \to \{\texttt{true}, \texttt{false}\}$ by

$$\mathcal{T}_{\texttt{dir}}(\theta, w) = \begin{cases} \texttt{true}, & \text{if } \texttt{dir}^{[1]} = \dots = \texttt{dir}^{[n]}, \\ \texttt{false}, & \text{otherwise}, \end{cases}$$

where $\theta = (\theta^{[1]}, \dots, \theta^{[n]})$, $w = (w^{[1]}, \dots, w^{[n]})$, and $w^{[i]} = (\texttt{dir}^{[i]}, \texttt{max-id}^{[i]})$, for $i \in I$. Furthermore, for $\varepsilon > 0$, we define the static *(agent) equidistance task* $\mathcal{T}_{\varepsilon\text{-eqdstnc}} : (\mathbb{S}^1)^n \to \{\texttt{true}, \texttt{false}\}$ to be `true` if and only if

$$\big| \min_{j \neq i} \operatorname{dist}_{\texttt{c}}(\theta^{[i]}, \theta^{[j]}) - \min_{j \neq i} \operatorname{dist}_{\texttt{cc}}(\theta^{[i]}, \theta^{[j]}) \big| < \varepsilon, \quad \text{for all } i \in I.$$

In other words, $\mathcal{T}_{\varepsilon\text{-eqdstnc}}$ is true when, for every agent, the distances to the closest clockwise neighbor and to the closest counterclockwise neighbor are approximately equal. •

3.3.2 Complexity notions

We are now ready to define the notions of time, space and communication complexity. These notions describe the cost that a certain control and communication law incurs while completing a certain coordination task. Additionally, the complexity of a task is the infimum of the costs incurred by all laws that achieve that task. We begin by highlighting a difference between what follows and the complexity treatment for synchronous networks.

Remark 3.23 (Termination via task completion). As discussed in Remark 1.44 in Section 1.5, it is possible to consider various algorithm termination notions. Here, we will establish the completion of an algorithm as the instant when a given task is achieved. •

First, we define the time complexity of an achievable task as the minimum number of communication rounds needed by the agents to achieve the task $\mathcal{T}$.

Definition 3.24 (Time complexity). Let $\mathcal{S}$ be a robotic network and let $\mathcal{T}$ be a coordination task for $\mathcal{S}$. Let $\mathcal{CC}$ be a control and communication law for $\mathcal{S}$ compatible with $\mathcal{T}$:

(i) the *(worst-case) time complexity to achieve $\mathcal{T}$ with $\mathcal{CC}$ from initial conditions* $(x_0, w_0) \in \prod_{i\in I} X_0^{[i]} \times \prod_{i\in I} W_0^{[i]}$ is

$$\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0, w_0) = \inf \{\ell \mid \mathcal{T}(x(k), w(k)) = \texttt{true}\,,\ \text{for all } k \geq \ell\},$$

where $t \mapsto (x(t), w(t))$ is the evolution of $(\mathcal{S}, \mathcal{CC})$ from the initial condition (x_0, w_0);

(ii) the *(worst-case) time complexity to achieve $\mathcal{T}$ with $\mathcal{CC}$* is

$$\mathrm{TC}(\mathcal{T}, \mathcal{CC}) = \sup \Big\{ \mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0, w_0) \mid (x_0, w_0) \in \prod_{i\in I} X_0^{[i]} \times \prod_{i\in I} W_0^{[i]} \Big\};$$

(iii) the *(worst-case) time complexity of $\mathcal{T}$* is

$$\mathrm{TC}(\mathcal{T}) = \inf\{\mathrm{TC}(\mathcal{T}, \mathcal{CC}) \mid \mathcal{CC} \text{ compatible with } \mathcal{T}\}. \qquad \bullet$$

Next, we quantify memory and communication requirements of communication and control laws. We assume that elements of the processor state set W or of the alphabet set $\mathbb{A}$ might amount to multiple "basic memory units" or "basic messages." We let $|W|_{\text{basic}}$ and $|\mathbb{A}|_{\text{basic}}$ denote the number of basic memory units and basic messages required to represent elements of W and $\mathbb{A}$, respectively. The `null` message has zero cost. To clarify this assumption, we adopt two conventions. First, as in Section 1.5.2, we assume that a "basic memory unit" or a "basic message" contains $\log(n)$ bits. This implies that the $\log(n)$ bits required to store or transmit a robot identifier $i \in \{1, \dots, n\}$ are equivalent to one "basic memory unit." Second, as mentioned in Remark 3.11, we assume that a processor can store and transmit a (finite number of) integer and real numbers, and we adopt the convention that any such number is quantized and represented by a constant number of basic memory units or basic messages.

We now quantify memory requirements of algorithms and tasks by counting the required number of basic memory units. Let the network $\mathcal{S}$, the task $\mathcal{T}$, and the control and communication law $\mathcal{CC}$ be as in Definition 3.24.

Definition 3.25 (Space complexity).

(i) The *(worst-case) space complexity to achieve $\mathcal{T}$ with $\mathcal{CC}$*, denoted by $\mathrm{SC}(\mathcal{T}, \mathcal{CC})$, is the maximum number of basic memory units required by a robot processor executing the $\mathcal{CC}$ on $\mathcal{S}$ among all robots and among all allowable initial physical and processor states until termination; and

(ii) the space complexity of $\mathcal{T}$ is the infimum among the space complexities of all control and communication laws that achieve $\mathcal{T}$. •

The set of all non-null messages generated during one communication round from network state (x, w) is denoted by

$$\mathcal{M}(x, w) = \{(i, j) \in E_{\mathrm{cmm}}(x) \mid \mathrm{msg}^{[i]}(x^{[i]}, w^{[i]}, j) \neq \texttt{null}\}.$$

We now quantify the mean and total communication requirements of algorithms and tasks by counting the number of transmitted basic messages.

Definition 3.26 (Mean and Total Communication complexity).

(i) The *(worst-case) mean communication complexity* and the *(worst-case) total communication complexity* to achieve $\mathcal{T}$ with $\mathcal{CC}$ from $(x_0, w_0) \in \prod_{i \in I} X_0^{[i]} \times \prod_{i \in I} W_0^{[i]}$ are, respectively,

$$\mathrm{MCC}(\mathcal{T}, \mathcal{CC}, x_0, w_0) = \frac{|\mathbb{A}|_{\mathrm{basic}}}{\tau} \sum_{\ell=0}^{\tau-1} |\mathcal{M}(x(\ell), w(\ell))|,$$

$$\mathrm{TCC}(\mathcal{T}, \mathcal{CC}, x_0, w_0) = |\mathbb{A}|_{\mathrm{basic}} \sum_{\ell=0}^{\tau-1} |\mathcal{M}(x(\ell), w(\ell))|,$$

where $t \mapsto (x(t), w(t))$ is the evolution of $(\mathcal{S}, \mathcal{CC})$ from the initial condition (x_0, w_0) and where $\tau = \mathrm{TC}(\mathcal{CC}, \mathcal{T}, x_0, w_0)$. Here, MCC is defined only for initial conditions (x_0, w_0) with the property that $\mathcal{T}(x_0, w_0) = \texttt{false}$;

(ii) the *(worst-case) mean communication complexity* (resp. the *(worst-case) total communication complexity*) to achieve $\mathcal{T}$ with $\mathcal{CC}$ is the supremum of $\mathrm{MCC}(\mathcal{T}, \mathcal{CC}, x_0, w_0)$ (resp. $\mathrm{TCC}(\mathcal{T}, \mathcal{CC}, x_0, w_0)$) over all allowable initial states (x_0, w_0); and

(iii) the *(worst-case) mean communication complexity* (resp. the *worst-case total communication complexity*) of $\mathcal{T}$ is the infimum among the mean communication complexity (resp. the total communication complexity) of all control and communication laws achieving $\mathcal{T}$. •

By construction, one can verify that it always happens that

$$\mathrm{TCC}(\mathcal{T},\mathcal{CC}) \leq \mathrm{MCC}(\mathcal{T},\mathcal{CC}) \cdot \mathrm{TC}(\mathcal{T},\mathcal{CC}). \tag{3.3.1}$$

We conclude this section with possible variations and extensions of the complexity definitions.

Remark 3.27 (Infinite-horizon mean communication complexity). The mean communication complexity MCC measures the average cost of the communication rounds required to achieve a task over a finite time horizon; a similar statement holds for the total communication complexity TCC. One might be interested in a notion of mean communication complexity required to maintain the task `true` for all times. Accordingly, the infinite-horizon mean communication complexity of $\mathcal{CC}$ from initial conditions (x_0, w_0) is

$$\text{IH-MCC}(\mathcal{CC}, x_0, w_0) = \lim_{\tau \to +\infty} \frac{|\mathbb{A}|_{\text{basic}}}{\tau} \sum_{\ell=0}^{\tau} |\mathcal{M}(x(\ell), w(\ell))|.$$ •

Remark 3.28 (Communication complexity in omnidirectional networks). In omnidirectional wireless networks, the standard operation mode is for all neighbors of a node to receive the signal that it transmits. In other words, the transmission is omnidirectional rather than unidirectional. It is straightforward to require the message-generation function to have the property that the output it generates be independent of the intended receiver. Under such assumptions, it make sense to count as communication complexity not the number of messages transmitted in the network, but the number of transmissions, that is, a unit cost per node rather than a unit cost per edge of the network. •

Remark 3.29 (Energy complexity). Given a model for the energy consumed by the robot to move and to transmit a message, one can easily define a notion of energy complexity for a control and communication law. In modern wireless transmitters, the energy consumption in transmitting a signal at a distance r varies with a power of r. Analogously, energy consumption is an increasing function of distance traveled. We consider this to be a promising avenue for further research. •

3.3.3 Invariance under rescheduling

Here we discuss the invariance properties of time and communication complexity under the *rescheduling* of a control and communication law. The idea behind rescheduling is to "spread" the execution of the law over time without affecting the trajectories described by the robots.

For simplicity we consider the setting of static laws; similar results can be obtained for the general setting. Also, for ease of presentation, we allow our communication and control laws to be time dependent, that is, we consider message-generation functions and motion control functions of the form $\text{msg}^{[i]} : \mathbb{Z}_{\geq 0} \times X^{[i]} \times I \to \mathbb{A}$ and $\text{ctl}^{[i]} : \mathbb{R}_{\geq 0} \times X^{[i]} \times X^{[i]} \times \mathbb{A}^n \to U^{[i]}$, respectively. Definition 3.10 for network evolution can be readily extended to this more general time-dependent setup.

