

UNIVERSITY of CALIFORNIA
Santa Barbara

Distributed boundary estimation and monitoring

A Dissertation submitted in partial satisfaction of the
requirements for the degree

Doctor of Philosophy

in

Electrical and Computer Engineering

by

Sara Susca

Committee in charge:

Professor Francesco Bullo, Chair

Professor Bassam Bamieh

Professor João P. Hespanha

Professor Sonia Martínez

December 2007

The dissertation of Sara Susca is approved.

Professor Bassam Bamieh

Professor João P. Hespanha

Professor Sonia Martínez

Professor Francesco Bullo, Committee Chair

December 2007

Distributed boundary estimation and monitoring

Copyright © 2007

by

Sara Susca

To my parents

Mario and Maria

Acknowledgements

I would like to thank my advisor Francesco Bullo for training me for this marathon (as he calls the PhD). His maniacal search for rigor and perfection will be with me for a long time.

I wish to thank Petar Kokotovic for believing in me when I did not. Petar introduced me to the wonderland of nonlinear dynamics and I guess since then I have been Alice.

I owe a special thank to Sonia Martínez for all the productive discussions.

I want to express all my gratitude to Prabir for being there especially in the most difficult moments. Prabir's intelligence, patience, and positivity have being indispensable for me to reach the finish line.

I cannot forget the wonderful summers in Honeywell that filled me up with immense joy. I have many people to thank for that, but I will mention only my mentor, Kailash Krishnaswamy. Thank you Kailash.

During this marathon, pottery and photography have being my energy drinks. I wish to thank my pottery teachers Genie Thomsen and Deanna Pini, my numerous ceramics classmates, and Dr. Dr. Chiranjeeb Buragohain (it is not a typo, he really has two PhDs), for teaching me much of what I know about photography. Ceramics and photography have given me the opportunity to reconvert unused neurons and find new energy to tackle research problems.

Finally, I want to thank my parents Mario and Maria, my sister Flavia and my dog Virgola for being encouraging, supportive, and above all for giving me a good reason to keep going: not to let my whole home town down!

Curriculum Vitæ

Sara Susca

Education

09/1995-04/2001 Laurea in Aerospace Engineering, Politecnico di Milano, Milano, Italy.

Experience

06/2007–09/2007 Summer Intern, Honeywell, Minneapolis, MN, USA.

06/2006–09/2006 Summer Intern, Honeywell, Minneapolis, MN, USA.

07/2001–07/2002 Visiting Scholar, Bioserve Space Technologies, Boulder, CO, USA.

Selected Publications

S. Susca, S. Martínez, and F. Bullo, “Gradient Algorithms for Polygonal Approximation of Convex Contours,” in *Automatica*, 2007, Note: To appear.

S. Susca, S. Martínez, and F. Bullo, “Monitoring Environmental Boundaries with a Robotic Sensor Network” in *IEEE Transaction on Control Systems Technology*, 2007, Note: To appear.

S. Susca, F. Bullo, and S. Martínez, “Synchronization of beads on a ring,” in *Proceedings of IEEE Conference on Decision and Control*, New Orleans, 2007.

S. Susca, F. Bullo, and S. Martínez, “Synchronization of N-beads on a ring by feedback control,” Note: In preparation.

Complete list and pdf files available at:

<http://motion.mee.ucsb.edu/papers/Author/SUSCA-S.html>.

Abstract

Distributed boundary estimation and monitoring

by

Sara Susca

This thesis illustrates some algorithms designed to enable a robotic sensor network to estimate a planar contour and to patrol it in a synchronized manner. The common thread of these algorithms are the tool needed to analyze and prove their correctness: consensus algorithms. In fact, the algorithms described in this thesis give rise to dynamical systems that can be easily analyzed once it is shown that they are just consensus algorithms in which inputs are present. This gives us the opportunity to extend the contribution of this thesis by studying some robustness properties of consensus algorithms with inputs.

Contents

Acknowledgments	v
Curriculum Vitæ	vi
Abstract	viii
List of Figures	xii
1 Introduction	1
1.1 Statement of contribution	1
1.2 Organization	4
2 Gradient algorithms for polygonal approximation of convex contours	5
2.1 Problem setup	6
2.2 Inner-polygon approximation algorithms	9
2.2.1 Discrete-time inner-polygon approximation algorithms	12
2.3 Outer-polygon approximation algorithms	16
2.3.1 Discrete-time outer-polygon approximation algorithms	21
2.4 “Outer minus inner” polygon approximation algorithms	22
2.5 Simulations	24
2.6 Summary	24

3	Monitoring environmental boundaries with a robotic sensor network	27
3.1	Approximation theory for convex bodies	29
3.2	Boundary estimation and agent pursuit algorithm	30
3.2.1	Algorithm description	32
3.2.2	Algorithm analysis	40
3.2.3	Simulations	47
3.3	Summary and open issues	52
4	ISS properties of discrete-time consensus algorithms	55
4.1	Review of ISS concepts	56
4.2	Consensus algorithms with inputs and outputs	58
4.3	Consensus algorithms with error outputs are IOS	61
4.3.1	IOS with respect to pairwise error	61
4.3.2	IOS with respect to max-min error	65
4.4	Consensus algorithms are iISnS	71
4.4.1	Consensus value for algorithms in \mathcal{C}_1	73
4.4.2	Consensus value for algorithms in \mathcal{C}_2	73
4.5	Summary	75
5	Synchronization of N-Beads on a Ring by Feedback Control	76
5.1	Introduction	76
5.2	Model and problem statement	78
5.3	Synchronization algorithm	80
5.4	Preliminary results	83
5.5	Convergence analysis	89
5.5.1	Convergence of nominal speed and desired sweeping arc	89
5.5.2	Balanced synchrony	91
5.5.3	Unbalanced synchrony	95
5.6	Simulations	106
5.6.1	Balanced collection of beads	107

5.6.2	Unbalanced collection of beads	108
5.7	Summary	114
6	Conclusions	119
	Bibliography	120
A	Metzler matrices	126

List of Figures

2.1	From left to right: the half-plane $\mathcal{H}(p_i)$ and its boundary $\ell(p_i) = \ell^+(p_i) \cup \ell^-(p_i)$, three points defining a bounded outer polygon, and three points defining an unbounded outer polygon.	8
2.2	From left to right: saddle point configuration, nearby configuration that increases the error E_I , bear by configuration that decreases the error E_I , configuration corresponding to a minimum error configuration.	11
2.3	(a) Illustration of α_i , α_{i-1} , A_i and B_i . (b) Variation of E_O described in Proposition 2.3.	16
2.4	From left to right and from top to bottom: initial condition of eleven nodes on a convex boundary, final condition after the implementation of the inner-polygon, outer-polygon, and “outer minus inner” polygon approximation algorithms.	25
2.5	From left to right: initial condition of eleven nodes on a convex boundary and final condition after the implementation of Algorithm 2.	26
3.1	In the figure the solid line is the boundary ∂Q , the triangles are the agents, the circles are the interpolation points, and the dotted line is the approximating polygon defined by the interpolation points.	31
3.2	Mobile agent moving along boundary, projecting (white arrow) and locally updating (black arrow) interpolation points.	35
3.3	Mobile agent projecting interpolation point onto the observed boundary	36
3.4	Mobile agent locally optimizing interpolation point $p_{\text{NOW}-1}$ along the observed boundary, after projecting p_{NOW}	37

3.5	This figure shows initial and final configuration after 50 seconds simulation obtained by the implementation of the ESTIMATE UPDATE AND PURSUIT ALGORITHM with $n_a = 3$, $n_{ip} = 30$, $v_0 = 1$, $k_{prop} = 0.05$, $\lambda = \frac{10}{11}$. ∂Q is time invariant. The agents position is represented by the triangles and are initialized to be on the boundary ∂Q . In the last frame also the approximating polygon is shown.	49
3.6	ESTIMATE UPDATE AND PURSUIT ALGORITHM This plots refers to the case of ∂Q being time-invariant. In the first plot from right it is shown the error $\max_{i \in \{1, \dots, n_{ip}\}} D_\lambda(p_i, p_{i+1}) - \min_{i \in \{1, \dots, n_{ip}\}} D_\lambda(p_i, p_{i+1})$ vs time. The second plot shows the arc length distances between the three agents.	50
3.7	This figure shows four different instants of the 200 seconds simulation obtained by implementing the ESTIMATE UPDATE AND PURSUIT ALGORITHM with $n_a = 4$, $n_{ip} = 35$, $v_0 = 1$, $k_{prop} = 0.05$, $\lambda = \frac{10}{11}$. The boundary ∂Q is slowly time-varying in this case. The agents positions are represented by triangles and initialized to be on the boundary ∂Q . The last frame also shows the approximating polygon.	51
3.8	ESTIMATE UPDATE AND PURSUIT ALGORITHM. This figure refers to the case of ∂Q being slowly time-varying. In the first plot from the right we shown the error $\max_{i \in \{1, \dots, n_{ip}\}} D_\lambda(p_i, p_{i+1}) - \min_{i \in \{1, \dots, n_{ip}\}} D_\lambda(p_i, p_{i+1})$ vs time. The second plot shows the arc length distances between the four agents.	53
5.1	The figure shows a collection of four beads which are synchronized.	79
5.2	This figure shows that, regardless from where and with which velocities beads i and $i+1$ impact, the order of the beads is preserved. The velocities in the figure are the velocities after the impact. The speed ν is just the average value of ν_i and ν_{i+1} before the impact.	87
5.3	This figure shows how the speeds of bead i and $i + 1$ change they are traveling towards each other. Note that bead i is early with respect to bead $i + 1$	92

5.4	This figure shows the periodic orbit described in Theorem 5.2. The white circles are the positions of beads $i - 2$, $i - 1$, i , and $i + 1$ when i and $i + 1$ meet at $U_i - \delta$. The black dots are the locations of the impacts for any two neighboring beads. Note that bead $i - 1$ and $i - 2$ are moving towards each other. Because bead $i - 2$ is in its desired sweeping arc, its speed is $\bar{\nu}$ while $i - 1$ is moving away from it and therefore its speed is $f\bar{\nu}$. The same holds for i and $i + 1$ respectively.	96
5.5	This figure illustrates $\mathcal{G}(t)$ for $t \in [t_{1,2}, t_{1,2} + 2\frac{2\pi}{N}\frac{1}{\bar{\nu}}]$ and the time at which each edge appears for $N = 5$ and $\sum_{i=1}^N d_i(0) = -1$ when unbalanced synchrony is reached.	98
5.6	This figure shows how the speeds of bead 1 and 2 change as they are traveling towards each other, shortly after bead 1 meets bead N	99
5.7	From top to bottom, the figure illustrates the position of \tilde{C}_{i-1} , \tilde{C}_i , and of $U_{i-1} - \delta - \Delta$ for $\delta < \frac{\pi}{N}$ and $\delta > \frac{\pi}{N}$	103
5.8	This figure shows θ_i vs time, obtained by implementing the SYNCHRONIZATION ALGORITHM with $N = 8$ beads, the beads are randomly positioned on \mathbb{T} , $\nu_i(0)$ uniformly distributed $\in]0, 1]$, $d_1(0) = d_2(0) = d_4(0) = d_6(0) = +1$, and $f = 0.7$. The positions of the beads 2, 4, 6, 8 are represented by solid lines, while the dash line, dash-dot line, point line, and thicker dash line represent the positions of beads 1, 3, 5, 7.	108
5.9	This figure shows $\max_i \nu_i - \min_i \nu_i$ vs time, obtained by implementing the SYNCHRONIZATION ALGORITHM with $N = 8$ beads, the beads are randomly positioned on \mathbb{T} , $\nu_i(0)$ uniformly distributed $\in]0, 1]$, $d_1(0) = d_2(0) = d_4(0) = d_6(0) = +1$, and $f = 0.7$	109
5.10	This figure shows $\theta_5(t)$ (solid line), $U_5(t)$ (thicker solid line), and $L_5(t)$ (dash-dot line), obtained by implementing the SYNCHRONIZATION ALGORITHM with $N = 8$ beads, the beads are randomly positioned on \mathbb{T} , $\nu_i(0)$ uniformly distributed in $]0, 1]$, $d_1(0) = d_2(0) = d_4(0) = d_6(0) = +1$, and $f = 0.7$	110
5.11	This figure shows θ_i vs time, obtained by implementing the SYNCHRONIZATION ALGORITHM with $N = 7$ beads, the beads are randomly positioned on \mathbb{T} , $\nu_i(0)$ uniformly distributed $\in]0, 1]$, $d_1(0) = d_4(0) = d_5(0) = d_7(0) = -1$, and $f = 0.6$. The positions of the beads 2, 4, 6 are represented by solid lines, while the dash line, dash-dot line, point line, and thicker dash line represent the positions of beads 1, 3, 5, 7.	111

- 5.12 This figure shows $\theta_3(t)$ (solid line), $U_3(t)$ (thicker solid line), and $L_3(t)$ (dash-dot line), obtained by implementing the SYNCHRONIZATION ALGORITHM with $N = 7$ beads, the beads are randomly positioned on \mathbb{T} , $\nu_i(0)$ uniformly distributed in $]0, 1]$, $d_1(0) = d_4(0) = d_5(0) = d_7(0) = -1$, and $f = 0.6$ 112
- 5.13 This figure shows θ_i vs time, obtained by implementing the SYNCHRONIZATION ALGORITHM with $N = 12$ beads, the beads are randomly positioned on \mathbb{T} , $\nu_i(0)$ uniformly distributed $\in]0, 1]$, $d_1(0) = d_2(0) = d_4(0) = d_6(0) = d_7(0) = d_9(0) = d_{12}(0) = -1$, and $f = 0.84$. The positions of the beads 2, 4, 6, 8, 10, 12 are represented by solid lines, while the dash line, dash-dot line, point line, and thicker dash line represent the positions of beads 1, 3, 5, 7, 9, 11. 115
- 5.14 This figure shows $\theta_3(t)$ (solid line), $U_3(t)$ (thicker solid line), and $L_3(t)$ (dash-dot line), obtained by implementing the SYNCHRONIZATION ALGORITHM with $N = 12$ beads, the beads are randomly positioned on \mathbb{T} , $\nu_i(0)$ uniformly distributed in $]0, 1]$, $d_1(0) = d_2(0) = d_4(0) = d_6(0) = d_7(0) = d_9(0) = d_{12}(0) = -1$, and $f = 0.84$. 116
- 5.15 This figure shows θ_i vs time, obtained by implementing the SYNCHRONIZATION ALGORITHM with $N = 12$ beads, the beads are randomly positioned on \mathbb{T} , $\nu_i(0)$ uniformly distributed $\in]0, 1]$, $d_1(0) = d_4(0) = d_6(0) = d_7(0) = d_8(0) = d_9(0) = d_{10}(0) = -1$, and $f = 0.87$. The positions of the beads 2, 4, 6, 8, 10, 12 are represented by solid lines, while the dash line, dash-dot line, point line, and thicker dash line represent the positions of beads 1, 3, 5, 7, 9, 11. 117
- 5.16 This figure shows $\theta_3(t)$ (solid line), $U_3(t)$ (thicker solid line), and $L_3(t)$ (dash-dot line), obtained by implementing the SYNCHRONIZATION ALGORITHM with $N = 12$ beads, the beads are randomly positioned on \mathbb{T} , $\nu_i(0)$ uniformly distributed in $]0, 1]$, $d_1(0) = d_4(0) = d_6(0) = d_7(0) = d_8(0) = d_9(0) = d_{10}(0) = -1$, and $f = 0.87$. 118

Chapter 1

Introduction

Advances in technology have led to miniaturizations of sensors and electronic components and, therefore, have led the fantasy of the researchers run across childhood prairies: fleets of small robots that cooperatively and autonomously accomplishing a task. Some examples of tasks to accomplish are: monitoring and patrolling of sensitive boundaries such as an oil spill in the ocean or a forest fire. In this thesis we propose (i) algorithms enabling a robotic sensor network to optimally estimate a planar contour and (ii) an algorithm that will steer a robotic sensor network to synchronously patrol a planar contour.

1.1 Statement of contribution

The contribution of this work can be summarized as follows:

- In this work our goal is to design distributed algorithms to optimally approximate smooth planar bodies (both convex and nonconvex) by a poly-

gon. By distributed we mean that the algorithms can be implemented by a fleet of robots which have communication constraints and therefore do not have global knowledge of the planar body to approximate. Our interest is motivated by all the possible applications for such algorithms, for example: environmental monitoring, data compression. We first consider smooth strictly convex bodies and three simple metrics. Given n points in fact, it is natural to define an enclosed (i.e., inscribed) polygon and an enclosing (i.e., circumscribed) polygon to the contour. Here the faces of the enclosing polygon are subsets of the tangent lines to the strictly convex contour. The first two metrics we consider are the difference between the area enclosed in the contour and the following areas: the inner polygon area and the outer polygon area. The third metric is the difference between the area of the outer polygon and the area of the inner polygon. We derive the expressions, two of which are novel contributions, of the error metrics as functions of the vertex positions of the approximating polygon. We propose three gradient descent laws for n points to dynamically construct the optimal approximating polygon. The laws are distributed with respect to the positions of the vertices of the approximating polygon. This makes the algorithms suitable for implementation by a robotic sensor network.

- Although the results on the gradient descent laws are encouraging, the extensions to nonconvex and possibly time-varying boundaries pose a number of challenges and they will still require a large number of agents. We then propose and analyze an algorithm to handle generally smooth nonconvex boundaries that requires far fewer agents. The agents rely only on sensed local information to position some interpolation points defining

an approximating polygon. This new algorithm distributes the vertices of the approximating polygon uniformly along the boundary. The notion of uniform placement relies on a metric inspired by approximation theory for convex bodies. The algorithm is provably convergent for static boundaries and efficient for slowly-moving boundaries because of certain input-to-state (ISS) stability properties. These properties are consequences of the fact that the algorithm can be seen as the combination of two consensus algorithms. Consensus algorithms, in their simplest form, are distributed algorithms that a team of autonomous agents can use to reach agreement on something without the help of a central authority. They are, however, more versatile than simply devices for reaching agreement. In fact, the analysis of several algorithms proposed in this dissertation have benefited from posing them as consensus-type algorithms.

- Consensus algorithms are widely studied and used so, starting from the ISS properties of the boundary estimation algorithm, we move on to study general robustness properties of discrete-time linear consensus algorithms with respect to measurement and communication noise for different topologies. For consensus algorithms with inputs, we establish integral input-to-state neutral stability (iISnS) under general assumptions and input-to-state stability (ISS) when a leader node is not subject to noise. Introducing appropriate error outputs, input-to-output stability (IOS) can be established via two alternate routes. By analyzing the contraction over time of the maximum minus the minimum value of the agents' states, we obtain specific gain values that characterize how close the trajectories of each node will be to each other.

- Finally we propose a discrete-time algorithm to synchronize a collection of agents moving back and forth on a unit circle. Each agent or “bead” changes direction upon encountering another bead moving in the opposite direction. Communication is sporadic: only when two beads come sufficiently close they are able to exchange information. This allows agents to update their state. Our analysis makes use of consensus algorithms tools and it guarantees local asymptotic stability of the synchronous behavior.

1.2 Organization

The thesis is organized as follows: in Chapter II we introduce and analyze gradient descent laws to optimally approximate a smooth strictly convex contour, in Chapter III we discuss a distribute algorithm to optimally approximate a smooth nonconvex contour, in Chapter IV we illustrate some robustness properties of linear discrete-time consensus algorithms, and finally, in Chapter V we introduce and analyze a switching law that enables a fleet of robots to synchronously patrol a planar boundary. The conclusions are summarized in Chapter VI.

Chapter 2

Gradient algorithms for polygonal approximation of convex contours

Constructing an optimal polygonal approximation of a contour has been a research subject for mathematicians and engineers across the last three centuries. Still interesting problems continue to remain unsolved especially for the general setting of non-convex bodies. Boundary estimation and tracking is also a relevant problem in computer vision [22]. Some references on the boundary estimation problem for robotic sensor networks include [28, 8, 7, 43]. A final motivation for this work is the interest in dynamical systems that solve optimization problems, as described for example in [16]; discrete-time gradient systems and discrete-time balancing algorithms for networks of agents are discussed in [1] and in [36].

As pointed out by the authors in [21], in the XIX century it was known how to geometrically characterize the polygon enclosed in a convex body that

minimizes the area difference between itself and the enclosing convex body. On the other hand, the geometric characterization of a polygon, enclosing a given strictly convex body, that again minimizes the difference of the areas is more complex and less intuitive. To the best of our knowledge, the earliest reference on this matter appeared only in 1949 by E. Trost, see [41]. In the XX century it was also proved that for a convex planar body the approximation error, for various useful metrics, goes to zero as $1/N^2$, where N is the number of vertices of the interpolating polygon. For a detailed list of references we refer to the surveys [13] and [14]. In this chapter we introduce three intuitive metrics and corresponding gradient flows to approximate a strictly convex body by a polygon of N vertices placed on the boundary. The metrics are: (i) the difference between the area of body and the enclosed polygon, (ii) the difference between the area of the enclosing polygon and the area of the body, and (iii) the difference between the area of the enclosing polygon and of the enclosed polygon. The gradient flows for (ii) and (iii) are novel results. The flows can be implemented, in a distributed fashion, by a robotic sensor network. For the metrics (i) and (ii) we also introduced a corresponding discrete-time algorithm.

2.1 Problem setup

Let $\bar{\mathbb{R}} = \mathbb{R} \cup \{+\infty, -\infty\}$. Let $Q \subseteq \mathbb{R}^2$ be a bounded, strictly convex body with a twice differentiable boundary ∂Q . Let $\mathbb{T} \subseteq \mathbb{R}^2$ denote the unit circle. We parameterize ∂Q by a map $\gamma: \mathbb{T} \rightarrow \partial Q$, and represent its signed curvature by $\kappa: \mathbb{T} \rightarrow \mathbb{R}$. Note that κ remains positive as we traverse the curve γ in a counter-clockwise manner. For $s_i \in \mathbb{T}$, $i \in \{1, \dots, N\}$, let $p_i = \gamma(s_i) = (x_i, y_i) \in \partial Q$ be

the position of N points on the boundary ordered in counter-clockwise direction. We assume $N \geq 3$ and use the identification $0 \equiv N$ and $N + 1 \equiv 1$. For $s \in \mathbb{T}$, let $\mathbf{t}(s)$ and $\mathbf{n}(s)$ denote the tangent vector $\gamma'(s)$ and the unit outward normal vector at $\gamma(s) \in \partial Q$. With a slight abuse of notation, we sometimes refer to unit tangent and normal vectors at the point p_i as \mathbf{t}_i and \mathbf{n}_i , and at the point $p \in \partial Q$ as $\mathbf{t}(p)$ and $\mathbf{n}(p)$. For $p \in \partial Q$, define the half-plane $\mathcal{H}(p) = \{z \in \mathbb{R}^2 \mid (p - z) \cdot \mathbf{n}(p) \leq 0\}$; see Figure 2.1. Given two points A and B , let \overline{AB} denote the segment between them.