Let $\mathcal{S} = (I, \mathcal{R}, E_{\text{cmm}})$ be a robotic network in which each robot is a driftless control system (see Section 1.3). Let $\mathcal{CC} = (\mathbb{A}, \{\text{msg}^{[i]}\}_{i \in I}, \{\text{ctl}^{[i]}\}_{i \in I})$ be a static control and communication law. In what follows, we define a new control and communication law by modifying $\mathcal{CC}$; to do so, we introduce some notation. Let $s \in \mathbb{N}$, with $s \leq n$, and let $\mathcal{P}_I = \{I_0, \dots, I_{s-1}\}$ be an *s-partition* of I, that is, $I_0, \dots, I_{s-1} \subset I$ are disjoint and nonempty and $I = \cup_{k=0}^{s-1} I_k$. For $i \in I$, define the message-generation functions $\text{msg}^{[i]}_{\mathcal{P}_I} : \mathbb{Z}_{\geq 0} \times X^{[i]} \times I \to \mathbb{A}$ by

$$\text{msg}^{[i]}_{\mathcal{P}_I}(\ell, x, j) = \text{msg}^{[i]}(\lfloor \ell/s \rfloor, x, j), \tag{3.3.2}$$

if $i \in I_k$ and $k = \ell \mod s$, and $\text{msg}^{[i]}_{\mathcal{P}_I}(\ell, x, j) = \texttt{null}$ otherwise. According to this message-generation function, only the agents with a unique identifier in I_k will send messages at time ℓ, where $\ell \in \{k + as\}_{a \in \mathbb{Z}_{\geq 0}}$. Equivalently, this can be stated as follows: according to (3.3.2), the messages originally sent at the time instant ℓ are now rescheduled to be sent at the time instants $F(\ell) - s + 1, \dots, F(\ell)$, where $F : \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$ is defined by $F(\ell) = s(\ell+1) - 1$. Figure 3.9 illustrates this idea. For $i \in I$, define the control functions

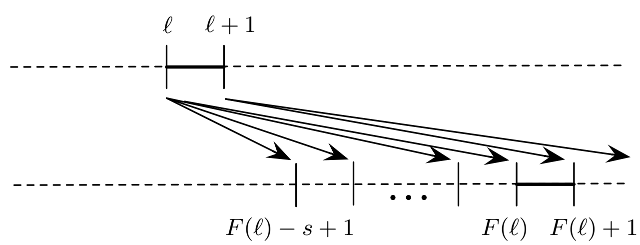


Figure 3.9 Under the rescheduling, the messages that are sent at the time instant ℓ under the control and communication law $\mathcal{CC}$ are rescheduled to be sent over the time instants $F(\ell) - s + 1, \dots, F(\ell)$ under the control and communication law $\mathcal{CC}_{(s, \mathcal{P}_I)}$.

$\text{ctl}^{[i]} : \mathbb{R}_{\geq 0} \times X^{[i]} \times X^{[i]} \times \mathbb{A}^n \to U^{[i]}$ by

$$\text{ctl}^{[i]}_{\mathcal{P}_I}(t, x, x_{\text{smpld}}, y) = \text{ctl}^{[i]}\left(t - \ell + F^{-1}(\ell), x, x_{\text{smpld}}, y\right), \tag{3.3.3}$$

if $t \in [\ell, \ell+1]$ and $\ell = -1 \mod s$, and $\text{ctl}^{[i]}_{\mathcal{P}_I}(t, x, x_{\text{smpld}}, y) = 0$ otherwise.

Here, $F^{-1} : \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$ is the inverse of F, defined by $F^{-1}(\ell) = \frac{\ell+1}{s} - 1$. Roughly speaking, the control law $\mathrm{ctl}^{[i]}_{\mathcal{P}_I}$ makes the agent i wait for the time intervals $[\ell, \ell+1]$, with $\ell \in \{as-1\}_{a\in\mathbb{N}}$, to execute any motion. Accordingly, the evolution of the robotic network under the original law $\mathcal{CC}$ during the time interval $[\ell, \ell+1]$ now takes place when all the corresponding messages have been transmitted, that is, along the time interval $[F(\ell), F(\ell)+1]$. The following definition summarizes this construction.

Definition 3.30 (Rescheduling of control and communication laws). Let $\mathcal{S} = (I, \mathcal{R}, E_{\mathrm{cmm}})$ be a robotic network with driftless physical agents, and let $\mathcal{CC} = (\mathbb{Z}_{\geq 0}, \mathbb{A}, \{\mathrm{msg}^{[i]}\}_{i\in I}, \{\mathrm{ctl}^{[i]}\}_{i\in I})$ be a static control and communication law. Let $s \in \mathbb{N}$, with $s \leq n$, and let $\mathcal{P}_I$ be an s-partition of I. The control and communication law $\mathcal{CC}_{(s,\mathcal{P}_I)} = (\mathbb{Z}_{\geq 0}, \mathbb{A}, \{\mathrm{msg}^{[i]}_{\mathcal{P}_I}\}_{i\in I}, \{\mathrm{ctl}^{[i]}_{\mathcal{P}_I}\}_{i\in I})$ defined by equations (3.3.2) and (3.3.3) is called a *$\mathcal{P}_I$-rescheduling of $\mathcal{CC}$.* •

The following result, whose proof is presented in Section 3.6.1, shows that the total communication complexity is invariant under rescheduling.

Proposition 3.31 (Complexity of rescheduled laws). *With the assumptions of Definition 3.30, let $\mathcal{T} : \prod_{i\in I} X^{[i]} \to \{\mathtt{true}, \mathtt{false}\}$ be a coordination task for $\mathcal{S}$. Then, for all $x_0 \in \prod_{i\in I} X_0^{[i]}$,*

$$\mathrm{TC}(\mathcal{T}, \mathcal{CC}_{(s,\mathcal{P}_I)}, x_0) = s \cdot \mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0)\,.$$

Moreover, if $\mathrm{C}_{\mathrm{rnd}}$ is additive, then, for all $x_0 \in \prod_{i\in I} X_0^{[i]}$,

$$\mathrm{MCC}(\mathcal{T}, \mathcal{CC}_{(s,\mathcal{P}_I)}, x_0) = \frac{1}{s} \cdot \mathrm{MCC}(\mathcal{T}, \mathcal{CC}, x_0)\,,$$

and therefore, $\mathrm{TCC}(\mathcal{T}, \mathcal{CC}_{(s,\mathcal{P}_I)}, x_0) = \mathrm{TCC}(\mathcal{T}, \mathcal{CC}, x_0)$, that is, the total communication complexity of $\mathcal{CC}$ is invariant under rescheduling.

Remark 3.32 (Appropriate complexity notions for driftless agents). Given the results in the previous theorem, one should be careful in choosing which notion of communication complexity to use in order to evaluate control and communication laws. For driftless physical agents, rather than the *mean* communication complexity MCC, one should really consider the *total* communication complexity TCC, since the latter is invariant with respect to rescheduling. Note that the notion of infinite-horizon mean communication complexity IH-MCC defined in Remark 3.27 satisfies the same relationship as MCC, that is, $\text{IH-MCC}(\mathcal{CC}_{(s,\mathcal{P}_I)}, x_0) = \frac{1}{s}\,\text{IH-MCC}(\mathcal{CC}, x_0)$. •

3.4 COMPLEXITY OF DIRECTION AGREEMENT AND EQUIDISTANCE

From Example 3.7, Section 3.1.3, and Example 3.22, recall the definition of a uniform network $\mathcal{S}_{\text{circle}}$ of locally connected first-order agents in $\mathbb{S}^1$, the AGREE & PURSUE control and communication law $\mathcal{CC}_{\text{AGREE \& PURSUE}}$, and the two coordination tasks $\mathcal{T}_{\texttt{dir}}$ and $\mathcal{T}_{\varepsilon\text{-eqdstnc}}$. In this section, we characterize the complexity to achieve these coordination tasks with $\mathcal{CC}_{\text{AGREE \& PURSUE}}$. Because the number of bits required to represent the variable $\texttt{max-id} \in \{1,\dots,n\}$ is $\log(n)$, note that the space complexity of $\mathcal{CC}_{\text{AGREE \& PURSUE}}$ is $\log(n)$ bits, that is, one basic memory unit in our convention discussed in Section 3.3.2.

Motivated by Remark 3.8, we model wireless communication congestion by assuming that the communication range is a monotone non-increasing function $r : \mathbb{N} \to]0,\pi[$ of the number of agents n. Likewise, we assume that the maximum control amplitude $u_{\max}$ is a non-increasing function $u_{\max} : \mathbb{N} \to]0,1[$; recall that $u_{\max}$ is the maximum robot speed. Finally, it is convenient to define the function $n \mapsto \delta(n) = nr(n) - 2\pi \in \mathbb{R}$ that compares the sum of the communication ranges of all the robots with the length of the unit circle.

We are now ready to state the main result of this section; proofs are postponed to Section 3.6.2.

Theorem 3.33 (Time complexity of agree-and-pursue law). *Given $k_{\text{prop}} \in]0,\frac{1}{2}[$, in the limit as $n \to +\infty$ and $\varepsilon \to 0^+$, the network $\mathcal{S}_{\text{circle}}$ with $u_{\max}(n) \geq k_{\text{prop}} r(n)$, the law $\mathcal{CC}_{\text{AGREE \& PURSUE}}$, and the tasks $\mathcal{T}_{\texttt{dir}}$ and $\mathcal{T}_{\varepsilon\text{-eqdstnc}}$ together satisfy the following properties:*

(i) $\mathrm{TC}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Theta(r(n)^{-1})$.

(ii) If $\delta(n)$ is lower bounded by a positive constant as $n \to +\infty$, then

$$\mathrm{TC}(\mathcal{T}_{\varepsilon\text{-eqdstnc}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Omega(n^2 \log(n\varepsilon)^{-1}),$$
$$\mathrm{TC}(\mathcal{T}_{\varepsilon\text{-eqdstnc}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in O(n^2 \log(n\varepsilon^{-1})).$$

If $\delta(n)$ is upper bounded by a negative constant, then in general the law $\mathcal{CC}_{\text{AGREE \& PURSUE}}$ does not achieve $\mathcal{T}_{\varepsilon\text{-eqdstnc}}$.

Next, we study the total communication complexity of the agree-and-pursue control and communication law. First, we note that any message in $\mathbb{A} = \mathbb{S}^1 \times \{\texttt{cc}, \texttt{c}\} \times \{1,\dots,n\} \cup \{\texttt{null}\}$ requires only a finite number of basic messages to encode, that is, $|\mathbb{A}|_{\text{basic}} \in O(1)$.