Definition 2.1 (Inner and outer polygons) *Let p_1, \dots, p_N be the positions of N points on ∂Q and let $\mathcal{P}(\mathbb{R}^2)$ denote the parts of \mathbb{R}^2 . Let us define $P_I: (\partial Q)^N \rightarrow \mathcal{P}(\mathbb{R}^2)$ by $P_I(p_1, \dots, p_n) = \text{co}(p_1, \dots, p_n)$, the inner polygon generated by the vertices $\{p_1, \dots, p_n\}$. With a slight abuse of notation, let us define the possibly unbounded outer polygon $P_O: (\partial Q)^N \rightarrow \mathcal{P}(\mathbb{R}^2)$ by $P_O(p_1, \dots, p_N) = \mathcal{H}(p_1) \cap \dots \cap \mathcal{H}(p_N)$.*

Definition 2.2 (Tangent lines and tangent connections) *Define the rays $\ell^+: \partial Q \rightarrow \mathcal{P}(\mathbb{R}^2)$ and $\ell^-: \partial Q \rightarrow \mathcal{P}(\mathbb{R}^2)$ by $\ell^+(p) = \{p + \lambda \mathbf{t}(p) \mid \lambda \geq 0\}$ and $\ell^-(p) = \{p + \lambda \mathbf{t}(p) \mid \lambda \leq 0\}$, respectively. Also, let $\ell(p) = \ell^+(p) \cup \ell^-(p)$. A pair (p, q) of points in ∂Q is counter-clockwise tangent-connected (abbreviated cc-tangent-connected) if $\ell^+(p) \cap \ell^-(q) \neq \emptyset$.*

The following result is illustrated in Figure 2.1.

Lemma 2.1 (Bounded outer polygon) *All pairs (p_i, p_{i+1}) , $i \in \{1, \dots, N\}$, are cc-tangent-connected if and only if $P_O(p_1, \dots, p_N)$ is bounded.*

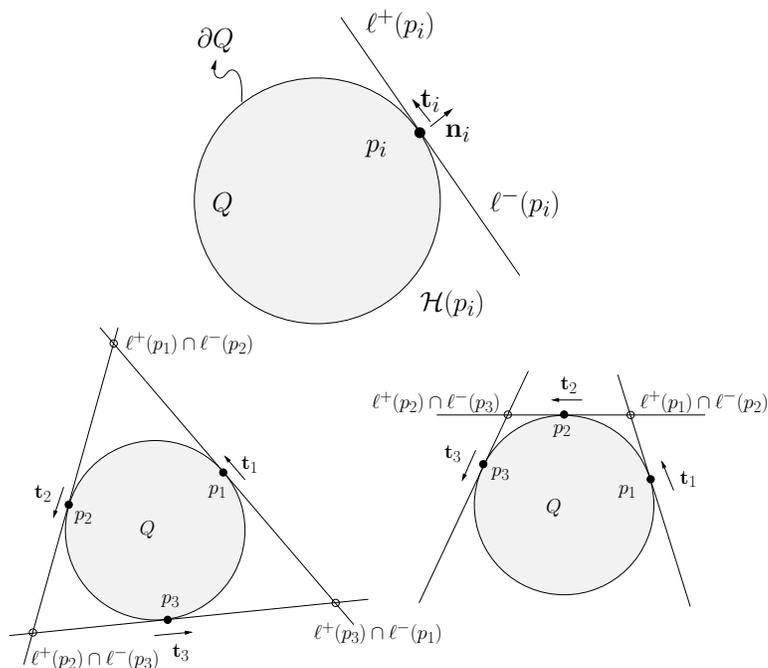


Figure 2.1. From left to right: the half-plane $\mathcal{H}(p_i)$ and its boundary $\ell(p_i) = \ell^+(p_i) \cup \ell^-(p_i)$, three points defining a bounded outer polygon, and three points defining an unbounded outer polygon.

Definition 2.3 (Error metrics) *We quantify the approximation error of Q through three different metrics:*

- *The inner set approximation error $E_I: (\partial Q)^N \rightarrow \mathbb{R}_+$ is defined by*

$$E_I(p_1, \dots, p_n) = \text{Area}(Q \setminus P_I(p_1, \dots, p_n)).$$
- *The outer set approximating error $E_O: (\partial Q)^N \rightarrow \bar{\mathbb{R}}_+$ is defined by*

$$E_O(p_1, \dots, p_n) = \text{Area}(P_O(p_1, \dots, p_n) \setminus Q).$$
- *The symmetric difference error $E_S: (\partial Q)^N \rightarrow \bar{\mathbb{R}}_+$ is defined by*

$$E_S(p_1, \dots, p_n) = \text{Area}(P_O(p_1, \dots, p_n) \setminus P_I(p_1, \dots, p_n)).$$

Remark 2.1 (Implementation by group of robots) In what follows we present descent algorithms for the minimization of these error metrics. The algorithms can be implemented by group of robots where we regard p_i as a robot that can sense a portion of ∂Q , communicate with some robots and move to improve the approximation of ∂Q . For all the algorithms that follow we establish how much sensing and communication are required. •

2.2 Inner-polygon approximation algorithms

The algorithms of this section are based on the interpolation error E_I . Observe that $E_I(p_1, \dots, p_N) = \text{Area}(Q) - \text{Area}(P_I(p_1, \dots, p_N))$. Recalling that the set of points $\{p_1, \dots, p_N\}$ is ordered in counter-clockwise direction, and that (x_i, y_i) are coordinates of p_i , then an expression for $\text{Area}(P_I(p_1, \dots, p_N))$ is:

$$\text{Area}(P_I(p_1, \dots, p_N)) = \frac{1}{2} \sum_{i=1}^N (x_i y_{i+1} - x_{i+1} y_i).$$

We now define a dynamical system by projecting the i th component of the gradient of E_I on the tangent \mathbf{t}_i :

$$\begin{aligned} \dot{p}_i &= \left(\mathbf{t}_i \cdot \frac{\partial \text{Area}(P_I(p_1, \dots, p_N))}{\partial p_i} \right) \mathbf{t}_i \\ &= \left(\frac{1}{2} \mathbf{t}_i^T \begin{pmatrix} y_{i+1} - y_{i-1} \\ x_{i-1} - x_{i+1} \end{pmatrix} \right) \mathbf{t}_i, \quad i \in \{1, \dots, N\}. \end{aligned} \quad (2.1)$$

Lemma 2.2 (Gradient flow for E_I) *If $t \mapsto \eta(t) = (p_1(t), \dots, p_N(t))$ denotes a trajectory of the dynamical system (2.1), then $E_I \circ \eta$ is monotonic non-increasing and η converges asymptotically to the set of critical configurations of E_I . A configuration p_1, \dots, p_N is critical for E_I if and only if, for all $i \in \{1, \dots, N\}$,*

$$\mathbf{t}_i \perp \begin{pmatrix} y_{i+1} - y_{i-1} \\ x_{i-1} - x_{i+1} \end{pmatrix}, \quad (2.2)$$

that is, $\mathbf{n}_i \perp (p_{i+1} - p_{i-1})$. Furthermore, if the boundary ∂Q is analytic, then η converges asymptotically to a critical configuration.

Proof. It is easy to see that $\dot{p}_i(t) = -\frac{\partial E_I(t)}{\partial p_i}|_{\partial Q}$ therefore (2.1) is a gradient system. As a consequence, E_I is monotonic non-increasing:

$$\frac{dE_I}{dt} = -\frac{\text{Area}(\mathcal{P}_I(p_1, \dots, p_N))}{dt} = -\sum_{i=1}^N \left(\mathbf{t}_i \cdot \frac{\partial \text{Area}(P_I(p_1, \dots, p_N))}{\partial p_i} \right)^2 \mathbf{t}_i \leq 0,$$

and the p_i 's asymptotically converge to the set of critical configurations of E_I . If the boundary ∂Q is analytic, then E_I is analytic (because it is a composition of analytic functions) and, by [27], we can conclude that every trajectory has finite length and tends to a single point belonging to the set of critical configurations. ■

Not every critical point of E_I is an extremum of E_I : Figure 2.2 illustrates a saddle point of E_I .

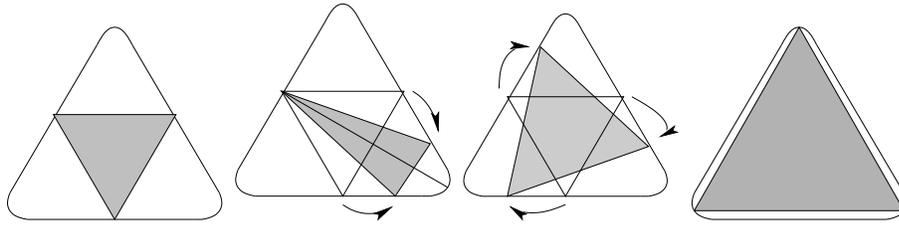


Figure 2.2. From left to right: saddle point configuration, nearby configuration that increases the error E_I , bear by configuration that decreases the error E_I , configuration corresponding to a minimum error configuration.

Remark 2.2 (Historical notes) The characterization (2.2) of the critical configurations was already obtained in the XIX century according to [21]. The paper [21] additionally shows how the critical point configurations satisfy the condition that points remain closer in regions of higher mean curvature, which is a desirable condition for shape representation. It is believed [13] that as the number of nodes increases, the type of configurations that satisfy (2.2) correspond only to global error minima. •

Remark 2.3 (Implementation by group of robots) In the dynamical system (2.1), the velocity \dot{p}_i depends only on p_{i-1} , p_{i+1} , and \mathbf{t}_i . Therefore, to implement this velocity control, every robot has to receive information about the positions of its immediate clockwise and counter-clockwise neighbors and sense the gradient of the contour at its position. Clearly, the communication graph is a ring graph. •

2.2.1 Discrete-time inner-polygon approximation algorithms

Here we present two discrete-time versions of the vector field in equation (2.1).

Given a strictly convex set Q , define $q_{\max} : (\partial Q)^2 \rightarrow \partial Q$ as follows: $q_{\max}(q_1, q_2)$ is the point of the counter-clockwise arc from q_1 to q_2 whose tangent to ∂Q is parallel to the segment $\overline{q_1 q_2}$. Note that $q_{\max}(q_1, q_2)$ maximizes $q \mapsto \text{Area}(P_I(q_1, q, q_2))$.

Algorithm 1. At each discrete time instant $k \in \mathbb{N}$ and for each node $i \in \{1, \dots, N\}$ define:

$$p_i(k+1) = \begin{cases} q_{\max}(p_{i-1}(k), p_{i+1}(k)), & \text{if } i \equiv k \pmod{N}, \\ p_i(k), & \text{if } i \not\equiv k \pmod{N}. \end{cases} \quad (2.3)$$

Proposition 2.1 (Convergence of Algorithm 1) *If*

$k \mapsto \eta(k) = (p_1(k), \dots, p_N(k))$ *denotes a trajectory of the dynamical system (2.3), then $E_I \circ \eta$ is monotonic non-increasing and η converges asymptotically to the set of critical configurations of E_I .*

Proof. Let $A_k = \text{Area}(P_I(p_1(k), \dots, p_N(k)))$ and let i be congruent \pmod{N} with k . We have that $A_k = T_k + \bar{A}_k$, where $T_k = \text{Area}(P_I(p_{i-1}(k), p_i(k), p_{i+1}(k)))$ and \bar{A}_k is the area of the inner polygon generated by the complementary set of nodes. Since $q_{\max}(q_1, q_2)$ maximizes $q \mapsto \text{Area}(P_I(q_1, q, q_2))$,

$T_k = \text{Area}(P_I(p_{i-1}(k), p_i(k), p_{i+1}(k))) \leq \text{Area}(P_I(p_{i-1}(k), p_i(k+1), p_{i+1}(k))) = \bar{T}_{k+1}$. Therefore, $A_k = T_k + \bar{A}_k \leq \bar{T}_{k+1} + \bar{A}_k = A_{k+1}$, i.e., the point that moves at time k does so in order to increase the area of the inner polygon or, equivalently, to decrease the error E_I . Using the extension of the LaSalle Invariance Principle for discrete-time systems ([25]) we can claim that the p_i 's will

asymptotically reach the largest weakly invariant set in $\{(p_1(k), \dots, p_N(k)) \in \partial Q^N | E_I \circ \eta(k+1) - E_I \circ \eta(k) = 0\}$, which is the set of critical configurations for E_I . ■

The quintuplet $(p_{-2}, p_{-1}, p_0, p_1, p_2)$ is *admissible* if the following three inequalities hold:

$$\begin{aligned} & \text{Area}(P_I(q_{\max}(p_{-2}, p_0), p_0, q_{\max}(p_0, p_2))) \\ & \leq \text{Area}(P_I(q_{\max}(p_{-2}, p_0), q_{\max}(p_{-1}, p_1), q_{\max}(p_0, p_2))), \\ & \text{Area}(P_I(p_{-1}, p_0, q_{\max}(p_0, p_2))) \leq \text{Area}(P_I(p_{-1}, q_{\max}(p_{-1}, p_1), q_{\max}(p_0, p_2))), \\ & \text{Area}(P_I(q_{\max}(p_{-2}, p_0), p_0, p_1)) \leq \text{Area}(P_I(q_{\max}(p_{-2}, p_0), q_{\max}(p_{-1}, p_1), p_1)). \end{aligned}$$

Algorithm 2. At each discrete time instant $k \in \mathbb{N}$ and for each node $i \in \{1, \dots, N\}$ define:

$$p_i(k+1) = q_{\max}(p_{i-1}(k), p_{i+1}(k)), \quad (2.4)$$

if $(p_{i-2}(k), p_{i-1}(k), p_i(k), p_{i+1}(k), p_{i+2}(k))$ is admissible, and $p_i(k+1) = p_i(k)$ otherwise. Here is our main analysis result in this section.

Proposition 2.2 (Convergence of Algorithm 2) *E_I is monotonic non-increasing along all trajectories of (2.4).*

Proof. The proof consists of two parts. As first fact (i), we prove inductively that the area of any polygon of N vertices increases by leaving any two consecutive nodes fixed and by moving the other $N-2$ vertices according to (2.4). As second fact (ii), building on the previous result, we show that the area of any polygon of N vertices increases by moving the all the N vertices according to (2.4).

Let us prove first (i) by induction on the number of vertices of a polygon. Let us consider $N = 3$. Clearly, if two of the three vertices are fixed and the

other one moves according to (2.4), the area of the triangle formed by the three nodes increases, just as seen for Algorithm 1. Assume now that, given a polygon $P_I(p_1, \dots, p_{N-1})$ with $N - 1$ vertices, its area can be increased by leaving any two consecutive nodes fixed and moving the other $N - 1 - 2$ vertices according to (2.4). Let us now prove that the same property holds for the polygon $P_I(p_1, \dots, p_N)$ with N vertices. Clearly, we have that:

$$\text{Area}(P_I(p_1, \dots, p_N)) = \text{Area}(P_I(p_1, \dots, p_{N-1})) + \text{Area}(P_I(p_{N-1}, p_N, p_1)),$$

where for simplicity of notation we dropped the time index k . By assumption, the area of a polygon with $N - 1$ vertices increases if any two consecutive points are fixed and the rest moves according to (2.4). Therefore, we have

$$\text{Area}(P_I(p_1, p_2 \dots, p_{N-2}, p_{N-1})) \leq \text{Area}(P_I(p_1, p_2^+ \dots, p_{N-2}^+, p_{N-1})),$$

where for simplicity of notation the superscript $+$ indicates that the node has updated its position according to (2.4). This implies:

$$\begin{aligned} & \text{Area}(P_I(p_1, \dots, p_N)) \\ & \leq \text{Area}(P_I(p_1, p_2^+ \dots, p_{N-2}^+, p_{N-1})) + \text{Area}(P_I(p_{N-1}, p_N, p_1)) \\ & = \text{Area}(P_I(p_2^+ \dots, p_{N-2}^+, p_{N-1}, p_N)) + \text{Area}(P_I(p_N, p_1, p_2^+)) \\ & \leq \text{Area}(P_I(p_2^+ \dots, p_{N-2}^+, p_{N-1}, p_N)) + \text{Area}(P_I(p_N, p_1^+, p_2^+)) \\ & = \text{Area}(P_I(p_1^+, p_2^+, \dots, p_{N-2}^+, p_{N-1}, p_N)). \end{aligned}$$

The second inequality holds because along the trajectories of (2.4) we have that $\text{Area}(P_I(p_N, p_1, p_2^+)) \leq \text{Area}(P_I(p_N, p_1^+, p_2^+))$. This concludes the proof of (i). To

prove (ii), note that

$$\begin{aligned}
& \text{Area}(P_I(p_1^+, p_2^+, \dots, p_{N-2}^+, p_{N-1}, p_N)) \\
&= \text{Area}(P_I(p_1^+, \dots, p_{N-2}^+, p_{N-1})) + \text{Area}(P_I(p_{N-1}, p_N, p_1^+)) \\
&\leq \text{Area}(P_I(p_1^+, \dots, p_{N-2}^+, p_{N-1})) + \text{Area}(P_I(p_{N-1}, p_N^+, p_1^+)) \\
&= \text{Area}(P_I(p_N^+, p_1^+, \dots, p_{N-2}^+)) + \text{Area}(P_I(p_{N-2}^+, p_{N-1}, p_N^+)) \\
&\leq \text{Area}(P_I(p_N^+, p_1^+, \dots, p_{N-2}^+)) + \text{Area}(P_I(p_{N-2}^+, p_{N-1}^+, p_N^+)) \\
&= \text{Area}(P_I(p_1^+, \dots, p_N^+)).
\end{aligned}$$

■

Remark 2.4 Stationary configurations of (2.4) are not necessarily critical points of E_I , i.e., at an equilibrium configuration for (2.4) there could exist a node for which condition (2.2) is not satisfied. A set of nodes could be “unlocked” by running a leader-election algorithm between neighbors and giving priority of motion to the consensual leader. This operation respects the descent nature of the algorithm and guarantees that we reach a desired critical configuration. •

Remark 2.5 (Implementation by group of robots) To implement Algorithm 1, each robot p_i needs to have knowledge about its own label number $i \in \{1, \dots, N\}$ and about the position of its one-hop neighbors. Algorithm 2, does not require a labeling of robots, but requires each robot to have knowledge about part of the contour and knowledge about the position of its two-hop neighbors. •

2.3 Outer-polygon approximation algorithms

The algorithms of this section are based on the interpolation error E_O . We begin with a geometric characterization of the partial derivative of E_O and of the critical configurations for E_O . First, for $i \in \{1, \dots, N\}$, we define $\alpha_i(p_i, p_{i+1})$ to be the angle (measured in counter-clockwise order) from \mathbf{t}_i to \mathbf{t}_{i+1} . Assuming any pair (p_i, p_{i+1}) is cc-tangent-connected, consider $A_i = \ell^-(p_i) \cap \ell^+(p_{i-1})$ and $B_i = \ell^-(p_{i+1}) \cap \ell^+(p_i)$. Let us denote by d_i^- (resp. d_i^+) the length of the segment $\overline{p_i A_i}$ (resp. the segment $\overline{p_i B_i}$), as in Figure 2.3(a). It is useful to define $d_i^- = +\infty$

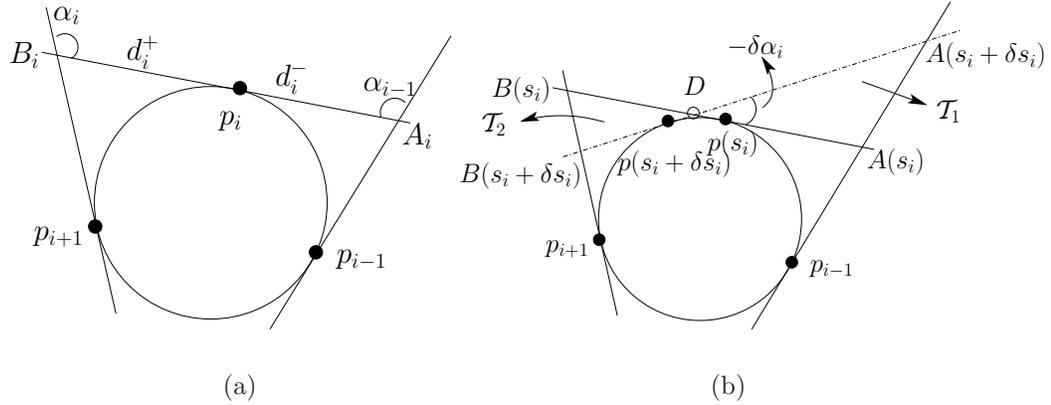


Figure 2.3. (a) Illustration of α_i , α_{i-1} , A_i and B_i . (b) Variation of E_O described in Proposition 2.3.

(resp. $d_i^+ = +\infty$) when the pair (p_{i-1}, p_{i+1}) is not cc-tangent-connected. One can

show that:

$$d_i^-(p_i, p_{i-1}) = \begin{cases} +\infty, & \text{if } p_i \neq p_{i-1} \\ & \text{and } \mathbf{t}_i \cdot \mathbf{n}_{i-1} = 0, \\ \frac{(p_i - p_{i-1}) \cdot \mathbf{n}_{i-1}}{\mathbf{t}_i \cdot \mathbf{n}_{i-1}}, & \text{otherwise,} \end{cases}$$

$$d_i^+(p_i, p_{i+1}) = \begin{cases} +\infty, & \text{if } p_i \neq p_{i+1} \\ & \text{and } \mathbf{t}_i \cdot \mathbf{n}_{i+1} = 0, \\ \frac{(p_{i+1} - p_i) \cdot \mathbf{n}_{i+1}}{\mathbf{t}_i \cdot \mathbf{n}_{i+1}}, & \text{otherwise.} \end{cases}$$

Proposition 2.3 (Partial derivative of E_O) *If all the pairs (p_i, p_{i+1}) are cc-tangent-connected, then*

$$\frac{\partial E_O(p_1, \dots, p_n)}{\partial s_i} = \frac{1}{2}(d_i^- + d_i^+)(d_i^- - d_i^+)\kappa(s_i).$$

Proof. Let us consider $p(s_i)$ and $p(s_i + \delta s_i)$, two points on the arc from p_{i-1} to p_{i+1} , as shown in Figure 2.3(b). Let $D = \ell^+(s_i) \cap \ell^-(s_i + \delta s_i)$. Let $\delta\alpha_i = \alpha(s_i + \delta s_i) - \alpha(s_i)$. Note that $\delta\alpha_i < 0$ when $\delta s_i > 0$. By construction:

$$\text{Area}(\mathcal{T}_1) - \text{Area}(\mathcal{T}_2) = E_O(p_1, \dots, p(s_i + \delta s_i), \dots, p_n) - E_O(p_1, \dots, p(s_i), \dots, p_n),$$

where \mathcal{T}_1 is the triangle with vertices D , $A(s_i)$, and $A(s_i + \delta s_i)$, and \mathcal{T}_2 is the triangle with vertices D , $B(s_i)$, and $B(s_i + \delta s_i)$. We now prove that $\text{Area}(\mathcal{T}_1) = (d_i^-)^2(-\delta\alpha_i) + o(\delta\alpha_i^2)$ and that $\text{Area}(\mathcal{T}_2) = (d_i^+)^2(-\delta\alpha_i) + o(\delta\alpha_i^2)$, for small δs_i . Note that $\|p(s_i + \delta s_i) - p(s_i)\| = \|\gamma'(s_i)\delta s_i\| + o(\delta s_i^2)$ and that $\frac{\partial\alpha(s)}{\partial s} = \frac{\partial\alpha(s)}{\partial \mathbf{t}(s)} \frac{\partial \mathbf{t}(s)}{\partial s} = -\kappa(s)$. Therefore, we have $\delta\alpha_i = -\kappa(s_i)\delta s_i + o(\delta s_i^2)$. In fact, provided that $p_i \neq p_{i+1}$, the function $\alpha_i(p_i, p_{i+1})$ is differentiable and its gradient is: $\frac{\partial\alpha_i(p_i, p_{i+1})}{\partial \mathbf{t}_i} = (t_i^2, -t_i^1)$, where $\mathbf{t}_i = (t_i^1, t_i^2)$. Clearly, $\frac{\partial \mathbf{t}_i}{\partial s_i} = (-t_i^2, t_i^1)\kappa(s_i)$. It can be shown that

$\|D - A(s_i)\| = d_i^- + \frac{\|\gamma'(s_i)\delta s_i\|}{2} + o(\delta s_i^2)$. Let h be the height of the triangle \mathcal{T}_1 with respect to the base $\|D - A(s_i)\|$. Clearly, we have $h = \|A(s_i) - A(s_i + \delta s_i)\| \sin(\alpha_{i-1})$ and

$$\frac{\|A(s_i) - A(s_i + \delta s_i)\|}{\sin(-\delta\alpha_i)} = \frac{\|D - A(s_i)\|}{\sin(\pi - (\alpha_{i-1} - \delta\alpha_i))},$$

and, therefore,

$$h = \|D - A(s_i)\| \frac{\sin(-\delta\alpha_i)}{\sin(\alpha_{i-1} - \delta\alpha_i)} \sin(\alpha_{i-1}).$$

We have then:

$$\text{Area}(\mathcal{T}_1) = \frac{1}{2} \|D - A(s_i)\| h = \frac{1}{2} \|D - A(s_i)\|^2 \frac{\sin(\alpha_{i-1})}{\sin(\alpha_{i-1} - \delta\alpha_i)} \sin(-\delta\alpha_i).$$

For small δs_i , and hence small $\delta\alpha_i$, we have that $\frac{\sin(\alpha_{i-1})}{\sin(\alpha_{i-1} - \delta\alpha_i)} = 1 + o(\delta\alpha_i)$, and $\sin(-\delta\alpha_i) = -\delta\alpha_i + o(\delta\alpha_i^2)$. Therefore:

$$\text{Area}(\mathcal{T}_1) = \frac{1}{2} (d_i^-)^2 (-\delta\alpha_i) + o(\delta\alpha_i^2).$$

Analogously, it can be proved that $\text{Area}(\mathcal{T}_2) = \frac{1}{2} (d_i^+)^2 (-\delta\alpha_i) + o(\delta\alpha_i^2)$. We can now compute

$$\begin{aligned} \frac{\partial E_O(p_1, \dots, p_n)}{\partial s_i} &= \lim_{\delta s_i \rightarrow 0} \frac{\text{Area}(\mathcal{T}_1) - \text{Area}(\mathcal{T}_2)}{\delta s_i}, \\ &= \lim_{\delta s_i \rightarrow 0} \frac{1}{2} ((d_i^+)^2 - (d_i^-)^2) \frac{\delta\alpha_i}{\delta s_i}, \\ &= \frac{1}{2} (d_i^- + d_i^+) (d_i^- - d_i^+) \kappa(s_i). \end{aligned}$$

where we used the fact that $\partial\alpha_i/\partial s_i = \kappa$. ■

Based on these notions and concepts, we define the dynamical system

$$\dot{p}_i = \text{sat}_v((d_i^+)^2 - (d_i^-)^2) \mathbf{t}_i, \quad i \in \{1, \dots, N\}, \quad (2.5)$$

where the function $\text{sat}_v: \overline{\mathbb{R}} \rightarrow \mathbb{R}$, defined for some positive saturation value $v \in (0, +\infty)$, is given by:

$$\text{sat}_v(x) = \begin{cases} x, & |x| \leq v, \\ \frac{x}{|x|} v, & |x| \geq v. \end{cases}$$

Equation (2.5) is well defined if we adopt the convention $|\pm\infty| = +\infty$, and the usual operations in $\overline{\mathbb{R}}$. We are ready for the main result of this section; note that the characterization of the critical points of E_O was originally given in [41].

Proposition 2.4 (Gradient flow for E_O) *If $t \mapsto \eta(t) = (p_1(t), \dots, p_N(t))$ denotes a trajectory of the dynamical system (2.5), then (i) $E_O \circ \eta$ is bounded in finite time and monotonic non-increasing afterwards, and (ii) η converges asymptotically to the set of critical configurations of E_O . A configuration p_1, \dots, p_N is critical for E_O if and only if, for all $i \in \{1, \dots, N\}$,*

$$d_i^+(p_i, p_{i+1}) = d_i^-(p_i, p_{i-1}).$$

Furthermore, if the boundary ∂Q is analytic, then η converges asymptotically to a critical configuration.

Proof. Suppose that there exists $i \in \{1, \dots, N\}$ such that (p_i, p_{i+1}) is not cc-tangent-connected (i.e., $d_i^- < +\infty$ and $d_i^+ = +\infty$). Since $d_i^+ = +\infty$ also $d_{i+1}^- = +\infty$ and E_O is unbounded. Since ∂Q is strictly convex, we have $d_j^- < +\infty$ (resp. $d_j^+ < +\infty$), for all $j \notin \{i, i+1\}$. Because of equation (2.5), p_i will move counter-clockwise with speed $v > 0$, while p_{i+1} will move clockwise with the same speed. Therefore, in finite time the two rays ℓ_i^+ and ℓ_{i+1}^- intersect, (p_i, p_{i+1}) become cc-tangent-connected and E_O becomes bounded. Now, we prove that

if all pairs (p_i, p_{i+1}) are cc-tangent-connected, then $E_O \circ \eta$ decreases. Using equation (2.5) we compute

$$\begin{aligned}\dot{p}_i &= \gamma'(s_i)\dot{s}_i = \|\gamma'(s_i)\|\mathbf{t}_i\dot{s}_i = \mathbf{t}_i \text{sat}_v((d_i^+)^2 - (d_i^-)^2) \\ \implies \dot{s}_i &= \frac{\text{sat}_v((d_i^+)^2 - (d_i^-)^2)}{\|\gamma'(s_i)\|},\end{aligned}$$

and therefore:

$$\begin{aligned}\frac{dE_O(p_1, \dots, p_N)}{dt} &= \sum_{i=1}^N \frac{\partial E_O(p_1, \dots, p_N)}{\partial \alpha_i} \frac{\partial \alpha_i}{\partial \mathbf{t}_i} \cdot \frac{\partial \mathbf{t}_i}{\partial s_i} \dot{s}_i \\ &= \sum_{i=1}^N \kappa(s_i) \frac{((d_i^-)^2 - (d_i^+)^2) \text{sat}_v((d_i^+)^2 - (d_i^-)^2)}{2\|\gamma'(s_i)\|}.\end{aligned}$$

Because we assumed $\kappa > 0$ on the entire boundary, the cost function E_O decreases monotonically along the trajectories of equation (2.5). Using the LaSalle Invariance Principle, it can be proved that the p_i 's will asymptotically converge to the set of critical configurations for E_O .

Let $\mathbf{s}(t) = [s_1(t), \dots, s_N(t)]^T \in \mathbb{T}^N$, and note that if ∂Q is analytic then E_O is analytic. Next, we recall a result from [1]. If there exists $\delta > 0$ and τ such that, for all $t > \tau$, the following holds

$$\frac{dE_O}{dt} \equiv \langle \nabla E_O(\mathbf{s}(t)), \dot{\mathbf{s}}(t) \rangle \leq -\delta \|\nabla E_O(\mathbf{s}(t))\| \|\dot{\mathbf{s}}(t)\|,$$

then $\mathbf{s}(t)$ converges to a unique critical configuration \mathbf{s}^* . We use this result as follows. Note that as $t \rightarrow +\infty$, $\mathbf{s}(t)$ approaches the set of critical configurations. We can then conclude that there exists a time τ after which sat_v is not active any longer and, hence, $\dot{s}_i(t) = -\lambda_i(t) \frac{\partial E_O}{\partial s_i}$, where $\lambda_i(t) = \frac{2}{\kappa(s_i)\|\gamma'(s_i)\|}$. Therefore we have

$$\langle \nabla E_O(\mathbf{s}(t)), \dot{\mathbf{s}}(t) \rangle = -\nabla E_O(\mathbf{s}(t))^T \Lambda(t) \nabla E_O(\mathbf{s}(t)) \leq -\lambda_{\min}(t) \|\nabla E_O(\mathbf{s}(t))\|^2,$$

where $\Lambda(t) \in \mathbb{R}^{N \times N}$ is a diagonal matrix with entries $[\Lambda(t)]_{ii} = \lambda_i(t) > 0$, and $\lambda_{\min}(t) = \min\{\lambda_1(t), \dots, \lambda_N(t)\}$. We require:

$$-\lambda_{\min}(t)\|\nabla E_O(\mathbf{s}(t))\|^2 \leq -\delta\|\nabla E_O(\mathbf{s}(t))\|\|\dot{\mathbf{s}}(t)\|,$$

or equivalently

$$\lambda_{\min}(t)\|\nabla E_O(\mathbf{s}(t))\| \geq \delta\|\dot{\mathbf{s}}(t)\|.$$

Note that $\|\dot{\mathbf{s}}(t)\| \leq \lambda_{\max}(t)\|\nabla E_O(\mathbf{s}(t))\|$, where $\lambda_{\max}(t) = \max\{\lambda_1(t), \dots, \lambda_N(t)\}$, therefore:

$$\delta = \inf_{t>\tau} \frac{\lambda_{\min}(t)\|\nabla E_O(\mathbf{s}(t))\|}{\|\dot{\mathbf{s}}(t)\|} \geq \inf_{t>\tau} \frac{\lambda_{\min}(t)}{\lambda_{\max}(t)} > 0.$$

We can then conclude that the p_i 's will asymptotically converge to a unique critical configuration for E_O . ■

Remark 2.6 (Implementation by group of robots) To dynamically construct the best outer-polygon approximation according to equation (2.5), the robots need to exchange information not only about their positions (like for the inner-polygon approximation) but also about their local tangent. ●

2.3.1 Discrete-time outer-polygon approximation algorithms

It is easy to prove that an algorithm analogous to Algorithm 1 in the previous section guarantees convergence to the critical configuration of E_O . We state the analogous results here omitting the corresponding proof.

Given a strictly convex set Q , define $q_{\min} : (\partial Q)^2 \rightarrow \partial Q$ as follows: $q_{\min}(q_1, q_2)$ is the point of the counter-clockwise arc from q_1 to q_2 whose tangent to ∂Q satisfies $d_i^- = d_i^+$. Note that $q_{\min}(q_1, q_2)$ minimizes $q \mapsto \text{Area}(P_O(q_1, q, q_2))$.

Algorithm 3. At each discrete time instant $k \in \mathbb{N}$ and for each node $i \in \{1, \dots, N\}$ define:

$$p_i(k+1) = \begin{cases} q_{\min}(p_{i-1}(k), p_{i+1}(k)), & \text{if } i \equiv k \pmod{N}, \\ p_i(k), & \text{if } i \not\equiv k \pmod{N}. \end{cases} \quad (2.6)$$

Proposition 2.5 (Convergence of Algorithm 3) *If*

$k \mapsto \eta(k) = (p_1(k), \dots, p_N(k))$ *denotes a trajectory of the dynamical system (2.6), then $E_O \circ \eta$ is monotonic non-increasing and η converges asymptotically to the set of critical configurations of E_O .*

Remark 2.7 Similarly to Algorithm 2 in the inner-polygon approximation problem, it is possible to design a discrete time algorithm based on admissible quintuplets. Such algorithm would have limitations similar to the ones of Algorithm 2 and we do not present it here in the interest of brevity.

2.4 “Outer minus inner” polygon approximation algorithms

An alternative cost function that quantifies the accuracy of a polygonal approximation of a convex body Q , is provided by the symmetric difference error E_S . In this section we provide a novel expression for $\frac{\partial E_S}{\partial p_i}$, $i \in \{1, \dots, N\}$ under

the assumption that the outer polygon is bounded. This expression leads to a new type of gradient decent algorithm.

Lemma 2.3 (Partial derivative of E_S) *If (p_i, p_{i+1}) is cc-tangent-connected, then the area of the triangle formed by the segment $\overline{p_{i+1}p_i}$ and the rays $\ell^+(p_i)$ and $\ell^-(p_{i+1})$ is*

$$A_i(p_i, p_{i+1}, \mathbf{n}_i, \mathbf{n}_{i+1}) = \frac{1}{2} \frac{(\mathbf{n}_i \cdot (p_i - p_{i+1}))(\mathbf{n}_{i+1} \cdot (p_i - p_{i+1}))}{(\mathbf{n}_i \times \mathbf{n}_{i+1}) \cdot \mathbf{e}_3}.$$

Regarding p_i and $\mathbf{n}_i = \mathbf{n}(p_i)$ as a functions of the parameter $s_i \in [0, 1]$, we have

$$\frac{\partial E_S(p_1, \dots, p_N)}{\partial s_i} = \left(\frac{\partial A_{i-1}}{\partial p_i} + \frac{\partial A_i}{\partial p_i} - \kappa(s_i) \left(\frac{\partial A_{i-1}}{\partial \mathbf{n}_i} + \frac{\partial A_i}{\partial \mathbf{n}_i} \right) \right) \cdot \mathbf{t}_i.$$

Proof. In the interest of space, we only mention here that the proof is based upon elementary calculations. ■

If we set $p_i = (x_i, y_i)$, $\mathbf{n}_i = (n_i^1, n_i^2)$ and $\mathbf{n}_{i-1} \times \mathbf{n}_i^+ := n_{i-1}^1 n_i^2 + n_i^1 n_{i-1}^2$, then explicit expressions for the relevant partial derivatives in Lemma 2.3 are:

$$\begin{aligned} \frac{\partial A_{i-1}}{\partial x_i} &= \frac{(p_i - p_{i-1}) \cdot (2n_{i-1}^1 n_i^1, \mathbf{n}_{i-1} \times \mathbf{n}_i^+)}{2(\mathbf{n}_{i-1} \times \mathbf{n}_i^+)}, \\ \frac{\partial A_{i-1}}{\partial y_i} &= \frac{(p_i - p_{i-1}) \cdot (\mathbf{n}_{i-1} \times \mathbf{n}_i^+, 2n_{i-1}^1 n_i^1)}{2(\mathbf{n}_{i-1} \times \mathbf{n}_i^+)}, \\ \frac{\partial A_{i-1}}{\partial n_i^1} &= \frac{n_i^2 (\mathbf{n}_{i-1} \cdot (p_i - p_{i-1}))^2}{2(\mathbf{n}_{i-1} \times \mathbf{n}_i^+)^2}, \\ \frac{\partial A_{i-1}}{\partial n_i^2} &= \frac{n_i^1 (\mathbf{n}_{i-1} \cdot (p_i - p_{i-1}))^2}{2(\mathbf{n}_{i-1} \times \mathbf{n}_i^+)^2}. \end{aligned}$$

Lemma 2.4 (Gradient flow for E_S) *If $t \mapsto \eta(t) = (p_1(t), \dots, p_N(t))$ denotes a trajectory of the dynamical system*

$$\dot{p}_i = - \mathbf{t}_i \left(\frac{\partial A_{i-1}}{\partial p_i} + \frac{\partial A_i}{\partial p_i} - \kappa(s_i) \left(\frac{\partial A_{i-1}}{\partial \mathbf{n}_i} + \frac{\partial A_i}{\partial \mathbf{n}_i} \right) \right) \cdot \mathbf{t}_i, \quad i \in \{1, \dots, N\},$$

with $E_S \circ \eta(0) < +\infty$, then $E_S \circ \eta$ is monotonic non-increasing and η converges asymptotically to the set of critical configurations of E_S . Furthermore, if the boundary ∂Q is analytic, then η converges asymptotically to a critical configuration.

We omit the proof of this lemma as it closely parallels that of Lemma 2.2.

Remark 2.8 (Implementation by group of robots) Even for this scenarios, the robots can move along the gradient of E_S relying upon information that is available to them through one-hop communication and through sensing of local tangent and curvature data. •

2.5 Simulations

Figure 2.4 shows the implementation results of the three continuous time descent algorithms described in Sections 2.2, 2.3, and 2.4. The eleven nodes are on the contour described by $\gamma(\theta) = (2.1 + \sin(2\pi\theta))(\cos(2\pi\theta), \sin(2\pi\theta))^T$, for $\theta \in [0, 1)$. Figure 2.5 shows the implementation results of the discrete-time Algorithm 2 described in Sections 2.2.

2.6 Summary

In this chapter we have introduced three intuitive metrics and corresponding gradient flows to approximate a strictly convex planar body by a polygon of N vertices. The metrics are: (i) the difference between the area of body and the inner polygon, (ii) the difference between the area of the outer polygon and the

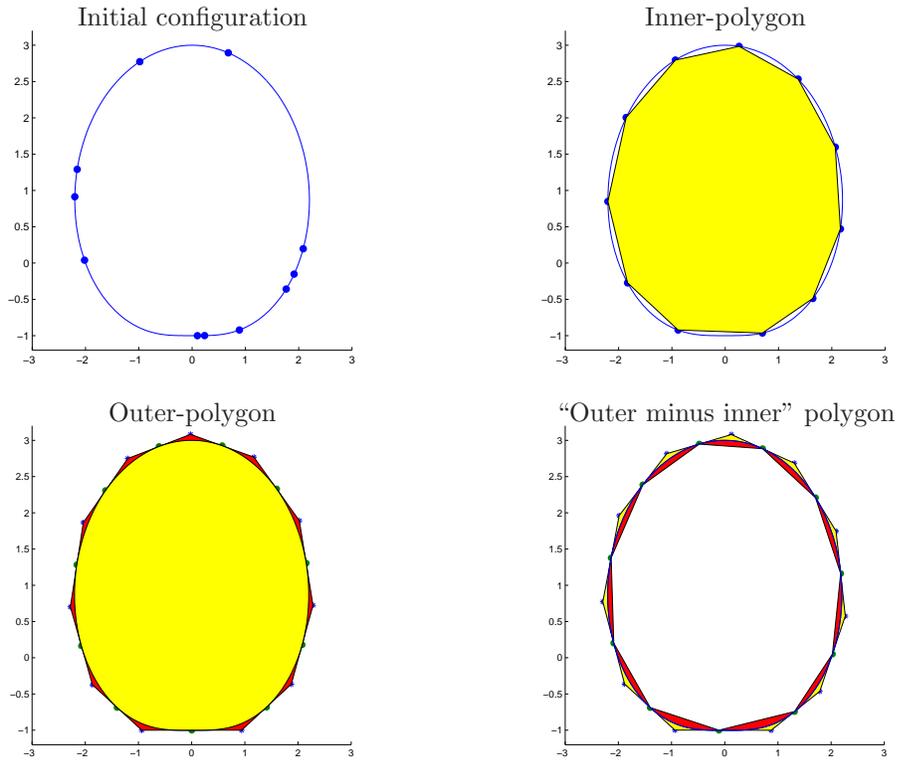


Figure 2.4. From left to right and from top to bottom: initial condition of eleven nodes on a convex boundary, final condition after the implementation of the inner-polygon, outer-polygon, and “outer minus inner” polygon approximation algorithms.

area of the body, and (iii) the difference between the area of the outer polygon and of the inner polygon. The gradient flows for (ii) and (iii) are novel results. The flows can be implemented, in a distributed fashion, by a robotic sensor network. For the metrics (i) and (ii) we also introduced a corresponding discrete-time algorithm.

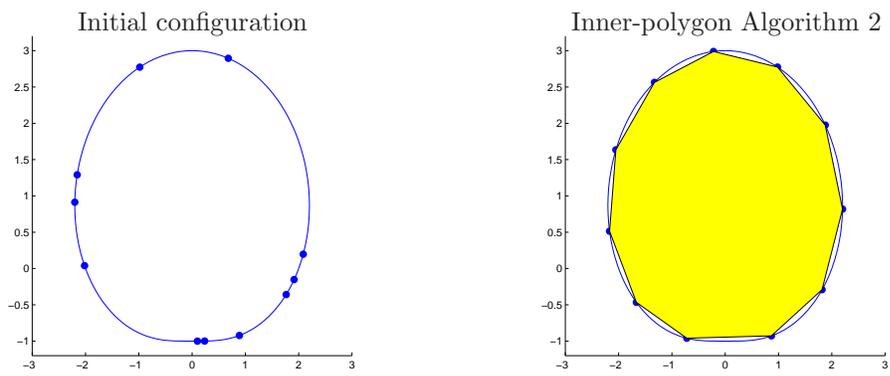


Figure 2.5. From left to right: initial condition of eleven nodes on a convex boundary and final condition after the implementation of Algorithm 2.

Chapter 3

Monitoring environmental boundaries with a robotic sensor network

Much recent attention has been given to the problem of boundary estimation and tracking by means of robotic networks. The common goal is to design a distributed algorithm that allows a limited number of mobile agents to detect the boundary of a region of interest and estimate it as it evolves. Boundary estimation and tracking is useful in numerous applications such as the detection of harmful algae bloom [28, 5], oil spill [8], and fire containment [7]. In [28], Marthaler and Bertozzi adopt the so-called “snake algorithm” (from the computer vision literature) to detect and track the boundary of harmful algae bloom. Each agent is equipped with a chemical sensor that is able to measure the concentration gradient and with a communication system that is able to exchange information with a data fusion center. In [5], Bertozzi *et al.* suggest an algo-

rithm that requires only a concentration sensor: the agents repeatedly cross the region boundary using a bang-bang angular velocity controller. In [8], Clark and Fierro use a random coverage controller, a collision avoidance controller and a bang-bang angular velocity controller to detect and surround an oil spill. In [7], Casbeer *et al.* describe an algorithm that allows Low Altitude Short Endurance Unmanned Vehicles (LASEUVs) to closely monitor the boundary of a fire. Each of the LASEUVs has an infrared camera and a short range communication device to exchange information with other agents and to download the information collected onto the base station. A different approach is considered by Zhang and Leonard in [43]. A formation of four robots tracks at unitary speed the level sets of a field. Their relative position changes so that they optimally measure the gradient and they estimate the curvature of the field in the center of the formation. Challenges in boundary estimation using motion-enabled sensors are discussed in [35].

In this chapter we propose an algorithm to estimate and reconstruct the boundary of a nonconvex time-varying region. The objective is for a group of mobile agents to optimally place some interpolation points on the boundary of a simply connected planar region. The boundary is then reconstructed by linear interpolation of the interpolation points. We assume that (i) at initial time the agents have an estimate of the boundary, (ii) each agent is equipped with a limited-footprint camera-like sensor and with algorithms to locally estimate the tangent and curvature of the boundary, and (iii) the agents exchange information through a ring-topology communication network. An example scenario for these assumptions is a situation where a group of Unmanned Air Vehicles (UAVs) with an on-board camera are tasked to reconstruct the boundary of an oil spill or of a

forest fire.