Theorem 3.34 (Total communication complexity of agree-and-pur-

sue law). *For $k_{\text{prop}} \in \,]0, \frac{1}{2}[$, in the limit as $n \to +\infty$ and $\varepsilon \to 0^+$, the network $\mathcal{S}_{\text{circle}}$ with $u_{\max}(n) \geq k_{\text{prop}} r(n)$, the law $\mathcal{CC}_{\text{AGREE \& PURSUE}}$, and the tasks $\mathcal{T}_{\texttt{dir}}$ and $\mathcal{T}_{\varepsilon\text{-eqdstnc}}$ together satisfy the following properties:*

(i) If $\delta(n) \geq \pi(1/k_{\text{prop}} - 2)$ as $n \to +\infty$, then

$$\text{TCC}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Theta(n^2 r(n)^{-1});$$

otherwise, if $\delta(n) \leq \pi(1/k_{\text{prop}} - 2)$ as $n \to +\infty$, then

$$\text{TCC}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Omega(n^3 + nr(n)^{-1}),$$
$$\text{TCC}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in O(n^2 r(n)^{-1}).$$

(ii) If $\delta(n)$ is lower bounded by a positive constant as $n \to +\infty$, then

$$\text{TCC}(\mathcal{T}_{\varepsilon\text{-eqdstnc}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Omega(n^3 \delta(n) \log(n\varepsilon)^{-1}),$$
$$\text{TCC}(\mathcal{T}_{\varepsilon\text{-eqdstnc}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in O(n^4 \log(n\varepsilon^{-1})).$$

Remark 3.35 (Comparison with leader election). Let us compare the agree-and-pursue control and communication law with the classical LE LANN-CHANG-ROBERTS (LCR) ALGORITHM for leader election discussed in Section 1.5.4. The leader election task consists of electing a unique agent among all agents in the network; therefore, it is different from, but closely related to, the coordination task $\mathcal{T}_{\texttt{dir}}$. The LCR ALGORITHM operates on a static network with the ring communication topology, and achieves leader election with time and total communication complexity $\Theta(n)$ and $\Theta(n^2)$, respectively. The agree-and-pursue law operates on a robotic network with the $r(n)$-disk communication topology, and achieves $\mathcal{T}_{\texttt{dir}}$ with time and total communication complexity, respectively, $\Theta(r(n)^{-1})$ and $O(n^2 r(n)^{-1})$. If wireless communication congestion is modeled by $r(n)$ of order $1/n$ as in Remark 3.8, then the two algorithms have identical time complexity and the LCR ALGORITHM has better communication complexity. Note that computations on a possibly disconnected, dynamic network are more complex than on a static ring topology. •

3.5 NOTES

The study of multi-robot systems has a long and rich history. Some recent examples include the surveys (Asama, 1992; Cao et al., 1997; Dias et al., 2006), the text by Arkin (1998) on behavior-based robotics, and the special issues (Arai et al., 2002; Abdallah and Tanner, 2007; Bullo et al., 2009). Together with this literature, the starting points for developing the material in this chapter are the standard notions of *synchronous and asynchronous networks* in distributed (Lynch, 1997; Peleg, 2000; Tel, 2001) and parallel (Bertsekas and Tsitsiklis, 1997; Parhami, 1999) computation. The established body of knowledge on synchronous networks is, however, not directly applicable to the robotic network setting because of the agents' mobility and the ensuing dynamic communication topology.

An influential contribution toward a network model of mobile interacting robots is the work by Suzuki and Yamashita (1999). This model consists of a group of identical "distributed anonymous mobile robots" characterized as follows: no explicit communication takes place between them, and at each time instant of an "activation schedule," each robot senses the relative position of all other robots and moves according to a pre-specified algorithm. An artificial intelligence approach to multi-agent behavior in a shared environment is taken in Moses and Tennenholtz (1995). Santoro (2001) provides, with an emphasis on computer science aspects, a brief survey of models, algorithms, and the need for appropriate complexity notions. Recently, a notion of communication complexity for control and communication algorithms in multi-robot systems has been analyzed by Klavins (2003); see also Klavins and Murray (2004). Notions of failures and robustness in robotic networks are discussed by Gupta et al. (2006). From a broad hybrid networked systems viewpoint, our robotic network model can be regarded as special cases of the general modeling paradigms discussed in Lygeros et al. (2003), Lynch et al. (2003), and Sanfelice et al. (2007).

A key feature of the synchronous robotic network model proposed in this chapter is the adoption of proximity graphs from computational geometry as a basis for our communication model. This design choice is justified by the vast wireless networking literature, where this assumption is made. The simplest communication model, in which two robots communicate only if they are within a fixed communication range, is a common model adopted, for example, in the studies by Gupta and Kumar (2000), Li (2003), Lloyd et al. (2005), and Santi (2005). These works study the proximity graph solutions to various communication optimization problems; this discipline is referred to as *topology control* (cf., Remark 3.8). Although we focus our presentation on the topological aspect of the communication service, more realistic communication models would include randomness, packet losses, coding, quantization, and delays (see, e.g., Toh, 2001; Tse and Viswanath, 2005).

Next, we review some literature on emergent and self-organized swarming behaviors in biological groups. Interesting dynamical systems arise in biological networks at multiple levels of resolution, all the way from interactions among molecules and cells (Miller and Bassler, 2001) to the behavioral ecology of animal groups (Okubo, 1986). Flocks of birds and schools of fish can travel in formation and act as one (see Parrish et al., 2002), allowing these animals to defend themselves against predators and protect their territories. Wildebeest and other animals exhibit complex collective behaviors when migrating, such as obstacle avoidance, leader election, and formation-keeping (see Sinclair, 1977; Gueron and Levin, 1993). Certain foraging behaviors include individual animals partitioning their environment into non-overlapping zones (see Barlow, 1974). Honey bees (Seeley and Buhrman, 1999), gorillas (Stewart and Harcourt, 1994), and whitefaced capuchins (Boinski and Campbell, 1995) exhibit synchronized group activities such as initiation of motion and change of travel direction. These remarkable dynamic capabilities are achieved apparently without following a group leader; see Barlow (1974), Okubo (1986), Gueron and Levin (1993), Stewart and Harcourt (1994), Seeley and Buhrman (1999), Boinski and Campbell (1995), and Parrish et al. (2002) for specific examples of animal species, and Conradt and Roper (2003), and Couzin et al. (2005) for general studies. A comprehensive exposition of bio-inspired optimization and control methods is presented in Passino (2004).

Regarding distributed motion coordination algorithms, much progress has been made on collective pattern formation (Suzuki and Yamashita, 1999; Belta and Kumar, 2004; Justh and Krishnaprasad, 2004; Sepulchre et al., 2007; Paley et al., 2007; Yang et al., 2008), flocking (Olfati-Saber, 2006; Lee and Spong, 2007; Tanner et al., 2007; Moshtagh and Jadbabaie, 2007), motion feasibility of formations (Tabuada et al., 2005), formation control using rigidity and persistence theory (Olfati-Saber and Murray, 2002; Baillieul and Suri, 2003; Hendrickx et al., 2007; Krick, 2007; Yu et al., 2009), formation stability (Tanner et al., 2004; Lafferriere et al., 2005; Kang et al., 2006; Dunbar and Murray, 2006; Smith and Hadaegh, 2007; Zheng et al., 2008), motion camouflage (Justh and Krishnaprasad, 2006), self-assembly (Klavins et al., 2006), swarm aggregation (Gazi and Passino, 2003), gradient climbing (Ögren et al., 2004; Cortés, 2007), cyclic-pursuit (Bruckstein et al., 1991; Marshall et al., 2004; Martínez and Bullo, 2006; Smith et al., 2005; Pavone and Frazzoli, 2007), vehicle routing (Sharma et al., 2007), motion planning with collision avoidance (Lumelsky and Harinarayan, 1997; Hu et al., 2007; Pallottino et al., 2007), and cooperative boundary estimation (Bertozzi et al., 2004; Zhang and Leonard, 2005; Clark and Fierro, 2007; Casbeer et al., 2006; Susca et al., 2008). It is also worth mentioning works on network localization, estimation, and tracking (see, e.g., Aspnes et al., 2006; Barooah and Hespanha, 2007; Oh et al., 2007; and the references therein).

Much research has been devoted to distributed task allocation problems. The work in (Gerkey and Mataric, 2004) proposes a taxonomy of task allocation problems. In papers such as (Godwin et al., 2006; Alighanbari and How, 2006; Schumacher et al., 2003; Moore and Passino, 2007; Tang and Özgüner, 2005), advanced heuristic methods are developed, and their effectiveness is demonstrated through analysis, simulation or real world implementation. Distributed auction algorithms are discussed in (Castañón and Wu, 2003;

Moore and Passino, 2007) building on the classic works in (Bertsekas and Castañón, 1991, 1993). A distributed mixed-integer-linear-programming solver is proposed in (Alighanbari and How, 2006). A spatially distributed receding-horizon scheme is proposed in (Frazzoli and Bullo, 2004; Pavone et al., 2007). There has also been prior work on target assignment problems (Beard et al., 2002; Arslan et al., 2007; Zavlanos and Pappas, 2007; Smith and Bullo, 2009). Target allocation for vehicles with nonholonomic constraints is studied in (Rathinam et al., 2007; Savla et al., 2008, 2009).

3.6 PROOFS

This section gathers the proofs of the main results presented in the chapter.

3.6.1 Proof of Proposition 3.31

Proof. Let $t \mapsto x(t)$ and $t \mapsto \tilde{x}(t)$ denote the network evolutions starting from $x_0 \in \prod_{i\in I} X_0^{[i]}$ under $\mathcal{CC}$ and $\mathcal{CC}_{(s,\mathcal{P}_I)}$, respectively. From the definition of rescheduling, one can verify that, for all $k \in \mathbb{Z}_{\geq 0}$,

$$\tilde{x}^{[i]}(t) = \begin{cases} \tilde{x}^{[i]}(F(k-1)+1), & \text{for } t \in \bigcup_{\ell=F(k-1)+1}^{F(k)-1} [\ell, \ell+1], \\ x^{[i]}(t - F(k) + k), & \text{for } t \in [F(k), F(k)+1]. \end{cases} \tag{3.6.1}$$