3.1 Approximation theory for convex bodies

In this section we review some known useful results from the literature on approximation of strictly convex bodies; e.g., see the extensive survey [13]. In the standard literature on convex bodies approximations, the symmetric difference δ^S between two compact, and strictly convex bodies $C, B \in \mathbb{R}^d$ is defined by $\delta^S(C, B) = \mu(C \cup B) - \mu(C \cap B)$, where μ is the Lebesgue measure on \mathbb{R}^d . If Q is the body to be approximated by an inscribed n -vertices polygon P_n , then $\delta^S(Q, P_n) = \mu(Q) - \mu(P_n)$. Let ∂Q be the boundary of Q , ℓ be the arc length along ∂Q , and θ be the angular position in a polar variable parameterization of ∂Q . Let ρ and $\kappa = \rho^{-1}$ be the curvature radius and curvature of the boundary, respectively. For n sufficiently large, McLure and Vitale [29] show that

$$\delta^S(Q, P_n^*) \approx \frac{1}{12n^2} \left(\int_0^{2\pi} \rho(\theta)^{2/3} d\theta \right)^3 = \frac{1}{12n^2} \left(\int_{\partial Q} \kappa(\ell)^{1/3} d\ell \right)^3,$$

where P_n^* is the best approximating polygon with n vertices inscribed in Q . To construct the best approximating polygon P_n^* for a strictly convex body McLure and Vitale in [29] suggest the *method of empirical distributions*. According to this method, the positions $\theta_i, i \in \{1, \dots, n\}$, of the vertices along ∂Q have the property that $D^S(i) = \int_{\theta_i}^{\theta_{i+1}} \rho(\theta)^{2/3} d\theta$ has the same value for every consecutive pair of vertices $(i, i+1)$. (Here and in what follows, we adopt the convention that $n+1 = 1$.) Interpolating polygons computed according to the method of empirical distributions converge to P_n^* as $n \rightarrow +\infty$.

For smooth nonconvex bodies with a finite number of inflection points, the

method of empirical distributions will also yield a nearly optimal distribution as $n \rightarrow +\infty$ because of the local convexity of the body away from inflection points. We show how to do this in what follows. Since the curvature radius may be unbounded at some point of a nonconvex boundary, the integral $D^S(i)$ may be unbounded for some i . We avoid this problem by considering the following general notion of distance along a boundary. For $\lambda \in [0, 1]$, we define the pseudo-distance D_λ between vertices $(i, i + 1)$ by:

$$D_\lambda(i) = \lambda \int_{\ell_i}^{\ell_{i+1}} \kappa(\ell)^{1/3} d\ell + (1 - \lambda)(\ell_{i+1} - \ell_i).$$

This definition is inspired by the fact that, for convex bodies, we have $\int_0^{2\pi} \rho(\theta)^\alpha = \int_{\partial Q} \kappa(\ell)^{1-\alpha} d\ell$ for $\alpha > 0$, see [13]. Introducing the convex combination with arc length, we guarantee that $D_\lambda(i)$ is nonzero whenever the vertices i and $i + 1$ do not coincide. Note that changing the value of the parameter λ has less noticeable impact in arcs of the boundary with high curvature and more in the arcs with low curvature. In what follows we develop a version of the method of empirical distributions in which consecutive vertices are uniformly distributed according to the pseudo-distance D_λ .

3.2 Boundary estimation and agent pursuit algorithm

In this section we propose and analyze an algorithm that leads a group of n_a agents to compute and constantly update an estimate of a slowly moving boundary. The estimate is computed in the form of an interpolating polygon; the algorithm aims to place the interpolation points so that they are uniformly

distributed according to the pseudo-distance D_λ introduced in the previous section. As discussed in the Introduction, we assume that (i) at initial time the agents have an estimate of the boundary, (ii) each agent can locally estimate the tangent and curvature of the boundary, and (iii) the agents are able to exchange information according to a ring-topology communication service.

We let $\{P_i\}_{i \in \{1, \dots, n_a\}}$ be the positions of the mobile agents and we let $\{p_\alpha\}_{\alpha \in \{1, \dots, n_{ip}\}}$ be the vertices of the interpolating polygon; in a practical implementation, we assume that each agent maintains a copy of these virtual positions. Relying upon the initial estimate of the boundary, we make the following additional assumptions: at time $t = 0$, the agents have reached a point of ∂Q and the interpolation points are distributed (possibly nonuniformly) on the estimated boundary. We assume that both the interpolation points and the agents are ordered counterclockwise, and that the agents move counterclockwise along the boundary with speed v_i , see Figure 3.1.

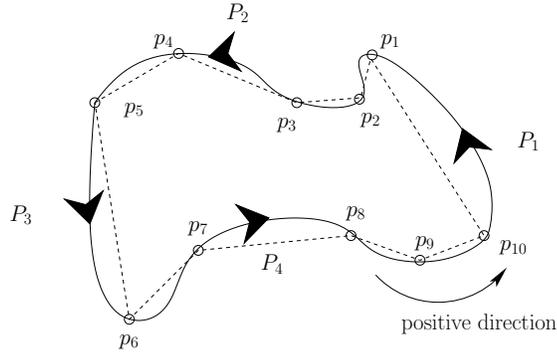


Figure 3.1. In the figure the solid line is the boundary ∂Q , the triangles are the agents, the circles are the interpolation points, and the dotted line is the approximating polygon defined by the interpolation points.

The agents have two objectives: (i) update the interpolation points such

that they are uniformly distributed along ∂Q according to the estimated pseudo-distance \widehat{D}_λ , (ii) move along the boundary equally distributed according to arc length distance. To achieve these two objectives we propose the novel ESTIMATE UPDATE AND PURSUIT ALGORITHM that can be summarized as follows.

Every agent moves counterclockwise along the time-varying ∂Q and collects estimates of the curve ∂Q and of its tangent and curvature. Using these estimates, the agent completes the following four actions: First, each agent updates the positions of the interpolation points so that they take value in ∂Q . In other words, as sufficient information is available, each interpolation point p_α , $\alpha \in \{1, \dots, n_{ip}\}$, is projected onto the measured boundary. Second, after an interpolation point p_α has been projected, the agent collects sufficient information so that it can locally optimize its position along the estimate of ∂Q . Third, every agent estimates the arc length distance between itself and its immediate clockwise and counterclockwise neighbors and uses this information to speed up or slow down. Fourth and last, the updated interpolation point p_α is transmitted to appropriate neighboring agents.

The first two steps have the combined effect of updating the local estimates of the boundary. The third step has the effect of distributing the agents uniformly along the boundary. The fourth step has the effect of maintaining correct distributed information about the boundary estimate.

3.2.1 Algorithm description

In this section we present the ESTIMATE UPDATE AND PURSUIT ALGORITHM in some detail and we analyze its stability. We begin by introducing some basic geometric notions about curves and making some smoothness assumptions. In what follows, we let $\|v\|$ be the Euclidean norm of $v \in \mathbb{R}^n$, $\overline{\mathbb{R}}_+$ be the set of

nonnegative real numbers, and \mathbb{N}_0 be the set of nonnegative integers. Let ∂Q be the boundary of a simply connected, and possibly nonconvex set Q in \mathbb{R}^2 . Let $\gamma: \bar{\mathbb{R}}_+ \times [0, 1] \rightarrow \mathbb{R}^2$ be a parametric representation of the time-varying boundary so that, at fixed $t \in \bar{\mathbb{R}}_+$ and for all $s \in [0, 1]$, $\gamma(t, s)$ describes the boundary $\partial Q(t)$. We assume that $\frac{\partial \gamma(t, s)}{\partial s} = \gamma'(t, s) \neq 0$ for all $s \in [0, 1]$ and for all t , that $\gamma(t, 0) = \gamma(t, 1)$, and that s increases as we traverse the curve in the counterclockwise direction. We also assume that $\gamma(t, s)$ is smooth with respect to s and t and that the length of the boundary ∂Q is upper and lower bounded uniformly in t . The curvature $\kappa: [0, 1] \rightarrow \bar{\mathbb{R}}_+$ of the curve γ is defined by $\kappa(s) = \frac{\|\gamma'(s) \times \gamma''(s)\|}{\|\gamma'(s)\|^3}$.

Now, we can begin our detailed description of our algorithm; we begin with the data structure. Each agent i maintains the following the following variables in its memory.

Variable #1: a counter NOW taking values in $\{1, \dots, n_{\text{ip}}\}$, when necessary we will use NOW^i to indicate the value of the counter NOW for agent i ;

Variable #2: a buffer BUFFERARC containing a collection of triplets $\{o_j, \hat{\gamma}'(o_j), \hat{\kappa}(o_j)\}$, where o_j is a point on ∂Q , $\hat{\gamma}'(o_j)$ and $\hat{\kappa}(o_j)$ are tangent vector and curvature at the point o_j , respectively, and j takes value in an index set $\{1, \dots, n_o\}$. It is also convenient to let $\mathcal{O} = \{o_j\}_{j \in \{1, \dots, n_o\}}$;

Variable #3: a boundary estimate given by interpolation points $p_1, \dots, p_{\text{ip}}$, tangent vectors at interpolation points $\gamma'_1, \dots, \gamma'_{\text{ip}}$, and pairwise pseudo-distance between interpolation points $\hat{D}_\lambda(p_\alpha, p_{\alpha+1})$, $\alpha \in \{1, \dots, n_{\text{ip}}\}$.

These variables are initialized as follows: NOW is set equal to the index of the

interpolation point that is immediately counterclockwise from $P_i(0)$, BUFFERARC is empty, and the boundary estimate is given by assumption.

Remark 3.1 (Interpretation) *The positions \mathcal{O} are points that an individual agent has recently visited while moving along ∂Q and are an arbitrarily accurate discretization of a portion of ∂Q ; these points reside in the memory of every individual agent. On the contrary, the interpolation points p_1, \dots, p_{ip} are a coarser discretization of a portion of ∂Q and are communicated among agents. The idea is that the agent moves and gathers sufficient information to update the interpolation point p_{NOW} with the set of observations in BUFFERARC, that is, to project p_{NOW} onto the discretized representation BUFFERARC of $\partial Q(t)$. •*

Let us illustrate the meaning of the variables in Figure 3.2. The curve of points represents the approximation BUFFERARC of ∂Q as seen by agent i , while the solid line represents ∂Q as known through the interpolation points p_1, \dots, p_{ip} and the tangent vectors $\gamma'_1, \dots, \gamma'_{ip}$ before any update takes place. The agent is represented by a triangle. The white circles are the interpolation points before the update, and the black circles represent the interpolation points after the update; the white arrows denote the projection of the interpolation points onto the recently measured boundary and the black arrow denotes the locally optimal repositioning of the interpolation points.

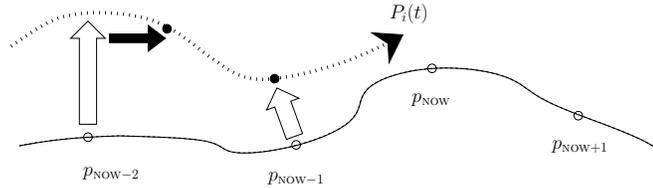


Figure 3.2. Mobile agent moving along boundary, projecting (white arrow) and locally updating (black arrow) interpolation points.

In what follows, we need to provide rules to perform the various data management tasks:

Rule #1: how to maintain the data in BUFFERARC, i.e., how long should the buffer be;

Rule #2: when and how to project onto ∂Q the next outstanding interpolation point p_{NOW} ;

Rule #3: when and how to locally optimize the updated interpolation point p_{NOW-1} ; and

Rule #4: when and what to communication and to whom.

Rule #1: If agent i is in the process of projecting interpolation point p_{NOW} , then BUFFERARC must contain information about ∂Q starting from interpolation point p_{NOW-2} up to the agent position.

Rule #2: In most cases, the projection takes place when the agent crosses the line ℓ_{NOW} that passes through p_{NOW}^- and is perpendicular to $\widehat{\gamma}'(p_{NOW}^-)$. To be specific, p_{NOW}^- denotes the interpolation point about to be updated, and $\widehat{\gamma}'(p_{NOW}^-)$ denotes the corresponding tangent vector. We can therefore define p_{NOW}^+ to be

the point where the mobile agent trajectory $P_i(t)$ crosses ℓ_{NOW} , and $\widehat{\gamma}'(p_{\text{NOW}}^+)$ to be the tangent to ∂Q at p_{NOW}^+ . This is indeed the correct definition if the agent does cross this ℓ_{NOW} . This projection operation is illustrated in Figure 3.3.

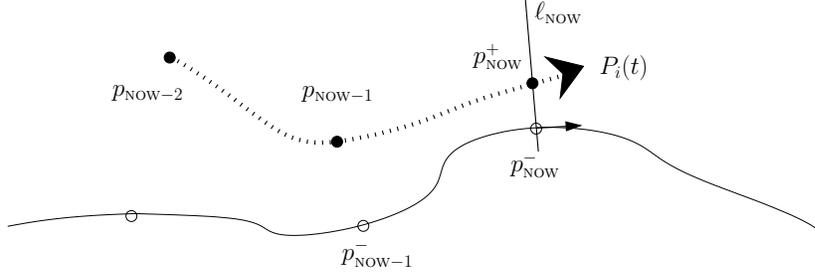


Figure 3.3. Mobile agent projecting interpolation point onto the observed boundary

We therefore amend the algorithm to act as follows. If sufficient time has elapsed without the agent crossing ℓ_{NOW} , e.g., if no crossing has happened at time t such that $\widehat{D}_\lambda(p_{\text{NOW}-1}, P(t)) = 2\widehat{D}_\lambda(p_{\text{NOW}-1}^-, p_{\text{NOW}}^-)$, then p_{NOW}^+ is set equal to the point on \mathcal{O} that is closest to p_{NOW}^- . The corresponding definition is also employed for $\widehat{\gamma}'(p_{\text{NOW}}^+)$. In both cases, this projection is well defined and has the following properties. If ∂Q is time-invariant, then $p_{\text{NOW}}^- = p_{\text{NOW}}^+$, if ∂Q is slowly time-varying, then p_{NOW}^+ is close to the orthogonal projection of p_{NOW}^- onto ∂Q .

Rule #3: The local optimization of $p_{\text{NOW}-1}$ takes place immediately after the update of p_{NOW} . Using the data in BUFFERARC, the agent computes the Voronoi cell inside \mathcal{O} of the interpolation point $p_{\text{NOW}-1}$ and moves $p_{\text{NOW}-1}$ to the center of this cell. This operation is illustrated in Figure 3.4.

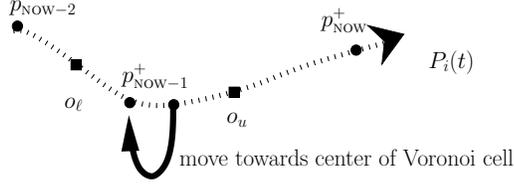


Figure 3.4. Mobile agent locally optimizing interpolation point $p_{\text{NOW}-1}$ along the observed boundary, after projecting p_{NOW}

To describe this local optimization accurately, let us introduce some notation. The Voronoi cell $\{o_\ell, \dots, o_u\} \subset \mathcal{O}$ of the interpolation point $p_{\text{NOW}-1}$ is defined implicitly by

$$\begin{aligned}\widehat{D}_\lambda(p_{\text{NOW}-2}, o_\ell) &= \widehat{D}_\lambda(o_\ell, p_{\text{NOW}-1}) = \frac{\widehat{D}_\lambda(p_{\text{NOW}-2}, p_{\text{NOW}-1})}{2}, \\ \widehat{D}_\lambda(p_{\text{NOW}-1}, o_u) &= \widehat{D}_\lambda(o_u, p_{\text{NOW}}^+) = \frac{\widehat{D}_\lambda(p_{\text{NOW}-1}, p_{\text{NOW}}^+)}{2}.\end{aligned}$$

In other words, the point o_ℓ is the midpoint between $p_{\text{NOW}-2}$ and $p_{\text{NOW}-1}$, while o_u is the midpoint between $p_{\text{NOW}-1}$ and p_{NOW}^+ after the latter was projected on ∂Q .

We now implicitly define the center $o_k \in \mathcal{O}$ of the Voronoi cell by

$$\widehat{D}_\lambda(o_\ell, o_k) = \widehat{D}_\lambda(o_k, o_u) = \frac{\widehat{D}_\lambda(p_{\text{NOW}-2}, p_{\text{NOW}-1}) + \widehat{D}_\lambda(p_{\text{NOW}-1}, p_{\text{NOW}}^+)}{4}. \quad (3.1)$$

Thus, the new position of $p_{\text{NOW}-1}$ is $p_{\text{NOW}-1}^+ = o_k$.

As a consequence $\widehat{D}_\lambda(p_{\text{NOW}-2}, p_{\text{NOW}-1})$ and $\widehat{D}_\lambda(p_{\text{NOW}-1}, p_{\text{NOW}})$ have changed, but we can easily calculate their new values:

$$\begin{aligned}\widehat{D}_\lambda(p_{\text{NOW}-2}, p_{\text{NOW}-1}^+) &= \widehat{D}_\lambda(p_{\text{NOW}-2}, o_\ell) + \widehat{D}_\lambda(o_\ell, p_{\text{NOW}-1}^+) \\ &= \frac{\widehat{D}_\lambda(p_{\text{NOW}-2}, p_{\text{NOW}-1})}{2} + \frac{\widehat{D}_\lambda(p_{\text{NOW}-2}, p_{\text{NOW}-1}) + \widehat{D}_\lambda(p_{\text{NOW}-1}, p_{\text{NOW}}^+)}{4},\end{aligned}$$

similarly, the new value for $\widehat{D}_\lambda(p_{\text{NOW}-1}^+, p_{\text{NOW}})$ can be calculated.

Rule #4: Transmission rule: after locally optimizing the position of the interpolation point $p_{\text{NOW}-1}$ and updating the corresponding data $\widehat{\gamma}(p_{\text{NOW}-1})$ and $\widehat{D}_\lambda(p_{\text{NOW}-2}, p_{\text{NOW}-1})$, agent i transmits this information to agent $i - 1$. We assume the transmission is reliable. After this local optimization is performed, the counter NOW is updated to NOW + 1 and Rule #1 is applied again, i.e., the buffer BUFFERARC is updated by dropping all observations o_j between $p_{\text{NOW}-2}$ and $p_{\text{NOW}-1}$.

Remark 3.2 (Synchronization assumption) *We assume that when agent i is relocating and transmitting information about $p_{\text{NOW}-1}$ agent $(i - 1)$ has not yet projected NOW - 2. If this assumption does not hold, i.e., if agent $i - 1$ is ready to apply Rule #2 before agent i has applied Rule #4, then agent $i - 1$ will have to keep collecting data in its buffer BUFFERARC until agent i transmits the new position of $p_{\text{NOW}-1}$.* •

Remark 3.3 (Extensions) *In the interest of simplicity, we have omitted two possible generalization that might be useful in practice. First, each agent does not need to know all interpolation points; it would suffice for it to know only the interpolation points located ahead of its position and before the position of the preceding agent. Second, each agent could locally optimize not only a single interpolation point, but it could store a longer buffer and locally optimize arrays of interpolation points.* •

This completes our description of the estimate update algorithm and we now focus on the pursuit objective. To uniformly distribute the agents along the boundary ∂Q according to arc length, we use the following update law for their

velocities:

$$v_i(t) = v_0 + k_{\text{prop}}(\widehat{L}(P_i, P_{i+1}) - \widehat{L}(P_{i-1}, P_i)),$$

with $k_{\text{prop}}, v_0 > 0$ and $\widehat{L}(P_n, P_m) = \sum_{\alpha=\text{NOW}^n+1}^{\text{NOW}^m} (\|p_{\alpha-1} - p_{\alpha}\|)$, for all $n, m \in \{1, \dots, n_a\}$. Here, recall that $p_{\text{NOW}^n}, p_{\text{NOW}^n+1}, \dots, p_{\text{NOW}^m}$ are the interpolation points separating agent n and agent m , with $n < m$, and therefore \widehat{L} is the estimated arc length of the portion of ∂Q that has to be traversed to go from the agent n to the agent m . The agents have only local information of ∂Q but still they have to estimate the distance, along ∂Q , from their clockwise and counterclockwise neighbors in order to calculate their speed. The estimate $\widehat{L}(P_n, P_m)$ is obtained from the approximating polygon formed by the interpolation points. In practice, any agent will speed up if it is closer to the agent behind it, and slow down if closer to the agent in front of it. With a saturation-like function: $\text{sat}(v_i(t)) = \max\{v_{\min}, \min\{v_i(t), v_{\max}\}\}$, we additionally impose that $0 < v_{\min} \leq v_i(t) \leq v_{\max}$ for all t .

Remark 3.4 (Partial knowledge) *The pursuit objective of the proposed algorithm requires more knowledge than the boundary estimation objective. In fact, to calculate $v_i(t)$, agent i needs to know the position not only of all the interpolation points between itself and P_{i+1} , but also of the ones between itself and P_{i-1} . Therefore, in addition to the data transmitted according to Rule #4, we require that agent i transmits p_{NOW}^+ and $p_{\text{NOW}+1}$ to $i-1$ and the counter $\text{NOW}+1$ to $i+1$.*

•

Now we summarize the discussion in this section with a pseudo-code description of the algorithm in Table 3.1.

3.2.2 Algorithm analysis

Some steps of the algorithm are affected by noise and error: (i) $\widehat{\gamma}'$ and $\widehat{\kappa}$ are only estimate of the true values, (ii) \widehat{L} is an approximation of L , (iii) the set \mathcal{O} is a discretization of the subset of ∂Q that agent i is visiting, therefore, the center of the Voronoi cell of the interpolation point p_{NOW^i-1} might not be calculated exactly. Let $\widehat{\mathbf{D}}(t)$ and $\mathbf{L}(t)$ be the column vectors:

$$\widehat{\mathbf{D}}(t) = [\widehat{D}_\lambda(p_1(t), p_2(t)), \dots, \widehat{D}_\lambda(p_{n_{\text{ip}}-1}(t), p_{n_{\text{ip}}}(t)), \widehat{D}_\lambda(p_{n_{\text{ip}}}(t), p_1(t))]^T,$$

$$\mathbf{L}(t) = [L(P_1(t), P_2(t)), \dots, L(P_{n_a-1}(t), P_{n_a}(t)), L(P_{n_a}(t), P_1(t))]^T.$$

Consider now the disagreement vectors $\mathbf{d}(t)$ and $\delta\mathbf{L}(t)$ defined by:

$$\mathbf{d}(t) = \widehat{\mathbf{D}}_\lambda(t) - \frac{\mathbf{1}^T \widehat{\mathbf{D}}_\lambda(t)}{n_{\text{ip}}} \mathbf{1}, \quad (3.2)$$

$$\delta\mathbf{L}(t) = \mathbf{L}(t) - \frac{\mathbf{1}^T \mathbf{L}(t)}{n_a} \mathbf{1}. \quad (3.3)$$

Note that they are orthogonal to the vector $\mathbf{1}$, i.e., the column vector in \mathbb{R}^n with all entries equal to 1.

We now establish that the dynamics of \mathbf{d} and $\delta\mathbf{L}$ are input-to-state stable (ISS) where the inputs are the errors and noises above discussed. Because of the ISS property we can conclude that, if the errors are bounded, then the states \mathbf{d} and $\delta\mathbf{L}$ are within a bounded distance from the origin.

Theorem 3.1 (ISS of the dynamics of the interp. points distances) *If the boundary is slowly time-varying and if $t \mapsto L(\partial Q(t))$ is upper and lower bounded uniformly in t , then, under the ESTIMATE UPDATE AND PURSUIT ALGORITHM, there exists a sequence of instants τ_k , for $k \in \mathbb{N}_0$, and a sequence of ergodic and doubly stochastic matrices $\mathcal{A}(k)$, for $k \in \mathbb{N}_0$, such that*

$$\mathbf{d}(\tau_{k+1}) = \mathcal{A}(k)\mathbf{d}(\tau_k) + \delta\mathbf{u}(\tau_k), \quad k \in \mathbb{N}_0,$$

where $\delta \mathbf{u}(\tau_k) = \mathbf{u}(\tau_k) - \frac{\mathbf{1}^T \mathbf{u}(\tau_k)}{n_{ip}} \mathbf{1}$, and $\mathbf{u}(\tau_k)$ is a bounded vector taking into account the effect of the estimation errors and of the boundary deformation during the interval $[\tau_k, \tau_{k+1}]$. Furthermore, the dynamics of \mathbf{d} are input-to-state stable with input $\delta \mathbf{u}$.

Proof. In what follows we identify $\tau_0 \equiv 0$ and $\tau_k \equiv k$. Let us suppose that $\partial Q(t)$ is time-invariant, and that \mathcal{O} is a continuous (and not discrete) representation of ∂Q , i.e., $\mathbf{u}(k) = \mathbf{0}$ for all $k \in \mathbb{N}_0$. Then, because of Rule #2, p_{NOW} is projected onto itself and, because of Rule #3, $p_{\text{NOW}-1}$ is moved exactly to the center of its Voronoi cell. Suppose that an agent has passed by the point p_{NOW} , and then it can optimally place $p_{\text{NOW}-1}$. As a consequence, the pseudo-distances $\widehat{D}_\lambda(p_{\text{NOW}-2}, p_{\text{NOW}-1})$ and $\widehat{D}_\lambda(p_{\text{NOW}-1}, p_{\text{NOW}})$ will take new values that can be expressed as follows, (recall Figure 3.4):

$$\begin{aligned} \widehat{D}_\lambda(p_{\text{NOW}-2}, p_{\text{NOW}-1})^+ &= \frac{3}{4} \widehat{D}_\lambda(p_{\text{NOW}-2}, p_{\text{NOW}-1}) + \frac{1}{4} \widehat{D}_\lambda(p_{\text{NOW}-1}, p_{\text{NOW}}), \\ \widehat{D}_\lambda(p_{\text{NOW}-1}, p_{\text{NOW}})^+ &= \frac{1}{4} \widehat{D}_\lambda(p_{\text{NOW}-2}, p_{\text{NOW}-1}) + \frac{3}{4} \widehat{D}_\lambda(p_{\text{NOW}-1}, p_{\text{NOW}}), \end{aligned}$$

where the superscript $^+$ indicates the new values of the pseudo-distances after $p_{\text{NOW}-1}$ has been optimally placed. For $\alpha \in \{1, \dots, n_{ip}\}$, define the doubly stochastic matrix $A_\alpha \in \mathbb{R}^{n_{ip} \times n_{ip}}$ by

$$(A_\alpha)_{jh} = \begin{cases} 3/4, & \text{if } j = h = \alpha, \text{ or } j = h = \alpha - 1, \\ 1/4, & \text{if } j = \alpha - 1 \text{ and } h = \alpha, \text{ or if } j = \alpha, \text{ and } h = \alpha - 1, \\ \delta_{jh}, & \text{otherwise.} \end{cases}$$

Therefore, A_α , for $\alpha \in \{1, \dots, n_{ip}\}$, are the matrices determining the dynamic system $\widehat{\mathbf{D}}_\lambda(t_2) = A_\alpha \widehat{\mathbf{D}}_\lambda(t_1)$, where $t_2 > t_1$ is the time when the interpolation point α is moved by an agent to its new Voronoi center and where we are assuming that between t_1 and t_2 no other interpolation point has been moved. If at the

same instant more interpolation points are relocated, then the matrix describing the dynamics is the product of all the A_α that correspond to the relocated interpolation points. The order of the matrix multiplication is irrelevant as one can show that these matrices commute. Let us now derive the dynamics of $\widehat{\mathbf{D}}_\lambda$ when ∂Q is slowly time-varying, while \mathcal{O} is still a continuous representation of ∂Q . By assumption $\gamma(t, s)$ is smooth in both its arguments and, as argued above, the projection of the interpolation points is well defined and unique.

Let t_α^{k+1} be the $k + 1$ -th time that p_α is optimally placed by an agent. Before optimally placing p_α , the agent will project $p_{\alpha+1}$. It can be proved that right before placing p_α , because the boundary has changed, the pseudo-distance $\widehat{D}_\lambda(p_\alpha, p_{\alpha+1}, t_\alpha^{k+1})$ will differ from $\widehat{D}_\lambda(p_\alpha, p_{\alpha+1}, t_{\alpha+1}^k)^+$ by some noise $g(t_\alpha^{k+1} - t_{\alpha+1}^k)$ which is a continuous function of $t_\alpha^{k+1} - t_{\alpha+1}^k$ and $g(0) = 0$. With $\widehat{D}_\lambda(p_\alpha, p_{\alpha+1}, t_{\alpha+1}^k)^+$ we denote the pseudo-distance between p_α and $p_{\alpha+1}$ right after $p_{\alpha+1}$ has been optimally placed for the k th time. Therefore, the system is evolving according to a dynamical system of the form:

$$\widehat{\mathbf{D}}_\lambda(p_\alpha, p_{\alpha+1}, t_\alpha^{k+1})^+ = A_\alpha \left(\widehat{\mathbf{D}}_\lambda(p_\alpha, p_{\alpha+1}, t_{\alpha+1}^k) + \mathbf{e}_{\alpha+1} g(t_\alpha^{k+1} - t_{\alpha+1}^k) \right), \quad (3.4)$$

where \mathbf{e}_α is the column vector with null entries but the α -th component that is equal to 1, and the subscript $^+$ indicates that the interpolation point p_α has just been optimally placed. Let $\Delta T = \sup_{t \in \mathbb{R}_+} \frac{L(\partial Q(t))}{v_{\min}}$. Note that $\Delta T < +\infty$ since by assumption the length of the boundary $\partial Q(t)$ is uniformly upperbounded. This means that at most after ΔT any interpolation point is updated at least once. Any time that an agent updates any interpolation point p_α the vector $\widehat{\mathbf{D}}_\lambda$ evolves according to (3.4), where $t_\alpha^{k+1} - t_{\alpha+1}^k$ is upperbounded by ΔT . Because ΔT is finite, there exists a sequence of instants τ_k , with $k \in \mathbb{N}_0$, such that across the interval $[\tau_k, \tau_{k+1}]$ every interpolation point has been updated at least once by an

agent, and:

$$\widehat{\mathbf{D}}_\lambda(k+1) = \mathcal{A}(k)\widehat{\mathbf{D}}_\lambda(k) + \mathbf{u}(k), \quad k \in \mathbb{N}_0. \quad (3.5)$$

The matrix $\mathcal{A}(k)$ is the product of a finite number $M(k)$ of matrices A_α , i.e., $\mathcal{A}(k) = \prod_{\beta=1}^{M(k)} A_{\alpha_\beta}$, $\alpha_\beta \in \{1, \dots, n_{\text{ip}}\}$. The value of the index α_β depends on the order in which the interpolation points are updated. It is easy to see that $n_{\text{ip}} \leq M(k) \leq n_a n_{\text{ip}}$. Note that $\mathcal{A}(k)$ is doubly stochastic because it is the product of doubly stochastic matrices. Since across the interval $[\tau_k, \tau_{k+1}]$ every interpolation point has been updated at least once by an agent, the graph associated with $\mathcal{A}(k)$ is connected and therefore $\mathcal{A}(k)$ is ergodic (see [12]). Furthermore, $\sup(\tau_{k+1} - \tau_k) \leq \Delta T < +\infty$, and (by [30]) we claim that, if $\mathbf{u}(k) \equiv 0$, then $\widehat{\mathbf{D}}_\lambda(k)$ converges exponentially fast to $\frac{\mathbf{1}^T \widehat{\mathbf{D}}_\lambda(k)}{n_{\text{ip}}} \mathbf{1}$ (see Remark 4.1 in Section 4.3). Consider now the disagreement vector $\mathbf{d}(k)$ defined in (3.2). Recalling (3.5), and that $\mathcal{A}(k)$ is doubly stochastic, we can derive the update law of the disagreement $\mathbf{d}(k)$:

$$\mathbf{d}(k+1) = \mathcal{A}(k)\mathbf{d}(k) + \delta\mathbf{u}(k), \quad k \in \mathbb{N}_0, \quad (3.6)$$

where $\delta\mathbf{u}(k) = \mathbf{u}(k) - \frac{\mathbf{1}^T \mathbf{u}(k)}{n_{\text{ip}}} \mathbf{1}$. Given the properties of the matrix $\mathcal{A}(k)$, the origin of the unforced system (3.6) is exponentially stable. Since the input $\delta\mathbf{u}$ enters linearly, we conclude that the system is input-to-state stable ([23]). This implies that $\widehat{\mathbf{D}}_\lambda$ will asymptotically reach a ball centered at $\frac{\mathbf{1}^T \widehat{\mathbf{D}}_\lambda(k)}{n_{\text{ip}}} \mathbf{1}$ (the equilibrium of the unforced system) and with radius that is a \mathcal{K} -function of the input, see [23] and 4.1. This also holds if we now relax the assumption that no errors affect the calculation of the Voronoi centers, because this error enters linearly in the system (3.6). ■

Theorem 3.2 (ISS of the dynamics of the inter-agent distances) *If the boundary is slowly time-varying and if $t \mapsto L(\partial Q(t))$ is upper and lower bounded*

uniformly in t , then, under the ESTIMATE UPDATE AND PURSUIT ALGORITHM,

$$\delta\dot{\mathbf{L}}(t) = k_{\text{prop}}A(c_1(t), \dots, c_{n_a}(t))\left(\delta\mathbf{L}(t) + \delta\mathbf{w}_1(t)\right) + \delta\mathbf{w}_2(t),$$

where $\delta\mathbf{w}_1(t) = \mathbf{w}_1(t) - \frac{\mathbf{1}^T\mathbf{w}_1(t)}{n_a}\mathbf{1}$, $\delta\mathbf{w}_2(t) = \mathbf{w}_2(t) - \frac{\mathbf{1}^T\mathbf{w}_2(t)}{n_a}\mathbf{1}$, and

$$(A(c_1(t), \dots, c_{n_a}(t)))_{jh} = \begin{cases} -c_i(t) - c_{i+1}(t) & \text{if } j = h = i, \\ c_{i+1}(t) & \text{if } j = i \text{ and } h = i + 1, \\ c_i(t) & \text{if } j = i \text{ and } h = i - 1, \\ 0 & \text{otherwise.} \end{cases}$$

with $c_i(t) \in [\beta(t), 1]$ for all $i \in \{1, \dots, n_a\}$, and $\beta(t) = \frac{\min\{v_{\max} - v_0, v_0 - v_{\min}\}}{k_{\text{prop}}L(\partial Q(t))}$. The variable $\mathbf{w}_2(t) \in \mathbb{R}^{n_a \times 1}$ expresses the change in the arc length distance between any two consecutive agents due to the deformation of $\partial Q(t)$, while $\mathbf{w}_1(t) = \mathbf{L}(t) - \widehat{\mathbf{L}}(t) \in \mathbb{R}^{n_a \times 1}$ is the error due to the fact that the agents do not know exactly \mathbf{L} , the arc length distance between them and their neighbors, but only an estimate through the interpolation points $\widehat{\mathbf{L}}$. Furthermore, the dynamics of $\delta\mathbf{L}$ is input-to-state stable with $t \mapsto \delta\mathbf{w}_1(t)$ and $t \mapsto \delta\mathbf{w}_2(t)$ as inputs.

Proof. Let us suppose that the $\partial Q(t)$ is time-invariant and that the agents can actually compute without error the arc length distance between them and their clockwise and counterclockwise neighbors, i.e., $\mathbf{w}_1(t) = \mathbf{w}_2(t) = \mathbf{0}$ for all $t \geq 0$. The dynamics for $\mathbf{L}(t)$ can be written as $\dot{L}(P_i(t), P_{i+1}(t)) = v_{i+1} - v_i$, where $v_{i+1} = \text{sat}(v_0 + k_{\text{prop}}(L(P_{i+1}, P_{i+2}) - L(P_i, P_{i+1})))$ and $v_i = \text{sat}(v_0 + k_{\text{prop}}(L(P_i, P_{i+1}) - L(P_{i-1}, P_i)))$. Therefore, if the saturation on the speeds is not active, we have:

$$\dot{L}(P_i(t), P_{i+1}(t)) = k_{\text{prop}}\left(L(P_{i+1}, P_{i+2}) - 2L(P_i, P_{i+1}) + L(P_{i-1}, P_i)\right),$$

which in matrix form becomes:

$$\dot{\mathbf{L}}(t) = k_{\text{prop}} \begin{bmatrix} -2 & 1 & 0 & \dots & 1 \\ 1 & -2 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 1 & -2 & 1 \\ 1 & 0 & \dots & 1 & -2 \end{bmatrix} \mathbf{L}(t) = k_{\text{prop}} A_L \mathbf{L}(t).$$

If, for agent i , the saturation is active, then we have that

$v_i = v_0 + k'_i (L(P_i, P_{i+1}) - L(P_{i-1}, P_i))$, where $k'_i = k_{\text{prop}} c_i$, $c_i \leq 1$. In other words, we can think of the saturation function as a change in the gain in the control law. If $v_i = v_{\text{max}}$, then

$$\begin{aligned} k'_i &= \frac{v_{\text{max}} - v_0}{(L(P_i, P_{i+1}) - L(P_{i-1}, P_i))} \geq \frac{v_{\text{max}} - v_0}{L(\partial Q(t))} \\ &\implies c_i = \frac{k'_i}{k_{\text{prop}}} \geq \frac{1}{k_{\text{prop}}} \frac{v_{\text{max}} - v_0}{L(\partial Q(t))} > 0. \end{aligned}$$

If $v_i = v_{\text{min}}$, then

$$\begin{aligned} k'_i &= \frac{v_0 - v_{\text{min}}}{(L(P_i, P_{i+1}) - L(P_{i-1}, P_i))} \geq \frac{v_0 - v_{\text{min}}}{L(\partial Q(t))} \\ &\implies c_i = \frac{k'_i}{k_{\text{prop}}} \geq \frac{1}{k_{\text{prop}}} \frac{v_0 - v_{\text{min}}}{L(\partial Q(t))} > 0, \end{aligned}$$

in any case $c_i \in [\beta(t), 1]$ and $\beta(t) = \frac{\min\{v_{\text{max}} - v_0, v_0 - v_{\text{min}}\}}{k_{\text{prop}} L(\partial Q(t))}$.

Clearly then, if we introduce the saturation on the speeds v_i , the dynamics of \mathbf{L} becomes:

$$\dot{\mathbf{L}}(t) = k_{\text{prop}} A(c_1(t), \dots, c_{n_a}(t)) \mathbf{L}(t). \quad (3.7)$$

Note that $\beta(t)$ is constant because we are still considering ∂Q time-invariant. Using the properties of Metzler matrices, it can be proved (see Appendix) that the new matrices $A(c_1(t), \dots, c_{n_a}(t))$, like A_L , are negative semidefinite. The unique zero eigenvalue is associated with the eigenvector $\mathbf{1}$.

Let us consider the disagreement $\delta\mathbf{L}$ as described by (3.3), then

$$\delta\dot{\mathbf{L}}(t) = k_{\text{prop}}A(c_1(t), \dots, c_{n_a}(t))\delta\mathbf{L}(t). \quad (3.8)$$

Let $V(\delta\mathbf{L}(t)) = \delta\mathbf{L}^T(t)\delta\mathbf{L}(t)$ be the candidate Lyapunov function for the system (3.8), then we have that $\dot{V}(\delta\mathbf{L}(t)) = 2k_{\text{prop}}\delta\mathbf{L}(t)A(c_1(t), \dots, c_{n_a}(t))\delta\mathbf{L}(t) \leq 0$, where the equality holds only if the entries of $\delta\mathbf{L}(t)$ are all zero. Since c_i belong to a compact set, the matrices $A(c_1(t), \dots, c_{n_a}(t))$ belong to a compact set, and since the eigenvalues of a matrix are continuous functions of its entries (see [17]), then there exists an upperbound $-\rho < 0$ for the eigenvalues that are different from zero and as a consequence $\dot{V}(\delta\mathbf{L}(t)) \leq -\rho\|\delta\mathbf{L}(t)\|^2$. We can then conclude that for the system (3.8) the origin is exponentially stable and therefore, for the system (3.7), the equilibrium $\frac{\mathbf{1}^T\mathbf{L}(t)}{n_a}\mathbf{1}$ is exponentially stable.

Let us now assume that the boundary is slowly-varying and that instead of $L(P_i, P_{i+1})$ the agents use only the approximation $\widehat{L}(P_i, P_{i+1})$, i.e., $\mathbf{w}_1(t), \mathbf{w}_2(t) \neq \mathbf{0}$. Then, the variation in time of the vector $\mathbf{L}(t)$ is due, not only to the fact that the agents speed up and slow down, but also to the deformation of ∂Q :

$$\dot{\mathbf{L}}(t) = k_{\text{prop}}A(t)\widehat{\mathbf{L}}(t) + \mathbf{w}_2(t) = k_{\text{prop}}A(t)(\mathbf{L}(t) + \mathbf{w}_1(t)) + \mathbf{w}_2(t),$$

where $A(t) = A(c_1(t), \dots, c_{n_a}(t))$, $\mathbf{w}_1(t) = \mathbf{L} - \widehat{\mathbf{L}}$, and $\mathbf{w}_2(t)$ expresses the deformation of $\partial Q(t)$. In particular the i -th entry of $\mathbf{w}_2(t)$ is equal to

$$\frac{\partial}{\partial t} \int_{s_i}^{s_{i+1}} \|\gamma'(s, t)\| ds - k_{\text{prop}}A_i(t)(\mathbf{L}(t) + \mathbf{w}_1(t)),$$

where $A_i(t)$ is the i -th row of $A(t)$. Note that $c_i(t) \in [\beta(t), 1]$ for all $i \in \{1, \dots, n_a\}$, and that $\beta(t) = \frac{\min\{v_{\max} - v_0, v_0 - v_{\min}\}}{k_{\text{prop}}L(\partial Q(t))}$

is indeed uniformly upper bounded even when ∂Q is time-varying because we assumed that $L(\partial Q(t))$ is upper and lower bounded uniformly in t . The vector \mathbf{w}_2 is bounded because by assumption the boundary is smooth and slowly time-varying. Using the change of variables in equation (3.3), and recalling that $A(t)\mathbf{1} = \mathbf{0}$, for

all t , we have:

$$\dot{\mathbf{L}}(t) = k_{\text{prop}}A(t)\delta\mathbf{L}(t) + k_{\text{prop}}A(t)\delta\mathbf{w}_1(t) + \delta\mathbf{w}_2(t) + \frac{\mathbf{1}^T\mathbf{w}_2(t)}{n_a}\mathbf{1},$$

where $\delta\mathbf{w}_2(t) = \mathbf{w}_2(t) - \frac{\mathbf{1}^T\mathbf{w}_2(t)}{n_a}\mathbf{1}$. It is easy to see that we can write $\dot{\mathbf{L}}(t) = \delta\dot{\mathbf{L}}(t) + \frac{\mathbf{1}^T\dot{\mathbf{L}}(t)}{n_a}\mathbf{1}$ and therefore:

$$\delta\dot{\mathbf{L}}(t) + \frac{\mathbf{1}^T\dot{\mathbf{L}}(t)}{n_a}\mathbf{1} = k_{\text{prop}}A(t)\delta\mathbf{L}(t) + k_{\text{prop}}A(t)\delta\mathbf{w}_1(t) + \delta\mathbf{w}_2(t) + \frac{\mathbf{1}^T\mathbf{w}_2(t)}{n_a}\mathbf{1}.$$

Since $\mathbf{1}$ is orthogonal to $\delta\mathbf{L}(t)$, $\delta\mathbf{w}_1(t)$, and $\delta\mathbf{w}_2(t)$ we have:

$$\delta\dot{\mathbf{L}}(t) = k_{\text{prop}}A(t)\delta\mathbf{L}(t) + k_{\text{prop}}A(t)\delta\mathbf{w}_1(t) + \delta\mathbf{w}_2(t). \quad (3.9)$$

The system described by equation (3.9) is input-to-state stable with inputs $\delta\mathbf{w}_1(t)$ and $\delta\mathbf{w}_2(t)$ because (i) the origin of the unforced system is exponentially stable, and (ii) the right-hand-side of (3.9) is differentiable and uniformly globally Lipschitz in $\delta\mathbf{L}$, $\delta\mathbf{w}_1(t)$ and $\delta\mathbf{w}_2(t)$, (see [23]).

The ISS property guarantees that if \mathbf{w}_1 and \mathbf{w}_2 are bounded, then \mathbf{L} will asymptotically reach a ball centered at $\frac{\mathbf{1}^T\mathbf{L}(t)}{n_a}\mathbf{1}$ (the equilibrium of the unforced system) and with radius that is a \mathcal{K} -function, see [23] and 4.1. The larger n_{ip} is and the slower the deformation of ∂Q is, then the smaller \mathbf{w}_1 and \mathbf{w}_2 are, and the closer to $\mathbf{0}$ the disagreement $\delta\mathbf{L}$ will be asymptotically. ■

3.2.3 Simulations

In this section we present results of two different simulations obtained with the implementation of the ESTIMATE UPDATE AND PURSUIT ALGORITHM. In the first simulation the boundary ∂Q is time invariant, while in the second is time varying.

Time-invariant boundary

In this simulation we use $n_a = 3$ agents to have an approximation of the nonconvex boundary ∂Q described by:

$$\gamma(\theta) = \left(2 + \cos(10\pi\theta) + 0.5 \sin(4\pi\theta)\right) \begin{bmatrix} \cos(2\pi\theta) \\ \sin(2\pi\theta) \end{bmatrix}.$$

The outcome is shown in Figure 3.5. In order to calculate their speeds, the agents use $v_0 = 1$, and $k_{\text{prop}} = 0.05$. The saturation function for the speed has lower limit $v_{\min} = 0.5$ and upper limit $v_{\max} = 2$. The number of interpolation points is $n_{\text{ip}} = 30$, while $\lambda = \frac{10}{11}$. The simulation time is 50 seconds and the sampling time 0.01 seconds. The plots in Figure 3.5 corresponds to the positions of the interpolation points and the agents at the initial and final configuration. The interpolation points p_{NOW^i} for $i \in \{1, \dots, n_a\}$ at time $t = 0$ coincide with the positions of the agents. The other interpolation points are randomly distributed on the boundary. In the last frame one can also see the approximating polygon and how close it is to the actual boundary.