By the definition of time complexity $\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0)$, we have $\mathcal{T}(x(k)) = \texttt{true}$, for all $k \geq \mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0)$, and $\mathcal{T}(x(\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0)-1)) = \texttt{false}$. We rewrite these equalities in terms of the trajectories of $\mathcal{CC}_{(s,\mathcal{P}_I)}$. From (3.6.1), we write $x^{[i]}(k) = \tilde{x}^{[i]}(F(k))$, for all $i \in I$ and $k \in \mathbb{Z}_{\geq 0}$. Therefore, we have

$$\begin{aligned} &\mathcal{T}(\tilde{x}(F(k))) = \mathcal{T}(x(k)) = \texttt{true}, \qquad \text{for all } F(k) \geq F(\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0)), \\ &\mathcal{T}(\tilde{x}(F(\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0)-1))) = \mathcal{T}(x(\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0)-1)) = \texttt{false}, \end{aligned}$$

where we have used the rescheduled message-generation function in (3.3.2). Now, note that by equation (3.6.1), $\tilde{x}^{[i]}(\ell) = \tilde{x}^{[i]}(F(\lfloor \ell/s \rfloor - 1)+1)$, for all $\ell \in \mathbb{Z}_{\geq 0}$ and all $i \in I$. Therefore, $\mathcal{T}(\tilde{x}(F(\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0)-1)+1)) = \mathcal{T}(\tilde{x}(F(\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0))))$ and we can rewrite the previous identities as

$$\begin{aligned} \mathcal{T}(\tilde{x}(k)) &= \texttt{true}, \quad \text{for all } k \geq F(\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0)-1)+1, \\ \mathcal{T}(\tilde{x}(F(\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0)-1))) &= \texttt{false}, \end{aligned}$$

which implies $\mathrm{TC}(\mathcal{T}, \mathcal{CC}_{(s,\mathcal{P}_I)}, x_0) = F(\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0)-1)+1 = s\,\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0)$. As for the mean communication complexity, additivity of $\mathrm{C}_{\mathrm{rnd}}$ implies

$$\begin{aligned} &\mathrm{C}_{\mathrm{rnd}} \circ \mathcal{M}(\ell, x(\ell)) \\ &= \mathrm{C}_{\mathrm{rnd}} \circ \mathcal{M}(F(\ell)-s+1, \tilde{x}(F(\ell)-s+1)) + \cdots + \mathrm{C}_{\mathrm{rnd}} \circ \mathcal{M}(F(\ell), \tilde{x}(F(\ell))), \end{aligned}$$

where we have used $F(\ell-1)+1 = F(\ell)-s+1$. We conclude the proof by computing

$$\begin{aligned}\sum_{\ell=0}^{\mathrm{TC}(\mathcal{T},\mathcal{CC}_{(s,\mathcal{P}_I)},x_0)-1} \mathrm{C}_{\mathrm{rnd}} \circ \mathcal{M}(\ell,\tilde{x}(\ell)) &= \sum_{\ell=0}^{F(\mathrm{TC}(\mathcal{T},\mathcal{CC},x_0)-1)} \mathrm{C}_{\mathrm{rnd}} \circ \mathcal{M}(\ell,\tilde{x}(\ell)) \\ &= \sum_{\ell=0}^{\mathrm{TC}(\mathcal{T},\mathcal{CC},x_0)-1} \sum_{k=F(\ell)-s+1}^{F(\ell)} \mathrm{C}_{\mathrm{rnd}} \circ \mathcal{M}(k,\tilde{x}(k)) \\ &= \sum_{\ell=0}^{\mathrm{TC}(\mathcal{T},\mathcal{CC},x_0)-1} \mathrm{C}_{\mathrm{rnd}} \circ \mathcal{M}(\ell,x(\ell)).\end{aligned}$$

■

3.6.2 Proof of Theorem 3.33

Proof. In the following four *STEPS*, we prove the two upper bounds and the two lower bounds.

STEP 1: We start by proving the upper bound in statement (i). We claim that $\mathrm{TC}(\mathcal{T}_{\mathtt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \leq 2\pi/(k_{\mathrm{prop}} r(n))$, and we reason by contradiction, that is, we assume that there exists an initial condition which gives rise to an execution with time complexity strictly larger than $2\pi/(k_{\mathrm{prop}} r(n))$. Without loss of generality, assume $\mathtt{dir}^{[n]}(0) = \mathtt{c}$. For $\ell \leq 2\pi/(k_{\mathrm{prop}} r(n))$, define

$$k(\ell) = \operatorname{argmin}\{\mathrm{dist}_{\mathtt{cc}}(\theta^{[n]}(0), \theta^{[i]}(\ell)) \mid \mathtt{dir}^{[i]}(\ell) = \mathtt{cc},\ i \in \{1,\dots,n\}\}.$$

In other words, agent $k(\ell)$ is the agent moving counterclockwise that has smallest counterclockwise distance from the initial position of agent n. Note that $k(\ell)$ is well-defined since, by hypothesis of contradiction, $\mathcal{T}_{\mathtt{dir}}$ is `false` for $\ell \leq 2\pi/(k_{\mathrm{prop}} r(n))$. According to the state-transition function of the law $\mathcal{CC}_{\text{AGREE \& PURSUE}}$ (cf., Section 3.1.3), messages with $\mathtt{dir} = \mathtt{cc}$ can only travel counterclockwise, while messages with $\mathtt{dir} = \mathtt{c}$ can only travel clockwise. Therefore, the position of agent $k(\ell)$ at time ℓ can only belong to the counterclockwise interval from the position of agent $k(0)$ at time 0 to the position of agent n at time 0.

Let us examine how fast the message from agent n travels clockwise. To this end, for $\ell \leq 2\pi/(k_{\mathrm{prop}} r(n))$, define

$$j(\ell) = \operatorname{argmax}\{\mathrm{dist}_{\mathtt{c}}(\theta^{[n]}(0), \theta^{[i]}(\ell)) \mid \mathtt{max\text{-}id}^{[i]}(\ell) = n,\ i \in \{1,\dots,n\}\}.$$

In other words, agent $j(\ell)$ has `max-id` equal to n, is moving clockwise, and is the agent furthest from the initial position of agent n in the clockwise direction with these two properties. Initially, $j(0) = n$. Additionally, for $\ell \leq 2\pi/(k_{\text{prop}} r(n))$, we claim that

$$\text{dist}_{\mathtt{c}}(\theta^{[j(\ell)]}(\ell), \theta^{[j(\ell+1)]}(\ell+1)) \geq k_{\text{prop}} r(n).$$

This happens because either (1) there is no agent clockwise-ahead of $\theta^{[j(\ell)]}(\ell)$ within clockwise distance $r(n)$, and therefore, the claim is obvious, or (2) there are such agents. In case (2), let m denote the agent whose clockwise distance to agent $j(\ell)$ is maximal within the set of agents with clockwise distance $r(n)$ from $\theta^{[j(\ell)]}(\ell)$. Then,

$$\begin{aligned}
&\text{dist}_{\mathtt{c}}(\theta^{[j(\ell)]}(\ell), \theta^{[j(\ell+1)]}(\ell+1)) \\
&\quad = \text{dist}_{\mathtt{c}}(\theta^{[j(\ell)]}(\ell), \theta^{[m]}(\ell+1)) \\
&\quad = \text{dist}_{\mathtt{c}}(\theta^{[j(\ell)]}(\ell), \theta^{[m]}(\ell)) + \text{dist}_{\mathtt{c}}(\theta^{[m]}(\ell), \theta^{[m]}(\ell+1)) \\
&\quad \geq \text{dist}_{\mathtt{c}}(\theta^{[j(\ell)]}(\ell), \theta^{[m]}(\ell)) + k_{\text{prop}}\big(r(n) - \text{dist}_{\mathtt{c}}(\theta^{[j(\ell)]}(\ell), \theta^{[m]}(\ell))\big) \\
&\quad = k_{\text{prop}} r(n) + (1 - k_{\text{prop}})\, \text{dist}_{\mathtt{c}}(\theta^{[j(\ell)]}(\ell), \theta^{[m]}(\ell)) \;\geq k_{\text{prop}} r(n),
\end{aligned}$$

where the first inequality follows from the fact that at time ℓ there can be no agent whose clockwise distance to agent m is less than $(r(n) - \text{dist}_{\mathtt{c}}(\theta^{[j(\ell)]}(\ell), \theta^{[m]}(\ell)))$. Therefore, after a number of communication rounds larger than $2\pi/(k_{\text{prop}} r(n))$, the message with $\texttt{max-id} = n$ has traveled the whole circle in the clockwise direction, and must therefore have reached agent $k(\ell)$. This is a contradiction.

STEP 2: We prove the lower bound in statement (i). If $r(n) > \pi$ for all n, then $1/r(n) < 1/\pi$, and the upper bound is $\text{TC}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in O(1)$. Obviously, the time complexity of any evolution with an initial configuration where $\texttt{dir}^{[i]}(0) = \texttt{cc}$ for $i \in \{1, \dots, n-1\}$, $\texttt{dir}^{[n]}(0) = \texttt{c}$, and $\mathcal{E}_{\mathcal{G}_{\text{disk}}(r)}(\theta^{[1]}(0), \dots, \theta^{[n]}(0))$ is the complete graph, is lower bounded by 1. Therefore, $\text{TC}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Omega(1)$. If $r(n) > \pi$ for all n, then we conclude $\text{TC}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Theta(r(n)^{-1})$. Assume now that $r(n) \leq \pi$ for sufficiently large n. Consider an initial configuration where $\texttt{dir}^{[i]}(0) = \texttt{cc}$ for $i \in \{1, \dots, n-1\}$, $\texttt{dir}^{[n]}(0) = \texttt{c}$, and the agents are placed as depicted in Figure 3.10. Note that, after each communication round, agent 1 has moved $k_{\text{prop}} r(n)$ in the counterclockwise direction, while agent n has moved $k_{\text{prop}} r(n)$ in the clockwise direction. These two agents keep moving at full speed toward each other until they become neighbors at a time lower bounded by

$$\frac{2\pi - r(n)}{2k_{\text{prop}} r(n)} > \frac{\pi}{k_{\text{prop}} r(n)} - 1.$$

We conclude that $\text{TC}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Omega(r(n)^{-1})$.

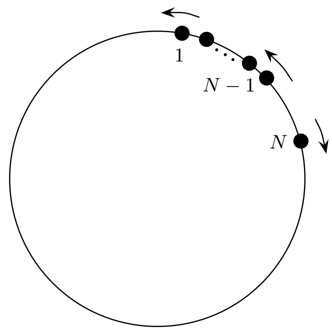


Figure 3.10 Initial condition for the lower bound of $\mathrm{TC}(\mathcal{T}_{\mathtt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}})$, with $0 < \mathrm{dist}_{\mathtt{c}}(\theta^{[n-1]}(0), \theta^{[n]}(0)) - r(n) < \varepsilon$ and $\mathrm{dist}_{\mathtt{c}}(\theta^{[1]}(0), \theta^{[n-1]}(0)) \leq r(n) - \varepsilon$, for some fixed $\varepsilon > 0$.