Since the pseudo-distance D_λ and the arc length L can be calculated after the simulation is completed, we use D_λ and L instead of their estimate \widehat{D}_λ and \widehat{L} to show the algorithm performance. Figure 3.6 does indeed show the convergence of the algorithm. In the first plot we can see that the consensus on the pseudo-distance $D_\lambda(p_i, p_{i+1})$, between any two consecutive interpolation points, is reached. The quantity

$$\max_{\alpha \in \{1, \dots, n_{\text{ip}}\}} D_\lambda(p_\alpha, p_{\alpha+1}) - \min_{\alpha \in \{1, \dots, n_{\text{ip}}\}} D_\lambda(p_\alpha, p_{\alpha+1})$$

does not vanish because of numerical errors in the estimate \widehat{D}_λ . The second plot shows how the agents get uniformly spaced along the boundary. The steady state

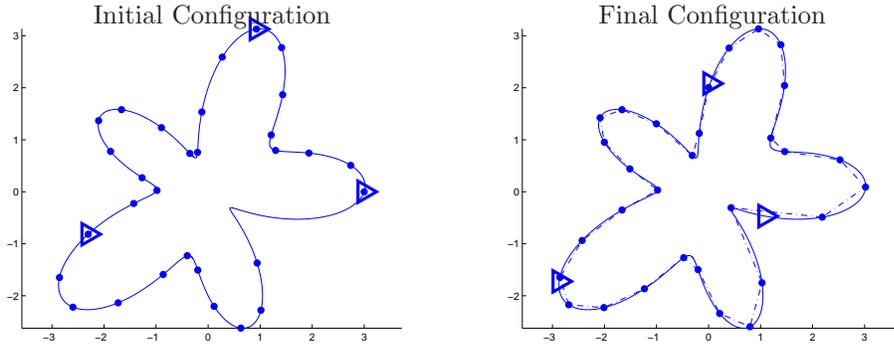


Figure 3.5. This figure shows initial and final configuration after 50 seconds simulation obtained by the implementation of the ESTIMATE UPDATE AND PURSUIT ALGORITHM with $n_a = 3$, $n_{ip} = 30$, $v_0 = 1$, $k_{prop} = 0.05$, $\lambda = \frac{10}{11}$. ∂Q is time invariant. The agents position is represented by the triangles and are initialized to be on the boundary ∂Q . In the last frame also the approximating polygon is shown.

values of the arc length distances oscillates around 8.3 which is the target value. The noise is again due to the fact that the agents only estimate the arc length using the positions of the interpolation points.

Slowly time-varying boundary

In this simulation we used $n_a = 4$ agents to have an approximation of the nonconvex boundary $\partial Q(t)$ described by:

$$\gamma(\theta, t) = \left(2 \left(1 - \frac{t}{t_f} \right) + \left(2 + \cos(10\pi\theta) + 0.5 \sin(4\pi\theta) \right) \frac{t}{t_f} \right) \begin{bmatrix} \cos(2\pi\theta) \\ \sin(2\pi\theta) \end{bmatrix},$$

with $\theta \in [0, 1)$, $t_f = 200$ seconds as shown in Figure 3.7. The values of v_0 , k_{prop} , v_{min} , v_{max} and λ are respectively: 1, 0.05, 0.5, 2, and $\frac{10}{11}$. The simulation time is 200 seconds, the sampling time 0.01 seconds. The plots in Figure 3.7

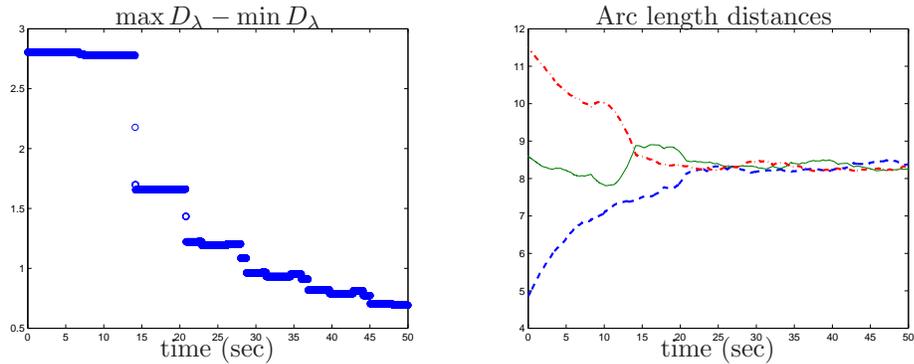


Figure 3.6. ESTIMATE UPDATE AND PURSUIT ALGORITHM This plots refers to the case of ∂Q being time-invariant. In the first plot from right it is shown the error $\max_{i \in \{1, \dots, n_{ip}\}} D_\lambda(p_i, p_{i+1}) - \min_{i \in \{1, \dots, n_{ip}\}} D_\lambda(p_i, p_{i+1})$ vs time. The second plot shows the arc length distances between the three agents.

correspond to the positions of the interpolation points and the agents at four different instants, $t = 0$, $t = 50$, $t = 100$, and $t = 200$ seconds respectively. The algorithm is initialized with the agents on the boundary. The interpolation points p_{NOW^i} at time $t = 0$ coincide with the positions of the agents. The other interpolation points are randomly distributed. In the last frame we can also see the approximating polygon and how close to the actual boundary is. From the frames in Figure 3.7 it is clear that the agents can adapt as ∂Q changes.

The pseudo-distance D_λ is well defined only if the interpolation points belong to the boundary ∂Q . Since the boundary changes with time, the interpolation points are only for some time on the boundary after an agents has projected them. So, we consider as pseudo-distance between any two consecutive interpolation points in a certain time τ the pseudo-distance between their radial projection onto $\partial Q(\tau)$. The disagreement in the placement of the interpolation points, where D_λ is redefined as just explained, is shown in the first plot of Figure 3.8.

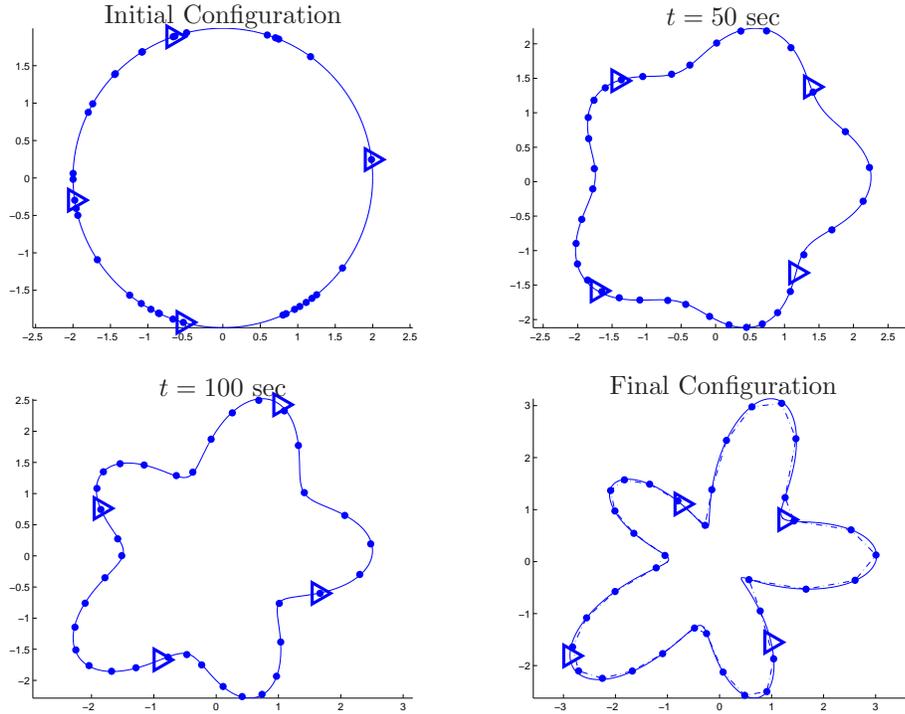


Figure 3.7. This figure shows four different instants of the 200 seconds simulation obtained by implementing the ESTIMATE UPDATE AND PURSUIT ALGORITHM with $n_a = 4$, $n_{ip} = 35$, $v_0 = 1$, $k_{prop} = 0.05$, $\lambda = \frac{10}{11}$. The boundary ∂Q is slowly time-varying in this case. The agents positions are represented by triangles and initialized to be on the boundary ∂Q . The last frame also shows the approximating polygon.

The arc length between any two consecutive agents is shown in the second plot of Figure 3.8. The four distances increase with time because $L(\partial Q)$, the total length of the boundary, increases with time. Clearly the variables n_a and n_{ip} are design variables and are results of two different trade-offs. In deciding the number of agents, the speed with which the boundary changes and the maximum speed at which the robots can move play an important role. If the boundary changes very fast and the maximum speed of the robots is fixed, then a larger network will guarantee better performances. If the boundary changes slowly, then few robots (1 or 2) might suffice. In deciding the number of interpolation points, on the other hand, the most important role is played by the complexity of the boundary measured by the number of inflection points. To have good performance, n_{ip} should increase as the number of the inflection points of the boundary increases.

3.3 Summary and open issues

In this chapter we have addressed the problem of boundary estimation and tracking by means of robotic sensors. We have presented an algorithm to position interpolation points along a time-varying boundary in such a way as to obtain an approximating polygon with some optimality features. The convergence was also established.

An important problem for future research concerns the possible and realistic occurrence of boundary splits. In other words, it would be of interest to consider problems where the region enclosed in the boundary can split into two or more separate regions. A natural extension to this work would be also be monitoring 3-dimensional regions, such as clouds of chemical pollutants.

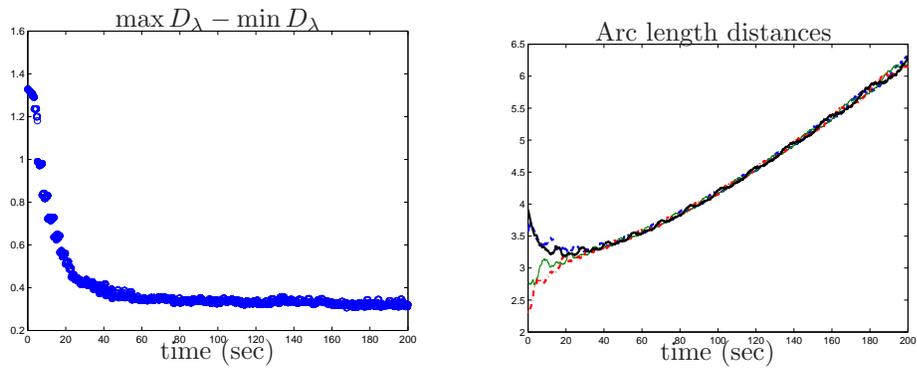


Figure 3.8. ESTIMATE UPDATE AND PURSUIT ALGORITHM. This figure refers to the case of ∂Q being slowly time-varying. In the first plot from the right we shown the error $\max_{i \in \{1, \dots, n_{ip}\}} D_\lambda(p_i, p_{i+1}) - \min_{i \in \{1, \dots, n_{ip}\}} D_\lambda(p_i, p_{i+1})$ vs time. The second plot shows the arc length distances between the four agents.

Table 3.1. ESTIMATE UPDATE AND PURSUIT ALGORITHM

Goal:	Uniformly distribute the interpolation points according to the pseudo-distance \widehat{D}_λ , and the agents according to the arc length \widehat{L} .
Data:	Location of the interpolation points, unitary tangent vector at ∂Q at those points, last value of \widehat{D}_λ between any two consecutive interpolation points, local tangent and local curvature of the boundary ∂Q .
Requires:	At $t_0 = 0$, p_i lie on ∂Q , \widehat{D}_λ between any two interpolation points is known, and $o_q = \emptyset$.

At every sensing instant, the agent at position $P_i(t) = P(t)$ performs:

- 1: update $\text{BUFFERARC}^+ := \text{BUFFERARC} \cup \{o_{n_o+1}, \widehat{\gamma}'(o_{n_o+1}), \widehat{\kappa}(o_{n_o+1}), \widehat{D}_\lambda(o_{n_o}, o_{n_o+1})\}$
- 2: **if** $o_q = \emptyset$, **then**
- 3: **if** $\overline{o_{n_o} o_{n_o+1}} \cap \ell_{\text{NOW}} \neq \emptyset$, **then**
- 4: $o_q := \operatorname{argmin}_{o_j \in \{o_{n_o}, o_{n_o+1}\}} \|\overline{o_{n_o} o_{n_o+1}} \cap \ell_{\text{NOW}} - o_j\|$
- 5: **else**
- 6: **if** $\overline{o_{n_o} o_{n_o+1}} \cap \ell_{\text{NOW}} = \emptyset$ and $\widehat{D}_\lambda(p_{\text{NOW}-1}, o_{n_o+1}) > 2\widehat{D}_\lambda(p_{\text{NOW}-1}, p_{\text{NOW}})$, **then**
- 7: $o_q := \operatorname{argmin}_{o_j \in \{p_{\text{NOW}-1}, \dots, o_{n_o+1}\} \subset O} \|o_j - p_{\text{NOW}}\|$
- 8: **if** $o_q \neq \emptyset$ and $p_{\text{NOW}^i} \neq p_{\text{NOW}^{i+1}-2}$, **then**
- 9: update the interpolation point p_{NOW} by projecting it onto ∂Q : $p_{\text{NOW}}^+ := o_q$
- 10: calculate o_k as in (3.1) and update $p_{\text{NOW}-1}$ by: $p_{\text{NOW}-1}^+ := o_k$
- 11: transmit to agent $i - 1$: $p_{\text{NOW}+1}, p_{\text{NOW}}^+, p_{\text{NOW}-1}^+, \widehat{\gamma}'(p_{\text{NOW}-1}^+), \widehat{D}_\lambda(p_{\text{NOW}-2}, p_{\text{NOW}-1}^+)$, and transmit to agent $i + 1$: $\text{NOW} + 1$
- 12: update the set BUFFERARC and the counter NOW :
$$\text{BUFFERARC}^+ := \{o_k, \dots, o_{n_o+1}\}, \quad \text{NOW}^+ := \text{NOW} + 1$$
- 13: calculate $v_i(t)$: $v_i(t) := \operatorname{sat}(v_0 + k_{\text{prop}}(\widehat{L}(P_i, P_{i+1}) - \widehat{L}(P_{i-1}, P_i)))$

Chapter 4

ISS properties of discrete-time consensus algorithms

Consensus algorithms and conditions for their convergence have been widely studied. An incomplete list of references includes continuous-time consensus algorithms [11, 32, 34], discrete-time consensus, flocking and rendezvous [19, 40, 26], time-delayed and asynchronous properties of consensus algorithms [38, 3, 6, 30, 32], and consensus over random graphs [15]. Extensions of consensus algorithms to achieve different consensus values has also been investigated, see [10, 4]. Consensus algorithms have already been proposed for the implementation of distributed data fusion techniques [38, 31].

Since the characteristics of networked systems require distributed algorithms to be robust to disturbances (see Chapter 3), a natural question to ask is how the consensus algorithms perform in the presence of undesired inputs. Previous work on this topic includes [42], [33], and [24]. In [42] the authors present a stochastic analysis of consensus algorithms in the presence of additive zero-mean

noise for fixed topologies. In [33] the authors show the asymptotic behavior of the ϵ -consensus time as the number of the nodes in the network grows for fixed topologies.

In this chapter we investigate the robustness properties of discrete-time consensus algorithms following a worst case analysis rather than a stochastic analysis. A early work in this sense is [24], where the authors prove that for continuous-time consensus algorithms the disagreement dynamics is ISS. Regarding discrete-time consensus algorithms, we include here two complementary analyses of the fact that disagreement values asymptotically remain within bounded distance of the consensus space when noise is present. The first proof is similar to that of [24], and provides a first theoretical explanation for the fact that linear consensus algorithms are IOS. Alternatively, the second approach goes further in producing specific estimates for the input-output gain and transient bounds that characterize the behavior of the disagreement output in terms of bounded disturbances, and is valid also when the network topology is not fixed. This second proof generalizes a certain contraction result from [6]. After this, we further analyze the robustness of the nodes' states to disturbance. We establish that, in general, discrete-time consensus algorithms are only iISnS and the consensus value is driven by the integral of the noise. Finally, we see how, in a single leader protocol, discrete-time consensus is exactly ISS. Our work leads to a deeper understanding of robustness properties of discrete-time consensus algorithms.

4.1 Review of ISS concepts

This section introduces notation and main concepts for ISS discrete-time systems as introduced in [20, 2, 37], which will be employed in the sequel.

We let \mathbb{N}_0 and $\bar{\mathbb{R}}_+$ denote the non-negative integer and real numbers, respectively. We let $\mathbf{1}_n = (1, \dots, 1)^T \in \mathbb{R}^n$. For $x \in \mathbb{R}^n$ and $D \in \mathbb{R}^{n \times k}$, we let $|x|$ and $|D|$ denote the Euclidean and the induced norm of x and of D , respectively. Given functions $\phi: \mathbb{N}_0 \rightarrow \mathbb{R}^n$ and $D: \mathbb{N}_0 \rightarrow \mathbb{R}^{n \times m}$, we define $\|\phi\| = \sup\{|\phi(t)| \mid t \in \mathbb{N}_0\}$, $\|D\| = \sup\{|D(t)| \mid t \in \mathbb{N}_0\}$, and $\bar{D} = \sup\{(\sum_{j=1}^m |D_i^j(t)|^2)^{\frac{1}{2}} \mid t \in \mathbb{N}_0, j \in \{1, \dots, n\}\}$.

A function $\gamma: \bar{\mathbb{R}}_+ \rightarrow \bar{\mathbb{R}}_+$ is a \mathcal{K} -function if it is continuous, strictly increasing, and $\gamma(0) = 0$; and it is a \mathcal{K}_∞ -function if it is a \mathcal{K} -function and if it is unbounded. A function $\beta: \bar{\mathbb{R}}_+ \times \bar{\mathbb{R}}_+ \rightarrow \bar{\mathbb{R}}_+$ is a \mathcal{KL} -function if, for each $t \in \bar{\mathbb{R}}_+$, the function $s \mapsto \beta(s, t)$ is a \mathcal{K} -function, and for each $s \in \bar{\mathbb{R}}_+$, $t \mapsto \beta(s, t)$ is decreasing and $\beta(s, t) \rightarrow 0$ as $t \rightarrow +\infty$.

Consider the discrete-time nonlinear system

$$x(t+1) = f(t, x(t), u(t)), \quad (4.1)$$

where t takes value in \mathbb{N}_0 , x takes value in \mathbb{R}^n , and u takes value in \mathbb{R}^m . We assume that $f: \mathbb{N}_0 \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is continuous.

System (4.1) is *input-to-state stable (ISS)* if there exist a \mathcal{KL} -function β and a \mathcal{K} -function γ such that, for each initial condition $x_0 \in \mathbb{R}^n$ at time $t_0 \in \mathbb{N}_0$ and for each bounded input $u: \mathbb{N}_0 \rightarrow \mathbb{R}^m$, the system evolution x satisfies, for each $t \geq t_0$,

$$|x(t)| \leq \beta(|x_0|, t - t_0) + \gamma(\|u\|).$$

Analogously, system (4.1) with an output $y = h(x)$ is *input-to-output stable*

(IOS), respectively *input-to-output stable (IOS) with gain* γ , if there exist a \mathcal{KL} -function β and a \mathcal{K} -function γ such that, for each initial condition $x_0 \in \mathbb{R}^n$ at time $t_0 \in \mathbb{N}_0$ and for each bounded input $u: \mathbb{N}_0 \rightarrow \mathbb{R}^m$, the output $y(t) = h(x(t))$ satisfies, for each $t \geq t_0$,

$$|y(t)| \leq \beta(|x_0|, t - t_0) + \gamma(\|u\|),$$

respectively

$$|y(t)| \leq \max \left\{ \beta(|x_0|, t - t_0), \gamma(\|u\|) \right\}.$$

System (4.1) is *integral input-to-state neutrally stable (iISnS)*, see [2], if there exist \mathcal{K}_∞ -functions α , σ and γ such that, for each initial condition $x_0 \in \mathbb{R}^n$ at time $t_0 \in \mathbb{N}_0$ and for each bounded input $u: \mathbb{N}_0 \rightarrow \mathbb{R}^m$, the system evolution x satisfies, for each $t \geq t_0$,

$$\alpha(|x(t)|) \leq \gamma(|x_0|) + \sum_{k=t_0}^{t-1} \sigma(|u_k|).$$

4.2 Consensus algorithms with inputs and outputs

In this section we present our problem statement regarding the ISS stability of discrete-time consensus systems. The consensus problem presentation mainly follows that of Moreau in [30]. A discrete-time consensus algorithm is defined as:

$$x(t+1) = A(t)x(t), \tag{4.2}$$

where t takes value in \mathbb{N}_0 and x takes value in \mathbb{R}^n . We let $x_i(t)$ and $A_{ij}(t)$ denote the entries of $x(t)$ and of $A(t)$, respectively. For each $t \in \mathbb{N}_0$, the matrix $A(t)$ gives

rise to a directed graph $G(t) = \{\{1, \dots, n\}, E(t)\}$ where $(j, i) \in E(t)$ if and only if $i \neq j$ and $A_{ij}(t) > 0$. In general, we will consider the following assumptions:

(A1) Non-degenerate Averaging: There exists $\alpha > 0$, such that

1. $A_{ii}(t) \geq \alpha$, for all i, t ,
2. $A_{ij}(t) \in \{0\} \cup [\alpha, 1]$, for all i, j, t .

(A2) Stochasticity: $\sum_{j=1}^n A_{ij}(t) = 1$, for all i, t .

(A3) Uniform Connectivity: There exists $B > 0$ such that for every $t \in \mathbb{N}_0$ the graph $\{\{1, \dots, n\}, \cup_{s \in [t, t+B]} E(s)\}$ has a node connected¹ to all other nodes.

Let us now briefly review a definition and a theorem from [30]. Assume that Φ is a set of equilibrium points for a time-dependent discrete-time dynamical system $x(t+1) = f(t, x(t))$. The dynamical system is *uniformly globally attractive* with respect to Φ if for each $\phi_1 \in \Phi$, for all $c_1, c_2 > 0$ and for all $t_0 \in \mathbb{N}$, there exists $T > 0$ such that every solution x has the following property: if $|x(t_0) - \phi_1| < c_1$, then there exists $\phi_2 \in \Phi$ such that $|x(t) - \phi_2| < c_2$ for all $t \geq t_0 + T$.

Theorem 4.1 (Uniform asymptotic consensus, [30]) *Under Assumptions (A1), (A2) and (A3), the discrete-time dynamical system (4.2) is uniformly global attractive with respect to the collection of equilibrium solutions $x_1(t) = \dots = x_n(t) = \text{constant}$.*

In other words, Theorem 4.1 states that the consensus algorithm (4.2) converges uniformly and asymptotically to the vector subspace generated by 1_n .

¹A node i is connected to a node $j \neq i$ in a directed graph if there exists an oriented path from i to j in the directed graph.

Motivated by this successful analysis, we shall consider the system

$$x(t+1) = A(t)x(t) + D(t)u(t), \quad (4.3)$$

where $t \mapsto u(t) \in \mathbb{R}^k$ is a disturbance possibly coming from noise or communication errors and $A(t)$, $t \in \mathbb{N}_0$, satisfies assumptions (A1), (A2) and (A3). A reasonable question to ask is how the evolution of the trajectories x is affected by the noise u . We will address this in the following section relying on the following assumption.

(A4) Uniformly Bounded Input Gain: The operator norm $\|D\|$ is bounded, i.e., the induced norm of $D(t)$, for $t \in \mathbb{N}_0$, is uniformly bounded.

We also will consider the following two outputs for the dynamical system (4.3):

$$y_{\max\text{-min}} = \max_{i \in \{1, \dots, n\}} x_i - \min_{i \in \{1, \dots, n\}} x_i,$$

$$y_{\text{err}} = \begin{pmatrix} x_1 - x_2 \\ \vdots \\ x_{n-1} - x_n \end{pmatrix} = Px,$$

for

$$P = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & -1 \end{pmatrix} \in \mathbb{R}^{(n-1) \times n}.$$

Both outputs can be thought of as error signals quantifying a measure of disagreement among the components of the state.

4.3 Consensus algorithms with error outputs are IOS

In this section we provide two complementary analyses of the fact that discrete-time consensus is IOS. First, we present a main proof that parallels that of continuous-time consensus systems in [24]. Second, we extend a result recently presented in [6] that leads to specific gain bounds for the evolution of the second type of disagreement affected by noise.

4.3.1 IOS with respect to pairwise error

Define $T \in \mathbb{R}^{n \times n}$ by

$$T = \begin{pmatrix} P \\ \frac{1}{n} \mathbf{1}^T \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & -1 \\ 1/n & 1/n & \dots & 1/n & 1/n \end{pmatrix}.$$

For the system (4.3), consider the change of variables

$$z(t) = Tx(t) = \begin{pmatrix} y_{\text{err}}(t) \\ x_{\text{ave}}(t) \end{pmatrix},$$

where $y_{\text{err}}(t) \in \mathbb{R}^{n-1}$, and $x_{\text{ave}}(t) \in \mathbb{R}$ is the average of the components of x . It is easy to verify that T is invertible and that, therefore, system (4.3) reads in the new variables

$$z(t+1) = TA(t)T^{-1}z(t) + TD(t)u(t).$$

Lemma 4.1 (Block decomposition) *If $A(t)$, $t \in \mathbb{N}_0$, satisfies the Stochasticity Assumption (A2), then there exists $A_{err}(t) \in \mathbb{R}^{(n-1) \times (n-1)}$ and $c_{err}(t) \in \mathbb{R}^{1 \times (n-1)}$, for $t \in \mathbb{N}_0$, such that*

$$TA(t)T^{-1} = \begin{pmatrix} A_{err}(t) & 0_{(n-1) \times 1} \\ c_{err}(t) & 1 \end{pmatrix}.$$

Moreover, if $A(t) = A^T(t)$, then $c_{err}(t) = 0_{1 \times (n-1)}$.

Proof. Define $e_n = \begin{pmatrix} 0 & \dots & 0 & 1 \end{pmatrix}^T$ and $A_z(t) = TA(t)T^{-1}$. Note that $A(t)1_n = 1_n$ by Assumption (A2) and that $T1_n = e_n$ by definition of T . From the equality $TA(t)1_n = A_z(t)T1_n$, we derive:

$$e_n = A_z(t)e_n.$$

This implies that the last column of $A_z(t)$ is always equal to e_n ; this concludes the proof of the first statement. The symmetry result is proved as follows. Note that $T^T A_z^T(t) = A(t)T^T$ by symmetry of $A(t)$ and that $T^T e_n = \frac{1}{n}1_n$ by definition of T . Because T is invertible, the last equality can also be rewritten as $e_n = (T^T)^{-1}(\frac{1}{n}1_n)$. Now, we compute:

$$T^T A_z^T(t)e_n = A(t)T^T e_n = \frac{1}{n}A(t)1_n = \frac{1}{n}1_n,$$

which implies $A_z^T(t)e_n = e_n$. This fact in turn is equivalent to $c_{err}(t) = 0_{1 \times (n-1)}$. ■

The lemma states in careful words the following intuition. Because of the definition of z and of the special structure of A , the variable x_{ave} plays no role in the evolution of y_{err} . Accordingly, we define the *error system* by

$$y_{err}(t+1) = A_{err}(t)y_{err}(t) + D_{err}(t)u(t), \quad (4.4)$$

and the *average system* by

$$x_{\text{ave}}(t+1) = x_{\text{ave}}(t) + c_{\text{err}}(t)y_{\text{err}}(t) + D_{\text{ave}}(t)u(t), \quad (4.5)$$

for $D_{\text{err}}(t) = PD(t)$, and $D_{\text{ave}}(t) = \frac{1}{n}1_n^T D(t)$. Finally, we are ready for the main result of this section.

Theorem 4.2 *Under Assumptions (A1), (A2), (A3) and (A4), the following equivalent statements hold:*

1. *The system (4.3) with output y_{err} is IOS.*
2. *The error system (4.4) is ISS.*

Proof. By Theorem 4.1, Assumptions (A1), (A2), and (A3) imply that system (4.3) with $u(t) = 0$, $t \in \mathbb{N}_0$, is uniformly global attractive with respect to the collection of equilibrium solutions $x_1(t) = \cdots = x_n(t) = \text{constant}$. The matrix T maps the equilibrium solutions for (4.3) with $u(t) = 0$ into the trivial equilibrium solutions for the error system, i.e., $t \mapsto y_{\text{err}}(t) = 0$. This implies that the origin is uniformly global attractive for the error system (4.4) with $u(t) = 0$. Because of linearity, we conclude that the origin is uniformly globally exponentially stable for (4.4). This implies the existence of $\Gamma \in \bar{\mathbb{R}}_+$ and $\lambda \in [0, 1)$, such that $|y_{\text{err}}(t)| \leq \Gamma \lambda^t |y_{\text{err}}(0)|$ for all $t \in \mathbb{N}_0$. If the input sequence is different from the zero sequence, then we write the trajectory of (4.4) as

$$y_{\text{err}}(t+1) = \prod_{j=0}^t A_{\text{err}}(j)y_{\text{err}}(0) + \sum_{j=0}^t A_{\text{err}}(t) \cdots A_{\text{err}}(j+1)D_{\text{err}}(j)u(j).$$

We then compute the following upper bounds:

$$\begin{aligned}
\left| \sum_{j=0}^t \left(\prod_{k=j+1}^t A_{\text{err}}(k) \right) D_{\text{err}}(j) u(j) \right| &\leq \|D_{\text{err}}\| \|u\| \sum_{j=0}^t \left| \prod_{k=j+1}^t A_{\text{err}}(k) \right| \\
&\leq \Gamma \|D_{\text{err}}\| \|u\| \sum_{j=0}^t \lambda^{t-j} = \Gamma \|D_{\text{err}}\| \|u\| \sum_{k=0}^t \lambda^k \\
&\leq \Gamma \|D_{\text{err}}\| \|u\| \sum_{k=0}^{\infty} \lambda^k = \Gamma \|D_{\text{err}}\| \|u\| \frac{1}{1-\lambda},
\end{aligned}$$

and therefore

$$|y_{\text{err}}(t+1)| \leq \Gamma \lambda^{t+1} |y_{\text{err}}(0)| + \Gamma \frac{1}{1-\lambda} \|D_{\text{err}}\| \|u\|.$$

This concludes the proof of fact (ii). Fact (i) is clearly equivalent. ■

Remark 4.1 (Deviation from the average) *Consider now the deviation from the average $\delta x(t) = x(t) - x_{\text{ave}}(t)$. As seen in Theorem 4.2, the origin of the system (4.4) is exponentially stable when $u(t) = 0$, i.e., $x(t)$ exponentially approaches $c1_n$ where c depends on the initial condition and on the matrices $A(t)$. As a consequence $\delta x(t)$ also will approach exponentially the origin.*

Remark 4.2 (Asymptotic gain) *The ISS gain $\Gamma \frac{1}{1-\lambda} \|D_{\text{err}}\|$ can easily be calculated if λ is known. An estimate for the value of λ is usually difficult to calculate unless $A(t)$ is constant, in this case λ is the largest eigenvalue of $A(t)$ smaller than 1. For some topologies and weights it is possible to calculate the eigenvalues of $A(t)$ in closed form and estimate the asymptotic behavior of the ISS gain as the number of nodes approaches infinity. Let us consider the ring topology for a network of n nodes*

4.3.2 IOS with respect to max-min error

As is well known [18], finding good estimates of the ISS gains is important to determine via small-gain theorems the classes of systems that can be stably interconnected to the multi-agent system. As a byproduct of an extension of a result in [6], we will prove IOS of system (4.3) with output $y_{\max\text{-min}}$ and obtain numerical bounds on the comparison gains.

Proposition 4.1 (Max-min contraction) *Let x be the solution to (4.3) and $C = n^2(B + 1)$. Under Assumptions (A1), (A2), (A3) and (A4), there exists a sequence $\{t_q\}_{q \in \mathbb{N}_0}$, where $t_{q+1} - t_q \leq C$, such that:*

$$y_{\max\text{-min}}(t_{q+1}) \leq (1 - \alpha^{(n-1)C+n})y_{\max\text{-min}}(t_q) + 2C\bar{D}\|u\|. \quad (4.6)$$

Proof. Assume that $t_0 = 0$, and $\min_i x_i(0) = 0$. Consider the n^2 intervals $[j(B + 1), j(B + 1) + B]$, for $j \in 1, \dots, n^2$. Then, by assumption (A3), there is a node that is connected to all other nodes in each one of these intervals. In particular, there exists a node i_0 which satisfies this property for at least n of these intervals. Otherwise, let ℓ_i , for $i \in \{1, \dots, n\}$, be the number of intervals $[j(B + 1), j(B + 1) + B]$ where i is the node connected to all other nodes by (A3). If $\ell_i < n$ for all $i \in \{1, \dots, n\}$, then $\sum_{i=1}^n \ell_i < n^2$ which is a contradiction with the fact that the number of intervals must sum up to n^2 . Let us denote by $s_0 = 0$ and let $[s_1(B + 1), s_1(B + 1) + B], \dots, [s_n(B + 1), s_n(B + 1) + B]$ be the first n consecutive time intervals for which i_0 is connected to all other nodes. Observe that $s_\ell(B + 1) + B + 1 \leq s_{\ell+1}(B + 1)$ for all $\ell \in \{1, \dots, n - 1\}$.

Let us use the shorthand notation $x_{i_0}(0) = x_0$, $\max_i x_i(0) = M_0$, and $c_0 = M_0 - x_0$. We define the following properties \bar{P}_k and P_k associated with i_0 . We say

that \bar{P}_k holds if there exist at least k nodes and a time $\tau_k \leq (s_{k-1} + 1)(B + 1) \leq s_k(B + 1)$ for which

$$x_i(\tau_k) \leq x_0 + (1 - \alpha^{(k-1)C+k})c_0 + \tau_k \bar{D} \|u\|,$$

Analogously, we say that P_k holds if there exist at least k nodes and a time $\tau_k \leq (s_{k-1} + 1)(B + 1) \leq s_k(B + 1)$ for which

$$x_i(\tau_k) \geq \alpha^{(k-1)C+k} x_0 - \tau_k \bar{D} \|u\|.$$

We will prove by induction on k that \bar{P}_n and P_n will eventually hold for all indices $i \in \{1, \dots, n\}$ at the same time $\tau_n \leq s_n(B + 1) \leq C$.

To see this define \bar{S}_k and S_k to be the sets of nodes for which properties \bar{P}_k and P_k are true. For $k = 1$ there exists at least i_0 such that at time $\tau_1 = 1 \leq (s_0 + 1)(B + 1) \leq s_1(B + 1)$ satisfies:

$$\begin{aligned} x_{i_0}(1) &= A_{i_0}^{i_0}(0)x_0 + A_{i_0}^j(0)x_j(0) + D_{i_0}^j(0)u_j(0) \\ &\leq A_{i_0}^{i_0}(0)x_0 + \sum_{j \neq i_0} A_{i_0}^j(0)M_0 + \bar{D} \|u\| \\ &= \sum_{j=1}^n A_{i_0}^j(0)x_0 + \sum_{j \neq i_0} A_{i_0}^j(0)c_0 + \bar{D} \|u\|. \end{aligned}$$

Here we used the following inequality:

$$-\bar{D} \|u\| \leq \sum_{j=1}^n D_i^j(t)u_j(t) \leq \bar{D} \|u\|.$$

Using (A2) and the fact that $\sum_{j \neq i_0} A_{i_0}^j(0) = 1 - A_{i_0}^{i_0} \leq 1 - \alpha$ because of (A1), we have that

$$x_{i_0}(1) \leq x_0 + (1 - \alpha)c_0 + \bar{D} \|u\|$$

Therefore \bar{P}_1 is satisfied at $\tau_1 = 1 \leq (s_0 + 1)(B + 1) \leq s_1(B + 1)$.

On the other hand, since $A_i^j(0) \geq 0$, $x_j(0) \geq 0$, $\forall i, j$ and $\underline{D} < 0$, we have that

$$\begin{aligned} x_{i_0}(1) &\geq A_{i_0}^{i_0}(0)x_0 + D_{i_0}^j(0)u_j(0) \\ &\geq \alpha x_0 - \bar{D}\|u\|, \end{aligned}$$

therefore P_1 also holds at $\tau_1 = 1 \leq (s_0 + 1)(B + 1) \leq s_1(B + 1)$ with the index i_0 . Suppose now that \bar{P}_k and P_k hold for $k > 1$ at the same time $\tau_k \leq (s_{k-1} + 1)(B + 1) \leq s_k(B + 1)$. Then we will prove that \bar{P}_{k+1} and P_{k+1} will hold at the same time $\tau_{k+1} \leq (s_k + 1)(B + 1) \leq s_{k+1}(B + 1)$. We reason for \bar{P}_{k+1} being the treatment for P_{k+1} analogous.

Let $\tau_k < \eta_1$ be the first time a connection between a node in \bar{S}_k^c , say l_0 , and a node in \bar{S}_k , say j , is established. Two situations are possible. Either (i) $A_j^{l_0}(\eta_1) \neq 0$; i.e. a node in \bar{S}_k^c influences a node in \bar{S}_k or (ii) $A_{l_0}^j(\eta_1) \neq 0$; i.e. a node in \bar{S}_k influences a node in \bar{S}_k^c .

For all time $\tau_k \leq t < \eta_1$, and $l \in \bar{S}_k$ we have that:

$$\begin{aligned} x_l(t+1) &= \sum_{i \in \bar{S}_k} A_l^i(t)x_i(t) + \sum_{i=1}^n D_l^i(t)u_i(t) \\ &\leq \sum_{i \in \bar{S}_k} A_l^i(t)(x_0 + (1 - \alpha^{(k-1)C+k})c_0 \\ &\quad + t\bar{D}\|u\|) + \bar{D}\|u\| \\ &\leq x_0 + (1 - \alpha^{(k-1)C+k})c_0 + (t+1)\bar{D}\|u\|. \end{aligned}$$

Assume that (ii) holds. At time $t = \eta_1$ and for l_0 , we have that:

$$\begin{aligned}
& x_{l_0}(\eta_1 + 1) \\
& \leq \sum_{i \in \bar{S}_k} A_{l_0}^i(\eta_1)(x_0 + (1 - \alpha^{(k-1)C+k}))c_0 \\
& \quad + \sum_{i \in \bar{S}_k} A_{l_0}^i(\eta_1)\eta_1 \bar{D}\|u\| + \sum_{i \in \bar{S}_k^c} A_{l_0}^i(\eta_1)x_i(\eta_1) \\
& \quad + D_{l_0}^j(\eta_1)u_j(\eta_1) \\
& \leq \sum_{i \in \bar{S}_k} A_{l_0}^i(\eta_1)(x_0 + (1 - \alpha^{(k-1)C+k}))c_0 + \\
& \quad \sum_{i \in \bar{S}_k^c} A_{l_0}^i(\eta_1)M_0 + (\eta_1 + 1)\bar{D}\|u\|.
\end{aligned}$$

Let us denote by $a = \sum_{i \in \bar{S}_k^c} A_i^{l_0} \neq 0$ and $1 - a = \sum_{i \in \bar{S}_k} A_i^{l_0} \geq \alpha$. Using now that $M_0 = x_0 + c_0$, we can reorganize the above terms in

$$\begin{aligned}
x_{l_0}(\eta_1 + 1) & \leq x_0 + ((1 - \alpha^{(k-1)C+k})(1 - a) + a)c_0 + (\eta_1 + 1)\bar{D}\|u\| \\
& = x_0 + (1 - \alpha^{(k-1)C+k}(1 - a))c_0 + (\eta_1 + 1)\bar{D}\|u\| \\
& \leq x_0 + (1 - \alpha^{(k-1)C+k+1})c_0 + (\eta_1 + 1)\bar{D}\|u\|.
\end{aligned}$$

For every other node l in \bar{S}_k that only receives influence from nodes in \bar{S}_k we can argue that

$$\begin{aligned}
x_l(\eta_1 + 1) & \leq x_0 + (1 - \alpha^{(k-1)C+k})c_0 + (\eta_1 + 1)\bar{D}\|u\| \\
& \leq x_0 + (1 - \alpha^{(k-1)C+k+1})c_0 + (\eta_1 + 1)\bar{D}\|u\|.
\end{aligned}$$

Assume now that (i) holds and suppose that there exist a sequence $\tau_k \leq \eta_1 \leq \dots \leq \eta_r$ of times such that a node from \bar{S}_k^c influences a node from \bar{S}_k and, moreover, during this time, nodes from \bar{S}_k do not get to influence nodes from \bar{S}_k^c . Reasoning as above, we can conclude that $\forall l \in \bar{S}_k$ we have

$$x_l(\eta_r + 1) \leq x_0 + (1 - \alpha^{(k-1)C+k+r})c_0 + (\eta_r + 1)\bar{D}\|u\|.$$

Now there must exist a first time $\eta_r < \rho_k$ when a node from \bar{S}_k influences a node from \bar{S}_k^c , that is eventually (ii) holds. We have that $\tau_k \leq (s_{k-1} + 1)(B + 1) \leq s_k(B + 1)$ and there exists an interval $[s_k(B + 1), s_k(B + 1) + B]$ where i_0 is connected to all other nodes. This implies that for every $j \in \bar{S}_k^c$ there is a path $(i_0, l_1), \dots, (l_k, j) \in \cup_{[s_{k+1}(B+1), s_{k+1}(B+1)+B]} E(t)$. But if nodes in \bar{S}_k only influence nodes in \bar{S}_k from τ_k on, this would imply $j \in \bar{S}_k$, which is a contradiction. Therefore there must exist a direct connection from a node of \bar{S}_k to a node in \bar{S}_k^c , j_0 , at a time $\rho_k \leq s_k(B + 1) + B$.

It is easy to see that at time $\rho_k + 1$ we have

$$x_i(\rho_k + 1) \leq x_0 + (1 - \alpha^{(k-1)C+k+r+1})c_0 + (\rho_k + 1)\bar{D}\|u\|$$

for $i \in \bar{S}_k \cup \{j_0\}$. Obviously we have that $r \leq C$ so we can upper bound the previous quantity by

$$x_i(\rho_k + 1) \leq x_0 + (1 - \alpha^{kC+k+1})c_0 + (\rho_k + 1)\bar{D}\|u\|$$

and take $\tau_{k+1} = \rho_k + 1$. Observe that $\tau_{k+1} \leq (s_k + 1)(B + 1) \leq s_{k+1}(B + 1)$, so we have been able to prove the induction.

From the above induction we can conclude that there exists a time $\tau_n \leq s_n(B + 1) \leq n^2(B + 1) = C$ for which \bar{P}_n and P_n hold. In particular this implies that

$$y_{\max\text{-min}}(\tau_n) \leq (1 - \alpha^{(n-1)C+n})M_0 + 2\tau_n\bar{D}\|u\|.$$

In other words, setting $t_1 = \tau_n$ we have that

$$y_{\max\text{-min}}(t_1) \leq (1 - \alpha^{(n-1)C+n})y_{\max\text{-min}}(t_0) + 2C\bar{D}\|u\|. \quad (4.7)$$

This argument can be repeated for the new vector $\tilde{x} = x - \min x_i(t_1)1_n$, satisfying the discrete-time equation (4.3) for $t \geq t_1$ and $\min_i \tilde{x}_i(t_1) = 0$. ■

We now state the main result of this section.

Theorem 4.3 *Under Assumptions (A1), (A2), (A3) and (A4), the system (4.3) with output $y_{\max\text{-min}}$ is IOS with gain*

$$\gamma(s) = 4C\bar{D}\left(1 + \frac{1}{\alpha^R}\right)s, \quad (4.8)$$

where $C = n^2(B + 1)$ and $R = (n - 1)C + n$.

Proof. Let us consider first the subsequence $\{y_{\max\text{-min}}(t_q)\}_{q \in \mathbb{N}_0}$ as defined in Proposition 4.1. Iterating equation (4.6) we can upper bound $|y_{\max\text{-min}}(t_q)|$ as follows:

$$\begin{aligned} |y_{\max\text{-min}}(t_q)| &\leq (1 - \alpha^R)^q |y_{\max\text{-min}}(0)| + 2C\bar{D}\|u\| \sum_{i=0}^{q-1} (1 - \alpha^R)^i \\ &\leq (1 - \alpha^R)^q |y_{\max\text{-min}}(0)| + 2C\bar{D}\|u\| \sum_{i=0}^{\infty} (1 - \alpha^R)^i \\ &= (1 - \alpha^R)^q |y_{\max\text{-min}}(0)| + \frac{2C\bar{D}}{\alpha^R} \|u\|. \end{aligned}$$

Observe that the sequence $\{t_q\}$ satisfies $t_q \rightarrow +\infty$. Then for all t , there exists $q(t)$ such that $t_{q(t)} \leq t \leq t_{q(t)+1}$. It is easy to see that

$$\begin{aligned} y_{\max\text{-min}}(t) &= y_{\max\text{-min}}(t_{q(t)} + (t - t_{q(t)})) \\ &\leq (1 - \alpha^R)^{q(t)} |y_{\max\text{-min}}(0)| + 2C\bar{D}\left(1 + \frac{1}{\alpha^R}\right) \|u\|. \end{aligned}$$

This implies that:

$$y_{\max\text{-min}}(t) \leq \max \left\{ 2(1 - \alpha^R)^{q(t)} |y_{\max\text{-min}}(0)|, 4C\bar{D}\left(1 + \frac{1}{\alpha^R}\right) \|u\| \right\},$$

that is, the IOS gain for the system (4.3) with output $y_{\max\text{-min}}$ is

$$\gamma(s) = 4C\bar{D}\left(1 + \frac{1}{\alpha^R}\right)s.$$

■

4.4 Consensus algorithms are iISnS

In the previous sections we analyzed the effect of the noise on the consensus behavior of algorithms such as (4.3). We now move our attention on the value of the consensus variable and how it is affected by the noise. We saw that y_{err} reaches asymptotically a ball centered at the origin with radius depending on $\|u\|$. This implies that, in general, the nodes are not able to reach asymptotic consensus but, as the effect of the initial condition vanishes, the trajectories x_i evolve close to each other. We will now show that in general the trajectories x_i do not stay close to any constant value. If no further information on $A(t)$ is given, an estimate of the consensus value is given by $x_{\text{ave}}(t)$, which evolves according to the integral of the noise.

Theorem 4.4 *Under Assumptions (A1), (A2), (A3), and (A4), the following statements hold:*

1. *the system (4.3) is iISnS with respect to the input u ,*
2. *the system (4.5) is iISnS with respect to the inputs y_{err} and u .*

Proof. We shall start with the first statement. Calculating the trajectory of (4.3) leads to:

$$x(t+1) = \prod_{j=0}^t A(j)x(0) + \sum_{j=0}^t A(t) \dots A(j+1)D(j)u(j).$$

We can then upper bound the norm of $x(t+1)$ as follows:

$$|x(t+1)| \leq \left| \prod_{j=0}^t A(j) \right| |x(0)| + \sum_{j=0}^t |A(t) \dots A(j+1)| \|D\| |u(j)|.$$

Since the matrices $A(t)$ are stochastic we have that $|\prod_{j=0}^m A(j)| \leq \sqrt{n}$, for all $m \in \mathbb{N}$, and therefore

$$|x(t+1)| \leq \sqrt{n}|x(0)| + \sum_{j=0}^t \sqrt{n} \|D\| |u(j)|.$$

To verify (ii) just note that:

$$x_{\text{ave}}(t+1) = x_{\text{ave}}(0) + \sum_{j=0}^t [c_{\text{err}}(j), D_{\text{ave}}(j)] [y_{\text{err}}(j), u(t)]^T.$$

Let $\|\tilde{D}\|$ be the upper bound of the induced norm of $[c_{\text{err}}(j), D_{\text{ave}}(j)]$, then:

$$|x_{\text{ave}}(t+1)| = |x_{\text{ave}}(0)| + \sum_{j=0}^t \|\tilde{D}\| |[y_{\text{err}}(j), u(t)]|.$$

■

We will now show that the bound, given for the consensus value in Theorem 4.4, can be tightened when the matrices $A(t)$ and $D(t)$ have certain properties.