STEP 3: We now prove the upper bound in (ii). We begin by noting that the lower bound on δ implies $r(n)^{-1} \in O(n)$. Therefore, we know that $\mathrm{TC}(\mathcal{T}_{\mathtt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}})$ belongs to $O(n)$ and is negligible as compared with the claimed upper bound estimates for $\mathrm{TC}(\mathcal{T}_{\varepsilon\text{-eqdstnc}}, \mathcal{CC}_{\text{AGREE \& PURSUE}})$. In what follows, we therefore assume that $\mathcal{T}_{\mathtt{dir}}$ has been achieved and that, without loss of generality, all agents are moving clockwise. We now prove a fact regarding connectivity. At time $\ell \in \mathbb{Z}_{\geq 0}$, let $H(\ell)$ be the union of all the empty "circular segments" of length at least $r(n)$, that is, let

$$H(\ell) = \{x \in \mathbb{S}^1 \mid \min_{i \in \{1,\dots,n\}} \mathrm{dist}_{\mathtt{c}}(x, \theta^{[i]}(\ell)) + \min_{j \in \{1,\dots,n\}} \mathrm{dist}_{\mathtt{cc}}(x, \theta^{[j]}(\ell)) > r(n)\}.$$

In other words, $H(\ell)$ does not contain any point between two agents separated by a distance less than $r(n)$, and each connected component of $H(\ell)$ has length at least $r(n)$. Let $n_H(\ell)$ be the number of connected components of $H(\ell)$; if $H(\ell)$ is empty, then we take the convention that $n_H(\ell) = 0$. Clearly, $n_H(\ell) \leq n$. We claim that if $n_H(\ell) > 0$, then $\tau \mapsto n_H(\ell + \tau)$ is non-increasing. Let $d(\ell) < r(n)$ be the distance between any two consecutive agents at time ℓ. Because both agents move in the same direction, a simple calculation shows that

$$\begin{aligned} d(\ell+1) &\leq d(\ell) + k_{\mathrm{prop}}(r - d(\ell)) = (1 - k_{\mathrm{prop}}) d(\ell) + k_{\mathrm{prop}} r(n) \\ &< (1 - k_{\mathrm{prop}}) r + k_{\mathrm{prop}} r(n) = r(n). \end{aligned}$$

This means that the two agents remain within distance $r(n)$ and, therefore, connected at the following time instant. Because the number of connected components of $\mathcal{E}_{\mathcal{G}_{\mathrm{disk}}(r)}(\theta^{[1]}, \dots, \theta^{[n]})$ does not increase, it follows that the number of connected components of H cannot increase. Next, we claim that if $n_H(\ell) > 0$, then there exists $\tau > \ell$ such that $n_H(\tau) < n_H(\ell)$. By contradiction, assume that $n_H(\ell) = n_H(\tau)$ for all $\tau \geq \ell$. Without loss of generality, let $\{1, \dots, m\}$ be a set of agents with the properties that $\mathrm{dist}_{\mathtt{cc}}\left(\theta^{[i]}(\ell), \theta^{[i+1]}(\ell)\right) \leq r(n)$, for $i \in \{1, \dots, m\}$, that $\theta^{[1]}(\ell)$ and $\theta^{[m]}(\ell)$

belong to the boundary of $H(\ell)$, and that there is no other set with the same properties and more agents. (Note that this implies that the agents $1, \dots, m$ are in counterclockwise order.) One can show that, for $\tau \geq \ell$,

$$\theta^{[1]}(\tau+1) = \theta^{[1]}(\tau) - k_{\text{prop}} r(n),$$
$$\theta^{[i]}(\tau+1) = \theta^{[i]}(\tau) - k_{\text{prop}} \operatorname{dist}_{\mathsf{c}}(\theta^{[i]}(\tau), \theta^{[i-1]}(\tau)),$$

for $i \in \{2, \dots, m\}$. If we consider the inter-agent distances

$$d(\tau) = \big(\operatorname{dist}_{\mathsf{cc}}(\theta^{[1]}(\tau), \theta^{[2]}(\tau)), \dots, \operatorname{dist}_{\mathsf{cc}}(\theta^{[m-1]}(\tau), \theta^{[m]}(\tau)) \big) \in \mathbb{R}^{m-1}_{>0},$$

then the previous equations can be rewritten as

$$d(\tau+1) = \operatorname{Trid}_{m-1}(k_{\text{prop}}, 1-k_{\text{prop}}, 0)\, d(\tau) + r(n) k_{\text{prop}} \boldsymbol{e}_1,$$

where the linear map $(a, b, c) \mapsto \operatorname{Trid}_{m-1}(a, b, c) \in \mathbb{R}^{(m-1)\times(m-1)}$ is defined in Section 1.6.4. This is a discrete-time affine time-invariant dynamical system with unique equilibrium point $r(n)\mathbf{1}_{m-1}$. By construction, the initial condition of this system satisfies $\|d(0) - r(n)\mathbf{1}_{m-1}\|_2 \leq r(n)\sqrt{m-1}$. By Theorem 1.79(ii) in Section 1.6.4, for $\eta_1 \in \,]0,1[$, the solution $\tau \mapsto d(\tau)$ to this system reaches a ball of radius η_1 centered at the equilibrium point in time $O(m \log m + \log \eta_1^{-1})$. (Here we have used the fact that the initial condition of this system is bounded.) In turn, this implies that $\tau \mapsto \sum_{i=1}^{m} d_i(\tau)$ is larger than $(m-1)(r(n) - \eta_1)$ in time $O(m \log m + \log \eta_1^{-1})$. We are now ready to find the contradiction, and show that $n_H(\tau)$ cannot remain equal to $n_H(\ell)$ for all time τ. After time $O(m \log m + \log \eta_1^{-1}) = O(n \log n + \log \eta_1^{-1})$, we have

$$\begin{aligned} 2\pi &\geq n_H(\ell) r(n) + \sum_{j=1}^{n_H(\ell)} (r(n) - \eta_1)(m_j - 1) \\ &= n_H(\ell) r(n) + (n - n_H(\ell))(r(n) - \eta_1) = n_H(\ell)\eta_1 + n(r(n) - \eta_1). \end{aligned}$$

Here, $m_1, \dots, m_{n_H(\ell)}$ are the numbers of agents in each isolated group, and each connected component of $H(\ell)$ has length at least $r(n)$. Now, take $\eta_1 = (nr(n) - 2\pi)n^{-1} = \delta(n)n^{-1}$, and the contradiction follows from

$$\begin{aligned} 2\pi &\geq n_H(\ell)\eta_1 + nr(n) - n\eta_1 \\ &= n_H(\ell)\eta_1 + nr(n) + 2\pi - nr(n) = n_H(\ell)\eta_1 + 2\pi. \end{aligned}$$

In summary, this shows that the number of connected components of $H(\ell)$ decreases by one in time $O(n \log n + \log \eta_1^{-1}) = O(n \log n + \log(n\delta(n)^{-1}))$. Note that δ being lower bounded implies that $n\delta(n)^{-1} = O(n)$ and, therefore, $O(n \log n + \log(n\delta(n)^{-1})) = O(n \log n)$. Iterating this argument n times, in time $O(n^2 \log n)$ the set H will become empty. At that time, the resulting network will obey the discrete-time linear time-invariant dynamical system

$$d(\tau+1) = \operatorname{Circ}_n(k_{\text{prop}}, 1-k_{\text{prop}}, 0)\, d(\tau), \tag{3.6.2}$$

where the linear map $(a, b, c) \mapsto \mathrm{Circ}_n(a, b, c) \in \mathbb{R}^{n\times n}$ is defined in Section 1.6.4 and where $d : \mathbb{Z}_{\geq 0} \to \mathbb{R}^n_{>0}$ is defined by

$$d(\tau) = \big(\operatorname{dist}_{\mathtt{cc}}(\theta^{[1]}(\tau), \theta^{[2]}(\tau)), \dots, \operatorname{dist}_{\mathtt{cc}}(\theta^{[n]}(\tau), \theta^{[n+1]}(\tau))\big),$$

with the convention $\theta^{[n+1]} = \theta^{[1]}$. By Theorem 1.79(iii) in Section 1.6.4, in time $O\big(n^2 \log \varepsilon^{-1}\big)$, the error 2-norm satisfies the contraction inequality $\big\|d(\tau) - d_*\big\|_2 \leq \varepsilon \|d(0) - d_*\|_2$, for $d_* = \frac{2\pi}{n}\mathbf{1}_n$. We convert this inequality on 2-norms into an appropriate inequality on ∞-norms as follows. Note that $\|d(0) - d_*\|_\infty = \max_{i\in\{1,\dots,n\}} |d^{[i]}(0) - d_*^{[i]}| \leq 2\pi$. For $\eta_2 \in \,]0, 1[$ and for τ of order $n^2 \log \eta_2^{-1}$,

$$\begin{aligned}\|d(\tau) - d_*\|_\infty \leq \|d(\tau) - d_*\|_2 &\leq \eta_2 \|d(0) - d_*\|_2 \\ &\leq \eta_2 \sqrt{n} \|d(0) - d_*\|_\infty \leq \eta_2 2\pi \sqrt{n}.\end{aligned}$$

This means that the desired configuration is achieved for $\eta_2 2\pi\sqrt{n} = \varepsilon$, that is, in time $O(n^2 \log \eta_2^{-1}) = O(n^2 \log(n\varepsilon^{-1}))$. In summary, the equidistance task is achieved in time $O(n^2 \log(n\varepsilon^{-1}))$.

STEP 4: Finally, we prove the lower bound in (ii). As we reasoned before, $\mathrm{TC}(\mathcal{T}_{\mathtt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}})$ is negligible as compared with the claimed lower bound estimate for $\mathrm{TC}(\mathcal{T}_{\varepsilon\text{-eqdstnc}}, \mathcal{CC}_{\text{AGREE \& PURSUE}})$ and, therefore, we assume that $\mathcal{T}_{\mathtt{dir}}$ has been achieved. We consider an initial configuration with the properties that: (i) agents are counterclockwise-ordered according to their unique identifier; (ii) the set $H(0)$ is empty; and (iii) the inter-agent distances $d(0) = \big(\operatorname{dist}_{\mathtt{cc}}(\theta^{[1]}(0), \theta^{[2]}(0)), \dots, \operatorname{dist}_{\mathtt{cc}}(\theta^{[n]}(0), \theta^{[1]}(0))\big)$ are

$$d(0) = \frac{2\pi}{n}\mathbf{1}_n + \frac{\pi - \varepsilon'}{n}(\mathbf{v}_n + \overline{\mathbf{v}}_n),$$

where $\varepsilon' \in \,]\pi, 0[$ and where $\mathbf{v}_n$ is the eigenvector of $\mathrm{Circ}_n(k_{\text{prop}}, 1 - k_{\text{prop}}, 0)$ corresponding to the eigenvalue $1 - k_{\text{prop}} + k_{\text{prop}} \cos\big(\frac{2\pi}{n}\big) - k_{\text{prop}}\sqrt{-1} \sin\big(\frac{2\pi}{n}\big)$ (see Section 1.6.4). Straightforward calculations show the equality $\mathbf{v}_n + \overline{\mathbf{v}}_n = 2(1, \cos(2\pi/n), \dots, \cos((n-1)2\pi/n))$ and that $\|\mathbf{v}_n + \overline{\mathbf{v}}_n\|_2 = \sqrt{2n}$. In turn, this implies that $d(0) \in \mathbb{R}^n_{>0}$ and that $\|d(0) - \frac{2\pi}{n}\mathbf{1}_n\|_2 \in O(1/\sqrt{n})$. Take $\eta_3 \in \,]0, 1[$. The argument described in the proof of Theorem 1.79(iii) leads to the following statement: the 2-norm of the difference between $\ell \mapsto d(\ell)$ and the desired configuration $\frac{2\pi}{n}\mathbf{1}_n$ decreases by a factor η_3 in time of order $n^2 \log \eta_3^{-1}$. Given an initial error of order $O(1/\sqrt{n})$ and a final desired error of order ε, we set $\eta_3 = \varepsilon\sqrt{n}$ and obtain the desired result that it takes time of order $n^2 \log(n\varepsilon)^{-1}$ to reduce the 2-norm error and, therefore, the ∞-norm error to size ε. This concludes the proof. ■

3.6.3 Proof of Theorem 3.34

Proof. Note that the number of edges in $\mathcal{S}_{\text{circle}}$ is at most $O(n^2)$, as it is possible that all robots are within distance $r(n)$ of each other. The upper bounds in (i) and (ii) then follow from inequality (3.3.1) and Theorem 3.33. To prove the lower bounds, we follow the steps and notation in the proof of Theorem 3.33. Regarding the lower bounds in (i), we examine the evolution of the initial configuration depicted in Figure 3.10. From *STEP 2:* in the proof of Theorem 3.33, recall that the time it takes agent 1 to receive the message with `max-id` $= n$ is lower bounded by $\pi/(k_{\text{prop}}r(n)) - 1$. Our proof strategy is to lower bound the number of edges in the graph until this event happens. Note that, at initial time, there are $(n-1)^2$ edges in the communication graph of the network and, therefore, $(n-1)^2$ messages get transmitted. At the next communication round, agent 1 has moved $k_{\text{prop}}r(n)$ counterclockwise and, therefore, the number of edges is lower bounded by $(n-2)^2$. Iterating this reasoning, we see that after $i < \pi/(k_{\text{prop}}r(n))$ communication rounds, the number of edges is lower bounded by $(n-i)^2$. Now, if $\delta(n) > \pi(1/k_{\text{prop}} - 2)$, then $n > \pi/k_{\text{prop}}r(n))$, and therefore, the total communication complexity is lower bounded by

$$\sum_{i=1}^{\frac{\pi}{k_{\text{prop}}r(n)}} (n-i)^2 \in \Omega(n^2 r(n)^{-1}).$$

On the other hand, if $\delta(n) < \pi(1/k_{\text{prop}} - 2)$, then $n < \pi/k_{\text{prop}}r(n))$, and after n time steps, we lower bound the number of edges in the communication graph by the number of edges in a chain of length n, that is, $n-1$. Therefore, the total communication complexity is lower bounded by

$$\sum_{i=1}^{n}(n-i)^2 + (n-1)\Big(\frac{\pi}{k_{\text{prop}}r(n)} - n\Big) \in \Omega(n^3 + nr(n)^{-1}).$$

The two lower bounds match when $\delta(n) = \pi(1/k_{\text{prop}} - 2)$.

Regarding the lower bound in (ii), we consider first the case when $n_H(0) = 0$. In this case, the network obeys the discrete-time linear time-invariant dynamical system (3.6.2). Consider the initial condition $d(0)$ that we adopted for *STEP 4:*. We know it takes time of order $n^2 \log(n\varepsilon)^{-1}$ for the appropriate contraction property to hold. At $d(0)$, the maximal inter-agent distance is $(4\pi - \varepsilon')/n$ and it decreases during the evolution. Because each robot can communicate with any other robot within a distance $r(n)$, the number of agents within communication range of a given agent is of order $r(n)n/(4\pi - \varepsilon')$, that is, of order $\delta(n)$. From here, we deduce that the total communication complexity belongs to $\Omega(n^3\delta(n)\log(n\varepsilon)^{-1})$. ∎

3.7 EXERCISES

E3.1 **(Orientation dynamics).** We review some basic kinematic concepts about orientation dynamics, (see, e.g., Bullo and Lewis, 2004; Spong et al., 2006. Define the set of skew-symmetric matrices in $\mathbb{R}^{d\times d}$ as

$$\mathfrak{so}(d) = \{S \in \mathbb{R}^{d\times d} \mid S = -S^T\}.$$

Let $\times$ denote the cross-product on $\mathbb{R}^3$ and define the linear map $\widehat{\cdot} : \mathbb{R}^3 \to \mathfrak{so}(3)$ by $\widehat{x}y = x \times y$ for all $y \in \mathbb{R}^3$.

(i) Show that, if $x = (x_1, x_2, x_3)$, then:

$$\widehat{x} = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}.$$

(ii) Given a differentiable curve $R : [0, T] \to \mathrm{SO}(3)$, show that there exists a curve $\omega : [0, T] \to \mathbb{R}^3$ such that

$$\dot{R}(t) = R(t)\widehat{\omega}(t).$$

These two results lead to a motion model of a relative sensing network with time-varying orientation. Generalizing the constant-orientation model in equation (3.2.2), the complete position and orientation dynamics may be written as

$$\dot{p}^{[i]}_{\text{fixed}}(t) = R^{[i]}_{\text{fixed}}(t)\, u^{[i]}_i,$$
$$\dot{R}^{[i]}_{\text{fixed}}(t) = R^{[i]}_{\text{fixed}}(t)\, \widehat{\omega}^{[i]}_i,$$

where, for $i \in \{1, \dots, n\}$, $u^{[i]}_i$ and $\omega^{[i]}_i$ are the linear and the body angular velocities of robot i, respectively.

E3.2 **(Variation of the agree & pursue control and communication law).** Consider the AGREE & PURSUE control and communication law defined in Section 3.1.3, with the state transition function replaced by the following:

```
function stf(θ, w, y)
for each non-null message (θ_rcvd, (dir_rcvd, max-id_rcvd)) in y do
  if (max-id_rcvd > max-id) then
    new-dir := dir_rcvd
    new-id := max-id_rcvd
return (new-dir, new-id)
```

The only difference between this law and the AGREE & PURSUE law in Section 3.1.3 is that, in each communication round, each agent picks the message with the largest value of `max-id` among all messages received (instead of among the messages received only from agents moving towards its position). We refer to this law as MOD-AGREE & PURSUE.

Consider the direction agreement task $\mathcal{T}_{\texttt{dir}} : (\mathbb{S}^1)^n \times W^n \to \{\texttt{true}, \texttt{false}\}$ defined in Example 3.22. Assume that $\texttt{dir}^{[n]}(0) = \texttt{c}$, and let $k \in \{1, \dots, n-1\}$ be the largest identity such that $\texttt{dir}^{[k]}(0) = \texttt{cc}$. Do the following tasks:

(i) Show that if the message from agent k gets delivered to agents clockwise-placed with respect to agent k along two consecutive communication rounds, then the message from agent k has traveled at least $(1-k_{\text{prop}})r(n)$ along the circle in the clockwise direction.

(ii) Show that, if $\mathrm{dist}_{\mathtt{cc}}(\theta^{[n]}(0), \theta^{[k]}(0)) < 2\, r(n)$, then

$$\mathrm{TC}(\mathcal{T}_{\mathtt{dir}}, \mathcal{CC}_{\text{MOD-AGREE \& PURSUE}}, x_0, w_0) = \Theta(r(n)^{-1}).$$

(iii) Implement the algorithm in your favorite simulation software (for example, Mathematica© Matlab© or Maple©), and compute the time complexity of multiple executions of the algorithm starting from different initial conditions. Does your simulation analysis support the conjecture that

$$\mathrm{TC}(\mathcal{T}_{\mathtt{dir}}, \mathcal{CC}_{\text{MOD-AGREE \& PURSUE}}) = \Theta(r(n)^{-1})?$$

For the simulation analysis to be relevant, you should use a large number of randomly generated initial physical positions and processor states.

E3.3 **(Leader-following flocking).** Consider a group of robots moving in $\mathbb{R}^2$ according to the following discrete-time version of the planar vehicle dynamics introduced in Example 3.1:

$$\begin{aligned} x(\ell+1) &= x(\ell) + v\cos(\theta(\ell)), \\ y(\ell+1) &= y(\ell) + v\sin(\theta(\ell)), \\ \theta(\ell+1) &= \theta(\ell) + \omega. \end{aligned}$$

We let $\{(p^{[1]}, \theta^{[1]}), \dots, (p^{[n]}, \theta^{[n]})\}$ denote the robot physical states, where $p^{[i]} = (x^{[i]}, y^{[i]}) \in \mathbb{R}^2$ corresponds to the position and $\theta^{[i]} \in [0, 2\pi)$ corresponds to the orientation of the robot $i \in I$. As communication graph, we adopt the r-disk graph $\mathcal{G}_{\mathrm{disk}}(r)$ introduced in Section 2.2.

Assume that all agents move at unit speed, $v = 1$, and update their heading according to the leader-following version of Vicsek's model (see equation (1.6.5)):

$$\begin{aligned} \theta^{[1]}(\ell+1) &= \theta^{[1]}(\ell), \\ \theta^{[i]}(\ell+1) &= \mathrm{avrg}\Big(\{\theta^{[i]}(\ell)\} \cup \{\theta^{[j]}(\ell) \mid j \text{ s.t. } \|p^{[j]}(\ell) - p^{[i]}(\ell)\|_2 \leq r\}\Big), \end{aligned} \tag{E3.1}$$

for $i \in \{2, \dots, n\}$. Do the following tasks:

(i) Write the algorithm formally as a control and communication law as defined in Section 3.1.2.

(ii) Given initial conditions for the position and orientation of the robots, express (E3.1) as the time-dependent linear iteration associated to a sequence of matrices $\{F(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\}$. Are these matrices stochastic? Are they symmetric? Is the sequence non-degenerate?

(iii) We loosely define the flocking task as achieving agreement on the heading of the agents. Using Theorem 1.63, identify connectivity conditions on the sequence of graphs determined by the evolution of the network that guarantee that agents achieve flocking. What is the final orientation in which the network flocks?

Bibliography

Abdallah, C. T. and Tanner, H. G. [2007] *Complex networked control systems: introduction to the special section*, IEEE Control Systems Magazine, **27**(4), 30–32.

Alighanbari, M. and How, J. P. [2006] *Robust decentralized task assignment for cooperative UAVs*, in *AIAA Conf. on Guidance, Navigation and Control*, Keystone, CO.

Arai, T., Pagello, E., and Parker, L. E. [2002] *Guest editorial: Advances in multirobot systems*, IEEE Transactions on Robotics and Automation, **18**(5), 655–661.

Arkin, R. C. [1998] *Behavior-Based Robotics*, MIT Press, ISBN 0262011654.

Arslan, G., Marden, J. R., and Shamma, J. S. [2007] *Autonomous vehicle-target assignment: A game theoretic formulation*, ASME Journal on Dynamic Systems, Measurement, and Control, **129**(5), 584–596.

Asama, H. [1992] *Distributed autonomous robotic system configurated with multiple agents and its cooperative behaviors*, Journal of Robotics and Mechatronics, **4**(3), 199–204.

Aspnes, J., Eren, T., Goldenberg, D. K., Morse, A. S., Whiteley, W., Yang, Y. R., Anderson, B. D. O., and Belhumeur, P. [2006] *A theory of network localization*, IEEE Transactions on Mobile Computing, **5**(12), 1663–1678.

Baillieul, J. and Suri, A. [2003] *Information patterns and hedging Brockett's theorem in controlling vehicle formations*, in *IEEE Conf. on Decision and Control*, pages 556–563, Maui, HI.

Barlow, G. W. [1974] *Hexagonal territories*, Animal Behavior, **22**, 876–878.

Barooah, P. and Hespanha, J. P. [2007] *Estimation from relative measurements: Algorithms and scaling laws*, IEEE Control Systems Magazine, **27**(4), 57–74.

Beard, R. W., McLain, T. W., Goodrich, M. A., and Anderson, E. P. [2002] *Coordinated target assignment and intercept for unmanned air vehicles*, IEEE Transactions on Robotics and Automation, **18**(6), 911–922.

Belta, C. and Kumar, V. [2004] *Abstraction and control for groups of robots*, IEEE Transactions on Robotics, **20**(5), 865–875.

Bertozzi, A. L., Kemp, M., and Marthaler, D. [2004] *Determining environmental boundaries: Asynchronous communication and physical scales*, in *Cooperative Control*, V. Kumar, N. E. Leonard, and A. S. Morse, editors, volume 309 of *Lecture Notes in Control and Information Sciences*, pages 25–42, Springer, ISBN 3540228616.

Bertsekas, D. P. and Castañón, D. A. [1991] *Parallel synchronous and asynchronous implementations of the auction algorithm*, Parallel Computing, **17**, 707–732.

— [1993] *Parallel primal-dual methods for the minimum cost flow problem*, Computational Optimization and Applications, **2**(4), 317–336.

Bertsekas, D. P. and Tsitsiklis, J. N. [1997] *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, ISBN 1886529019.

Boinski, S. and Campbell, A. F. [1995] *Use of trill vocalizations to coordinate troop movement among whitefaced capuchins – a 2nd field-test*, Behaviour, **132**, 875–901.

Bollobás, B. [2001] *Random Graphs*, second edition, Cambridge University Press, ISBN 0521809207.

Bruckstein, A. M., Cohen, N., and Efrat, A. [1991] *Ants, crickets, and frogs in cyclic pursuit*, Technical Report CIS 9105, Center for Intelligent Systems, Technion, Haifa, Israel, available at `http://www.cs.technion.ac.il/tech-reports`.

Bullo, F., Cortés, J., and Piccoli, B. [2009] *Special issue on control and optimization in cooperative networks*, SIAM Journal on Control and Optimization, **48**(1), vii–vii.

Bullo, F. and Lewis, A. D. [2004] *Geometric Control of Mechanical Systems*, volume 49 of *Texts in Applied Mathematics*, Springer, ISBN 0387221956.

Cao, Y. U., Fukunaga, A. S., and Kahng, A. [1997] *Cooperative mobile robotics: Antecedents and directions*, Autonomous Robots, **4**(1), 7–27.

Casbeer, D. W., Kingston, D. B., Beard, R. W., Mclain, T. W., Li, S.-M., and Mehra, R. [2006] *Cooperative forest fire surveillance using a team of small unmanned air vehicles*, International Journal of Systems Sciences, **37**(6), 351–360.

Castañón, D. A. and Wu, C. [2003] *Distributed algorithms for dynamic reassignment*, in *IEEE Conf. on Decision and Control*, pages 13–18, Maui, HI.

Clark, J. and Fierro, R. [2007] *Mobile robotic sensors for perimeter detection and tracking*, ISA Transactions, **46**(1), 3–13.

Conradt, L. and Roper, T. J. [2003] *Group decision-making in animals*, Nature, **421**(6919), 155–158.

Cortés, J. [2007] *Distributed Kriged Kalman filter for spatial estimation*, IEEE Transactions on Automatic Control, submitted.

Couzin, I. D., Krause, J., Franks, N. R., and Levin, S. A. [2005] *Effective leadership and decision-making in animal groups on the move*, Nature, **433**(7025), 513–516.

Dias, M. B., Zlot, R., Kalra, N., and Stentz, A. [2006] *Market-based multirobot coordination: A survey and analysis*, Proceedings of the IEEE, **94**(7), 1257–1270.

Dubins, L. E. [1957] *On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents*, American Journal of Mathematics, **79**, 497–516.

Dunbar, W. B. and Murray, R. M. [2006] *Distributed receding horizon control for multi-vehicle formation stabilization*, Automatica, **42**(4), 549–558.

Emerson, A. E. [1994] *Temporal and modal logic*, in *Handbook of Theoretical Computer Science, Vol. B: Formal Models and Semantics*, J. van Leeuwen, editor, pages 997–1072, MIT Press, ISBN 0262720159.

Fainekos, G. E., Kress-Gazit, H., and Pappas, G. J. [2005] *Temporal logic motion planning for mobile robots*, in *IEEE Int. Conf. on Robotics and Automation*, pages 2032–2037, Barcelona, Spain.

Frazzoli, E. and Bullo, F. [2004] *Decentralized algorithms for vehicle routing in a stochastic time-varying environment*, in *IEEE Conf. on Decision and Control*, pages 3357–3363, Paradise Island, Bahamas.

Gazi, V. and Passino, K. M. [2003] *Stability analysis of swarms*, IEEE Transactions on Automatic Control, **48**(4), 692–697.

Gerkey, B. P. and Mataric, M. J. [2004] *A formal analysis and taxonomy of task allocation in multi-robot systems*, International Journal of Robotics Research, **23**(9), 939–954.

Godwin, M. F., Spry, S., and Hedrick, J. K. [2006] *Distributed collaboration with limited communication using mission state estimates*, in *American Control Conference*, pages 2040–2046, Minneapolis, MN.

Gueron, S. and Levin, S. A. [1993] *Self-organization of front patterns in large wildebeest herds*, Journal of Theoretical Biology, **165**, 541–552.

Gupta, P. and Kumar, P. R. [2000] *The capacity of wireless networks*, IEEE Transactions on Information Theory, **46**(2), 388–404.

Gupta, V., Langbort, C., and Murray, R. M. [2006] *On the robustness of distributed algorithms*, in *IEEE Conf. on Decision and Control*, pages 3473–3478, San Diego, CA.

Hendrickx, J. M., Anderson, B. D. O., Delvenne, J.-C., and Blondel, V. D. [2007] *Directed graphs for the analysis of rigidity and persistence in autonomous agents systems*, International Journal on Robust and Nonlinear Control, **17**(10), 960–981.

Hu, J., Prandini, M., and Tomlin, C. [2007] *Conjugate points in formation constrained optimal multi-agent coordination: A case study*, SIAM Journal on Control and Optimization, **45**(6), 2119–2137.

Justh, E. W. and Krishnaprasad, P. S. [2004] *Equilibria and steering laws for planar formations*, Systems & Control Letters, **52**(1), 25–38.

— [2006] *Steering laws for motion camouflage*, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, **462**(2076), 3629–3643.

Kang, K., Yan, J., and Bitmead, R. R. [2006] *Communication resources for disturbance rejection in coordinated vehicle control*, in *IEEE Conf. on Decision and Control and European Control Conference*, pages 5730–5735, Seville, Spain.

Klavins, E. [2003] *Communication complexity of multi-robot systems*, in *Algorithmic Foundations of Robotics V*, J.-D. Boissonnat, J. W. Burdick, K. Goldberg, and S. Hutchinson, editors, volume 7 of *Tracts in Advanced Robotics*, Springer, Berlin Heidelberg, ISBN 3540404767.

Klavins, E., Ghrist, R., and Lipsky, D. [2006] *A grammatical approach to self-organizing robotic systems*, IEEE Transactions on Automatic Control, **51**(6), 949–962.

Klavins, E. and Murray, R. M. [2004] *Distributed algorithms for cooperative control*, IEEE Pervasive Computing, **3**(1), 56–65.

Krick, L. [2007] *Application of Graph Rigidity in Formation Control of Multi-Robot Networks*, Master's thesis, University of Toronto, Canada.

Lafferriere, G., Williams, A., Caughman, J., and Veerman, J. J. P. [2005] *Decentralized control of vehicle formations*, Systems & Control Letters, **54**(9), 899–910.

Lee, D. and Spong, M. W. [2007] *Stable flocking of multiple inertial agents on balanced graphs*, IEEE Transactions on Automatic Control, **52**(8), 1469–1475.

Li, X.-Y. [2003] *Algorithmic, geometric and graphs issues in wireless networks*, Wireless Communications and Mobile Computing, **3**(2), 119–140.

Lin, Z., Francis, B., and Maggiore, M. [2005] *Necessary and sufficient graphical conditions for formation control of unicycles*, IEEE Transactions on Automatic Control, **50**(1), 121–127.

Lloyd, E. L., Liu, R., Marathe, M. V., Ramanathan, R., and Ravi, S. S. [2005] *Algorithmic aspects of topology control problems for ad hoc networks*, Mobile Networks and Applications, **10**(1-2), 19–34.

Lumelsky, V. J. and Harinarayan, K. R. [1997] *Decentralized motion planning for multiple mobile robots: The cocktail party model*, Autonomous Robots, **4**(1), 121–135.

Lygeros, J., Johansson, K. H., Simić, S. N., Zhang, J., and Sastry, S. S. [2003] *Dynamical properties of hybrid automata*, IEEE Transactions on Automatic Control, **48**(1), 2–17.

Lynch, N. A. [1997] *Distributed Algorithms*, Morgan Kaufmann, ISBN 1558603484.

Lynch, N. A., Segala, R., and Vaandrager, F. [2003] *Hybrid I/O automata*, Information and Computation, **185**(1), 105–157.

Marshall, J. A., Broucke, M. E., and Francis, B. A. [2004] *Formations of vehicles in cyclic pursuit*, IEEE Transactions on Automatic Control, **49**(11), 1963–1974.

Martínez, S. and Bullo, F. [2006] *Optimal sensor placement and motion coordination for target tracking*, Automatica, **42**(4), 661–668.

Martínez, S., Bullo, F., Cortés, J., and Frazzoli, E. [2007] *On synchronous robotic networks – Part I: Models, tasks and complexity*, IEEE Transactions on Automatic Control, **52**(12), 2199–2213.

Miller, M. B. and Bassler, B. L. [2001] *Quorum sensing in bacteria*, Annual Review of Microbiology, **55**, 165–199.

Moore, B. J. and Passino, K. M. [2007] *Distributed task assignment for mobile agents*, IEEE Transactions on Automatic Control, **52**(4), 749–753.

Moses, Y. and Tennenholtz, M. [1995] *Artificial social systems*, Computers and AI, **14**(6), 533–562.

Moshtagh, N. and Jadbabaie, A. [2007] *Distributed geodesic control laws for flocking of nonholonomic agents*, IEEE Transactions on Automatic Control, **52**(4), 681–686.

Ögren, P., Fiorelli, E., and Leonard, N. E. [2004] *Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment*, IEEE Transactions on Automatic Control, **49**(8), 1292–1302.

Oh, S., Schenato, L., Chen, P., and Sastry, S. S. [2007] *Tracking and coordination of multiple agents using sensor networks: system design, algorithms and experiments*, Proceedings of the IEEE, **95**(1), 163–187.

Okubo, A. [1986] *Dynamical aspects of animal grouping: swarms, schools, flocks and herds*, Advances in Biophysics, **22**, 1–94.

Olfati-Saber, R. [2006] *Flocking for multi-agent dynamic systems: Algorithms and theory*, IEEE Transactions on Automatic Control, **51**(3), 401–420.

Olfati-Saber, R. and Murray, R. M. [2002] *Graph rigidity and distributed formation stabilization of multi-vehicle systems*, in *IEEE Conf. on Decision and Control*, pages 2965–2971, Las Vegas, NV.

Paley, D. A., Leonard, N. E., Sepulchre, R., Grunbaum, D., and Parrish, J. K. [2007] *Oscillator models and collective motion*, IEEE Control Systems Magazine, **27**(4), 89–105.

Pallottino, L., Scordio, V. G., Frazzoli, E., and Bicchi, A. [2007] *Decentralized cooperative policy for conflict resolution in multi-vehicle systems*, IEEE Transactions on Robotics, **23**(6), 1170–1183.

Parhami, B. [1999] *Introduction to Parallel Processing: Algorithms and Architectures*, Plenum Series in Computer Science, Springer, ISBN 0306459701.

Parrish, J. K., Viscido, S. V., and Grunbaum, D. [2002] *Self-organized fish schools: an examination of emergent properties*, Biological Bulletin, **202**, 296–305.

Passino, K. M. [2004] *Biomimicry for Optimization, Control, and Automation*, Springer, ISBN 1852338040.

Pavone, M. and Frazzoli, E. [2007] *Decentralized policies for geometric pattern formation and path coverage*, ASME Journal on Dynamic Systems, Measurement, and Control, **129**(5), 633–643.

Pavone, M., Frazzoli, E., and Bullo, F. [2007] *Decentralized algorithms for stochastic and dynamic vehicle routing with general target distribution*, in *IEEE Conf. on Decision and Control*, pages 4869–4874, New Orleans, LA.

Peleg, D. [2000] *Distributed Computing. A Locality-Sensitive Approach*, Monographs on Discrete Mathematics and Applications, SIAM, ISBN 0898714648.

Penrose, M. [2003] *Random Geometric Graphs*, Oxford Studies in Probability, Oxford University Press, ISBN 0198506260.

Rathinam, S., Sengupta, R., and Darbha, S. [2007] *A resource allocation algorithm for multi-vehicle systems with non holonomic constraints*, IEEE Transactions on Automation Sciences and Engineering, **4**(1), 98–104.

Reeds, J. A. and Shepp, L. A. [1990] *Optimal paths for a car that goes both forwards and backwards*, Pacific Journal of Mathematics, **145**(2), 367–393.

Sanfelice, R. G., Goebel, R., and Teel, A. R. [2007] *Invariance principles for hybrid systems with connections to detectability and asymptotic stability*, IEEE Transactions on Automatic Control, **52**(12), 2282–2297.

Santi, P. [2005] *Topology Control in Wireless Ad Hoc and Sensor Networks*, John Wiley, ISBN 0470094532.

Santoro, N. [2001] *Distributed computations by autonomous mobile robots*, in *SOFSEM 2001: Conference on Current Trends in Theory and Practice of Informatics (Piestany, Slovak Republic)*, L. Pacholski and P. Ruzicka, editors, volume 2234 of *Lecture Notes in Computer Science*, pages 110–115, Springer, ISBN 3-540-42912-3.

Savla, K., Bullo, F., and Frazzoli, E. [2009] *Traveling Salesperson Problems for a double integrator*, IEEE Transactions on Automatic Control, (Submitted Nov. 2006) to appear.

Savla, K., Frazzoli, E., and Bullo, F. [2008] *Traveling Salesperson Problems for the Dubins vehicle*, IEEE Transactions on Automatic Control, **53**(6), 1378–1391.

Schumacher, C., Chandler, P. R., Rasmussen, S. J., and Walker, D. [2003] *Task allocation for wide area search munitions with variable path length*, in *American Control Conference*, pages 3472–3477, Denver, CO.

Seeley, T. D. and Buhrman, S. C. [1999] *Group decision-making in swarms of honey bees*, Behavioral Ecology and Sociobiology, **45**, 19–31.

Sepulchre, R., Paley, D. A., and Leonard, N. E. [2007] *Stabilization of planar collective motion: All-to-all communication*, IEEE Transactions on Automatic Control, **52**(5), 811–824.

Sharma, V., Savchenko, M., Frazzoli, E., and Voulgaris, P. [2007] *Transfer time complexity of conflict-free vehicle routing with no communications*, International Journal of Robotics Research, **26**(3), 255–272.

Sinclair, A. R. [1977] *The African Buffalo, A Study of Resource Limitation of Population*, The University of Chicago Press.

Smith, R. S. and Hadaegh, F. Y. [2007] *Closed-loop dynamics of cooperative vehicle formations with parallel estimators and communication*, IEEE Transactions on Automatic Control, **52**(8), 1404–1414.

Smith, S. L., Broucke, M. E., and Francis, B. A. [2005] *A hierarchical cyclic pursuit scheme for vehicle networks*, Automatica, **41**(6), 1045–1053.

Smith, S. L. and Bullo, F. [2009] *Monotonic target assignment for robotic networks*, IEEE Transactions on Automatic Control, **54**(10), (Submitted June 2007) to appear.

Spong, M. W., Hutchinson, S., and Vidyasagar, M. [2006] *Robot Modeling and Control*, third edition, John Wiley, ISBN 0-471-64990-2.

Stewart, K. J. and Harcourt, A. H. [1994] *Gorillas vocalizations during rest periods – signals of impending departure*, Behaviour, **130**, 29–40.

Susca, S., Martínez, S., and Bullo, F. [2008] *Monitoring environmental boundaries with a robotic sensor network*, IEEE Transactions on Control Systems Technology, **16**(2), 288–296.

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

Tabuada, P., Pappas, G. J., and Lima, P. [2005] *Motion feasibility of multi-agent formations*, IEEE Transactions on Robotics, **21**(3), 387–392.

Tang, Z. and Özgüner, Ü. [2005] *Motion planning for multi-target surveillance with mobile sensor agents*, IEEE Transactions on Robotics, **21**(5), 898–908.

Tanner, H. G., Jadbabaie, A., and Pappas, G. J. [2007] *Flocking in fixed and switching networks*, IEEE Transactions on Automatic Control, **52**(5), 863–868.

Tanner, H. G., Pappas, G. J., and Kumar, V. [2004] *Leader-to-formation stability*, IEEE Transactions on Robotics and Automation, **20**(3), 443–455.

Tel, G. [2001] *Introduction to Distributed Algorithms*, second edition, Cambridge University Press, ISBN 0521794838.

Toh, C.-K. [2001] *Ad Hoc Mobile Wireless Networks: Protocols and Systems*, Prentice Hall, ISBN 0130078174.

Tse, D. and Viswanath, P. [2005] *Fundamentals of Wireless Communication*, Cambridge University Press, ISBN 0521845270.

Xue, F. and Kumar, P. R. [2004] *The number of neighbors needed for connectivity of wireless networks*, Wireless Networks, **10**(2), 169–181.

Yang, P., Freeman, R. A., and Lynch, K. M. [2008] *Multi-agent coordination by decentralized estimation and control*, IEEE Transactions on Automatic Control, **53**(11), 2480–2496.

Yu, C., Anderson, B. D. O., Dasgupta, S., and Fidan, B. [2009] *Control of minimally persistent formations in the plane*, SIAM Journal on Control and Optimization, **48**(1), 206–233.

Zavlanos, M. M. and Pappas, G. J. [2007] *Dynamic assignment in distributed motion planning with local information*, in *American Control Conference*, pages 1173–1178, New York.

Zhang, F. and Leonard, N. E. [2005] *Generating contour plots using multiple sensor platforms*, in *IEEE Swarm Intelligence Symposium*, pages 309–316, Pasadena, CA.

Zheng, Z., Spry, S. C., and Girard, A. R. [2008] *Leaderless formation control using dynamic extension and sliding control*, in *IFAC World Congress*, pages 16027–16032, Seoul, Korea.

Algorithm Index

Subject Index

Symbol Index

IH-MCC($\mathcal{CC}, x_0, w_0$) : infinite-horizon mean communication complexity, 28
TCC($\mathcal{T}$) : total communication complexity to achieve $\mathcal{T}$, 27
TCC($\mathcal{T}, \mathcal{CC}$) : total communication complexity to achieve $\mathcal{T}$ with $\mathcal{CC}$, 27
TCC($\mathcal{T}, \mathcal{CC}, x_0, w_0$) : total communication complexity to achieve $\mathcal{T}$ with $\mathcal{CC}$ from (x_0, w_0), 27
$\mathcal{CC}_{\text{AGREE \& PURSUE}}$: agree and pursue control and communication law, 15
$\mathcal{CC}$: control and communication law, 11
$\mathcal{CC}_{(s,\mathcal{P}_I)}$: $\mathcal{P}_I$-rescheduling of $\mathcal{CC}$, 30
ctl : motion control function, 11
$\mathcal{M}(x, w)$: set of all non-null messages generated during one communication round from (x, w), 27
$\mathcal{R}$: set of mobile robots, 7
SC($\mathcal{T}, \mathcal{CC}$) : space complexity to achieve $\mathcal{T}$ with $\mathcal{CC}$, 27
$\mathcal{T}$: coordination task, 24
$\mathcal{T}_{\texttt{dir}}$: direction agreement task, 25
$\mathcal{T}_{\varepsilon\text{-eqdstnc}}$: equidistance task, 25
TC($\mathcal{T}$) : time complexity to achieve $\mathcal{T}$, 26
TC($\mathcal{T}, \mathcal{CC}$) : time complexity to achieve $\mathcal{T}$ with $\mathcal{CC}$, 26
TC($\mathcal{T}, \mathcal{CC}, x_0, w_0$) : time complexity to achieve $\mathcal{T}$ with $\mathcal{CC}$ from (x_0, w_0), 26