Let \mathcal{C} be the set of all pairs of sequences $\{A(t)\}_{t \in \mathbb{N}_0}$, $\{D(t)\}_{t \in \mathbb{N}_0}$ that describe the algorithms in (4.3). We shall consider now two disjoint subsets of \mathcal{C} :

1. $\mathcal{C}_1 = \{(\{A(t)\}_{t \in \mathbb{N}_0}, \{D(t)\}_{t \in \mathbb{N}_0}) \mid A(t) = A(t)^T \text{ for all } t \in \mathbb{N}_0\}$,

2. $\mathcal{C}_2 = \{(\{A(t)\}_{t \in \mathbb{N}_0}, \{D(t)\}_{t \in \mathbb{N}_0}) \mid A(t) = \begin{pmatrix} 1 & \mathbf{0}^T \\ Q(t) & A_{\text{net}}(t) \end{pmatrix} \text{ and}$

$$D(t) = \begin{pmatrix} \mathbf{0}^T \\ D_{\text{net}}(t) \end{pmatrix} \text{ for all } t \in \mathbb{N}_0\}, \text{ where } A_{\text{net}}(t) \in \mathbb{R}^{n-1 \times n-1}.$$

The graphs of the consensus algorithms that belong to \mathcal{C}_1 are undirected and the weights are symmetric while the graphs of the consensus algorithms that belong to \mathcal{C}_2 have a node that behaves as a leader (i.e., does not average the value of its

internal variable with anybody) and that is not affected by noise. For this two special subsets of \mathcal{C} we will further analyze how the noise affects the consensus variable value.

4.4.1 Consensus value for algorithms in \mathcal{C}_1

When $u(t) = 0$, $x_{\text{ave}}(t)$ is constant with time and is the asymptotic value of the consensus variable, as is well known. When $u(t) \neq 0$, $x_{\text{ave}}(t)$ is not constant but, because $A(t) = A(t)^T$, $c_{\text{err}}(j) = 0_{1 \times (n-1)}$ for all $j \in \mathbb{N}_0$ (see Lemma 4.1). Hence, from Theorem 4.4, it is clear that the dynamics of x_{ave} is iISnS respect to the sole input u . From the difference equation (4.5) for x_{ave} we can conclude that the network is overall behaving like an integrator: the states $x_i(t)$ are pushed around by the average $\frac{1}{n}1_n D(t)u(t)$.

4.4.2 Consensus value for algorithms in \mathcal{C}_2

When $u(t) = 0$, $x_1(t)$ is constant with time and is the asymptotic value of the consensus variable simply because node 1 behaves as a leader and does not average with anybody. Even if noise is introduced, the value of $x_1(t)$ does not change and, therefore, the consensus value for the nodes is still $x_1(0)$. The noise, though, prevents the nodes from reaching $x_1(0)$ asymptotically as shown in the next theorem.

Define $\tilde{T} \in \mathbb{R}^{n \times n}$ by:

$$\tilde{T} = \begin{pmatrix} 1 & 0_{1 \times (n-1)} \\ -1_{n-1} & I_{n-1} \end{pmatrix}.$$

For the system (4.3), consider the change of variables

$$\tilde{z}(t) = \tilde{T}x(t) = \begin{pmatrix} x_1(t) \\ \tilde{y}_{\text{err}}(t) \end{pmatrix},$$

where $\tilde{y}_{\text{err}}(t) = [x_1(t) - x_2(t), \dots, x_1(t) - x_n(t)]^T \in \mathbb{R}^{n-1}$. The dynamics for the new variable $\tilde{z}(t)$ is:

$$\tilde{z}(t+1) = \tilde{T}A(t)\tilde{T}^{-1}z(t) + \tilde{T}D(t)u(t).$$

It is easy to check that:

$$\tilde{y}_{\text{err}}(t+1) = A_{\text{net}}(t)\tilde{y}_{\text{err}}(t) + D_{\text{net}}(t)u(t). \quad (4.9)$$

Let $\tilde{x}(t) = [x_2(t), \dots, x_n(t)]^T$, then:

$$\tilde{x}(t+1) = A_{\text{net}}(t)\tilde{x}(t) + Q(t)x_1(t) + D_{\text{net}}(t)u(t). \quad (4.10)$$

Lemma 4.2 *Let $(\{A(t)\}_{t \in \mathbb{N}_0}, \{D(t)\}_{t \in \mathbb{N}_0}) \in \mathcal{C}_2$. Under Assumptions (A1), (A2), (A3) and (A4), the following equivalent statements hold:*

1. *The system (4.3) with output \tilde{y}_{err} is IOS.*
2. *The error system (4.9) is ISS.*
3. *The system (4.10) is ISS respect to the inputs x_1 and u .*

Proof. The proof of the equivalence of (i) and (ii) proceeds along the same lines as that for Theorem 4.2 and, therefore, we will use the results omitting the details. That is, there exist $\Gamma > 0$ and $\lambda \in (0, 1)$ such that:

$$|\tilde{y}_{\text{err}}(t+1)| \leq \Gamma \lambda^{t+1} |\tilde{y}_{\text{err}}(0)| + \Gamma \frac{1}{1-\lambda} \|D_{\text{err}}\| \|u\|.$$

Since the state dynamics matrix for equations (4.10) and (4.9) are the same:

$$|\tilde{x}(t+1)| \leq \Gamma \lambda^{t+1} |\tilde{x}(0)| + \Gamma_2 (\|Q\| |x_1| + \|D_{\text{err}}\| \|u\|),$$

where $\Gamma_2 = \Gamma \frac{1}{1-\lambda}$. Note that $\|Q\| \leq \sqrt{n-1}$ and that x_1 is constant with time.

The statement (iii) is then proved. ■

For algorithms in \mathcal{C}_2 , the network behaves like a low pass filter with respect to the input $x_1(t)$. Lemma 4.2 shows that, after a transient due to the initial condition, the trajectories $x_i(t)$ (for $i \neq 1$) will stay close to $x_1(t)$. If $x_1(t)$ changes slowly with time, then each other state $x_i(t)$ will track that value but with non-vanishing tracking error.

4.5 Summary

In this chapter we have established some ISS properties of linear consensus algorithms with inputs and outputs. We have shown that, in the presence of noise, the disagreement is bounded but the consensus value, in general, does not remain bounded since the network behaves like an integrator. In applications for which consensus value boundness is crucial and the network topology is not known, a linear algorithm cannot be implemented. On the other hand, for some nonlinear update laws (i.e., as in the Kuramoto coupled oscillators) it can be proved that also the consensus value is bounded. This will be subject of further investigation.

Chapter 5

Synchronization of N -Beads on a Ring by Feedback Control

5.1 Introduction

Consider N motion-enabled agents that can speed up and slow down, and also communicate when in proximity of each other. If the N agents control their motion to simulate N beads sliding on a frictionless ring, we know that their dynamics is very rich. In fact, in [9], the authors study extensively the case of $N = 3$ and prove the existence of periodic as well as chaotic orbits. The authors also describe how to use the three-bead system dynamics for a random number generator algorithm which is computationally efficient. We therefore pose the question: can N intelligent beads, capable of controlling their motion, reach a periodic orbit and get in sync? In other words, can each bead sweep a sector of the ring and impact with the neighboring beads always at the boundaries of

the sector? We show that synchronization can be indeed achieved by a simple feedback law. We present an algorithm which requires only occasional communication – two beads exchange information only when they impact – and, using of the theory of discrete-time consensus algorithms, we prove its correctness.

Relevant to this chapter is the reference [7], which presents a synchronization algorithm for cooperative surveillance of a forest fire using a team of unmanned aerial vehicles. A preliminary version of this work appeared in [39].

The contributions of this chapter can be summarized as follows. We design a distributed algorithm that allows a collection of beads to reach synchronization. The definitions of synchronization for both the case of an even and odd number of beads is given. The beads can be deployed with arbitrary initial positions and speeds. At the desired steady state, every bead sweeps a sector of equal length, and neighboring beads meet always at the same point. If N is even the beads will all travel at the same speed, while if N is odd the beads will travel at the same average speed. Two beads exchange information only when they impact. We prove a local convergence result – the agents will reach the desired steady state – under some assumptions.

Extensive simulations show that synchronization is reached in general, even when the assumptions are not satisfied. In [7] pairs of agents have to be released at the same point, sequentially, and with the same speed. In contrast, in our algorithm the number of agents can be odd, the agents can be released at arbitrary positions, with arbitrary speeds and directions. For the case of N even we only require half of the agents to move clockwise direction and the rest to move in the counterclockwise direction.

Notation

On the torus \mathbb{T} , by convention, let us define positions as angles measured counterclockwise from the positive horizontal axis. The *counterclockwise distance between two angles* $\text{dist}_{\text{cc}}: \mathbb{T} \times \mathbb{T} \rightarrow [0, 2\pi)$ is the path length from an angle to the other traveling counterclockwise. Specifically, if $x, y \in \mathbb{T}$, then $\text{dist}_{\text{cc}}(x, y) = (y - x) \bmod 2\pi$, where $x \bmod 2\pi$ is the remainder of the division of x by 2π . We denote by $\mathbf{1} \in \mathbb{R}^{N \times 1}$ the column vector with entries all equal to 1.

5.2 Model and problem statement

In this section we describe a synchronized collection of beads moving on a circle and our model of robotic agents.

Definition 5.1 (Balanced synchronization) *Consider a collection of N beads moving on a ring. The collection of beads is balanced synchronized with period T , if (i) any two neighboring beads impact always at the same point, (ii) the time interval between two consecutive impacts, involving the same beads, has length T , and (iii) all the beads impact simultaneously. In other words, in a synchronized collection, each bead sweeps an arc of length $2\pi/N$ at constant speed $2\frac{2\pi}{NT}$.*

An example of a collection of four beads in sync is shown in Figure 5.1.

If N is odd, synchronization, as defined in Definition 5.1, cannot be reached. Therefore we give a more general definition of synchronization reachable also by an odd number of beads.

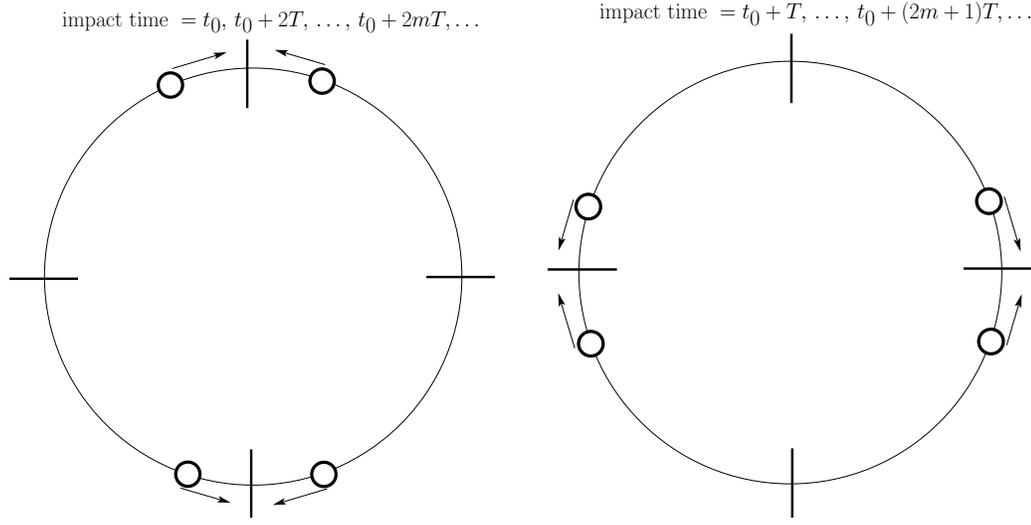


Figure 5.1. The figure shows a collection of four beads which are synchronized.

Definition 5.2 (Unbalanced synchronization) *Consider a collection of N beads moving on a ring. The collection of beads is unbalanced synchronized with period T , if (i) any two beads impact always at the same point and (ii) the time interval between two consecutive impacts, involving the same beads, has length T . In other words, in an unbalanced synchronized collection, each bead sweeps an arc of length $2\pi/N$ at average speed $2\frac{2\pi}{NT}$.*

In this chapter we propose and analyze a distributed algorithm that will steer a collection of “intelligent beads”, i.e., mobile robots, to be synchronized according to either Definition 5.1 or Definition 5.2. The model of agent we consider is described as follows. We assume a collection of N agents moves on the torus \mathbb{T} . Let $\theta_i(t) \in \mathbb{T}$, $i \in \{1, \dots, N\}$ be the agents’ positions at time $t \geq 0$, ordered in counterclockwise direction, at let $\dot{\theta}_i(t)$ be their velocity. Each agent senses its position on the circle. Motivated by surveillance application, we consider agents equipped with short-range communication device; for simplicity and elegance we

assume they communicate only when they impact. We use the identifications $0 \equiv N$ and $N + 1 \equiv 1$.

5.3 Synchronization algorithm

In this section we describe an algorithm that allows the collection of agents to achieve synchronization. We begin by defining all variables that each agent maintains in memory and we later state how these variables are updated as time evolves and “communication impacts” take place.

Variables in memory

Let us define $d_i \in \{-1, +1\}$ to be the direction of motion of the i -th bead, and let the counterclockwise direction of motion be positive. Let $\nu_i > 0$ be the i -th bead’s nominal speed.

Definition 5.3 (Impacts classification) *If at time t it holds $\theta_i(t) = \theta_{i+1}(t)$, then we say that an impact has occurred between beads i and $i + 1$. If $d_i(t) = d_{i+1}(t)$, then the impact is said to be of head-tail type, otherwise it is said to be of head-head type.*

Next, we associate to each bead its desired sweeping arc, i.e., an arc of the circle that eventually each bead would sweep if balanced synchrony, as defined in 5.1, is reached.

Definition 5.4 (Desired sweeping arc) *Let $[L_i(t), U_i(t)] \subset \mathbb{T}$ be the desired sweeping arc of bead i with $L_i(t) \in \mathbb{T}$ and $U_i(t) \in \mathbb{T}$ as its clockwise and*

counterclockwise boundary. Then $[L_i(t), U_i(t)] = \{\theta \in \mathbb{T} \mid \text{dist}_{\text{cc}}(L_i(t), \theta) \leq \text{dist}_{\text{cc}}(L_i(t), U_i(t))\}$. Let $C_i(t)$ be the center of the desired sweeping arc defined by $C_i(t) = L_i(t) + \frac{1}{2} \text{dist}_{\text{cc}}(L_i(t), U_i(t))$.

Note that nothing is assumed on the length of the desired sweeping arcs.

In summary, we denote by $x_i(t)$ the logic state that bead i maintains in its memory:

$$x_i(t) := (\nu_i(t), d_i(t), L_i(t), U_i(t), \text{sdir}_i(t)),$$

where we let $\text{sdir}(t)$ be a flag that can assume values ± 1 . We let (θ_i, x_i) denote the state of each bead.

Regarding initialization, we will assume that $\nu_i > 0$, $d_i \in \{+1, -1\}$, $L_i(0) = U_i(0) = \theta_i(0)$, and $\text{sdir}_i(0) = d_i(0)$.

Rules

This concludes our description of the agents' memory. Next we define the algorithm. At all time $t \geq 0$, each bead checks whether it is traveling inside its desired sweeping arc, or outside the sweeping arc while moving away from it or towards it. Each bead will then travel at nominal speed $\nu_i(t)$ when inside its desired sweeping arc, it will slow down when moving away from it, and speeding up while moving towards it. Formally,

$$\dot{\theta}_i(t) = \begin{cases} d_i(t)\nu_i(t), & \text{if } \theta_i(t) \in [L_i(t), U_i(t)], \\ fd_i(t)\nu_i(t), & \text{if } \theta_i(t) \notin [L_i(t), U_i(t)] \text{ and } d_i(t) = \text{sdir}_i(t), \\ hd_i(t)\nu_i(t), & \text{if } \theta_i(t) \notin [L_i(t), U_i(t)] \text{ and } d_i(t) = -\text{sdir}_i(t), \end{cases}$$

where $f \in]0.5, 1[$ and $h = \frac{f}{2f-1} > 1$.

The logic state for bead i changes only when one of the following events occurs: (1) an impact takes place between with either bead $i - 1$ or i , and (2) bead i crosses either L_i or U_i while leaving its desired sweeping arc.

Event 1. If at time t an impact occurs for bead i with either bead $i + 1$ or $i - 1$, then two events will take place: (1) both beads will exchange through communication their logic state, and (2) each bead will update its memory. Bead i will update its nominal speed and direction by:

$$\nu_i(t^+) = \frac{\nu_i(t) + \nu_{i+1}(t)}{2}; \quad (5.1)$$

$$d_i(t^+) = \begin{cases} -d_i(t), & \text{if the impact is "head-head type",} \\ d_i(t), & \text{otherwise,} \end{cases} \quad (5.2)$$

where the upper-script $+$ indicates the value of the state variables right after the impact. Furthermore, bead i updates the boundary of its sweeping arc by

$$L_i(t^+) = \begin{cases} C_i(t) - \frac{\text{dist}_{\text{cc}}(C_{i-1}(t), C_i(t))}{2}, & \text{if the impact occurs with } i - 1, \\ L_i(t), & \text{otherwise,} \end{cases} \quad (5.3)$$

and

$$U_i(t^+) = \begin{cases} C_i(t) + \frac{\text{dist}_{\text{cc}}(C_i(t), C_{i+1}(t))}{2}, & \text{if the impact occurs with } i + 1, \\ U_i(t), & \text{otherwise,} \end{cases} \quad (5.4)$$

where the center of the j th desired sweeping arc is computed by $C_j(t) = L_j(t) + \text{dist}_{\text{cc}}(L_j(t), U_j(t))/2$, for all j . Note that after any impact between beads i and $i - 1$ equations (5.3) and (5.4) imply that $L_{i-1}(t^+) = U_i(t^+)$, simply because they are defined as the midpoint in the arc from $C_{i-1}(t)$ to $C_i(t)$. The flag sdir_i does not change its value, i.e.:

$$\text{sdir}_i(t^+) = \text{sdir}_i(t).$$

Event 2. The memory of each bead i will be update also when it reaches either $L_i(t)$ or $U_i(t)$, and it crosses it while leaving the desired sweeping arc. While the nominal speed ν_i , the direction d_i and the boundary of the desired sweeping arc L_i and U_i do not change:

$$\begin{aligned}\nu_i(t^+) &= \nu_i(t), \\ d_i(t^+) &= d_i(t), \\ L_i(t^+) &= L_i(t), \\ U_i(t^+) &= U_i(t),\end{aligned}$$

the flag $sdir_i$ is updated as follows:

$$sdir_i(t^+) = d_i(t),$$

where the upper-script $+$ indicates the value of the memory right after bead i crosses the boundary of its desired sweeping arc.

5.4 Preliminary results

In this section we prove some preliminary results before we can prove the correctness of the SYNCHRONIZATION ALGORITHM. We begin with an important characterization of initial states.

Definition 5.5 (Admissible, balanced and unbalanced configurations) *A state*

$\{(\theta_i(0), x_i(0))\}_{i \in \{1, \dots, N\}}$ *is*

1. *admissible if, for all $i, j \in \{1, \dots, N\}$, $j \neq i$,*

$$\nu_i(0) > 0, \quad \text{and} \quad \theta_i(0) \neq \theta_j(0);$$

2. balanced if it is admissible and if N is even and $\sum_{i=1}^N d_i(0) = 0$, that is, $N/2$ beads are moving clockwise and $N/2$ are moving counterclockwise; and
3. D -unbalanced, for $D \in \{-N, \dots, N\} \setminus \{0\}$, if it is admissible and if $\sum_{i=1}^N d_i(0) = D$.

The set of admissible configurations, balanced configurations, and D -unbalanced configurations are denoted by \mathcal{A} , $\mathcal{A}_{0\text{-bal}}$, and $\mathcal{A}_{D\text{-unbal}}$, respectively.

Next, we construct an undirected graph $\mathcal{G}(t)$ with vertex set $\{1, \dots, N\}$ and edge from i to $i + 1$ if the beads i and $i + 1$ collide at time t .

Proposition 5.1 (Uniform connectivity) *Along the trajectories of the SYNCHRONIZATION ALGORITHM, with $\{(\theta_i(0), x_i(0))\}_{i \in \{1, \dots, N\}} \in \mathcal{A}$, for all $t_0 \geq 0$ the graph $\bigcup_{t \in [t_0, t_0 + \frac{2\pi}{f\nu_{\min}}]} \mathcal{G}(t)$ is connected.*

The proof of Proposition 5.1 builds up on the following facts.

Lemma 5.1 (Properties) *Along the trajectories of the SYNCHRONIZATION ALGORITHM, with $\{(\theta_i(0), x_i(0))\}_{i \in \{1, \dots, N\}} \in \mathcal{A}$:*

1. $\sum_{i=1}^n d_i(t)$ is constant,
2. any two desired sweeping arcs are disjoint sets or at most share a boundary point, furthermore their label index increases in the counterclockwise direction, i.e., $L_{i+1}(t) = U_i(t)$,
3. the order of the beads is preserved, i.e., for all $i, j \in \{1, \dots, N\}$, $t \geq 0$, and for $j \neq i$, $\text{dist}_{\text{cc}}(\theta_{i-1}(t), \theta_i(t)) \leq \text{dist}_{\text{cc}}(\theta_{i-1}(t), \theta_{i+1}(t))$ and

$\text{dist}_{\text{cc}}(\theta_{i-1}(t), \theta_j(t)) \geq \text{dist}_{\text{cc}}(\theta_{i-1}(t), \theta_{i+1}(t))$. Therefore, a bead i can be involved only in impacts its immediate neighbors $i - 1$ and $i + 1$.

Proof. We first prove 1. Let $\sum_{i=1}^n d_i(0) = D$. The only instants in which $\sum_{i=1}^n d_i(t)$ can change is when an impact occurs, as in equation (5.2). If the impact is of “head-tail type” the directions of both the beads involved do not change. On the other hand, if the impact is of “head-head type”, the directions of the beads involved are just swapped, therefore $\sum_{i=1}^n d_i(t) = D$ for any $t \geq 0$.

We now prove 2. For convenience let $\mathcal{D}_i = [L_i(t), U_i(t)]$. Part 2 is simply proved by noticing that, to initialize the algorithm, $\mathcal{D}_i(0) = L_i(0) = U_i(0) = \theta_i(0)$, and $\theta_i(0)$ are ordered along the counterclockwise direction. The desired sweeping arc \mathcal{D}_i is updated only when the bead i is involved in an impact and, by equations (5.3) and (5.4). It is elementary to show that the update equations for L_i and U_i will force $U_i(t^+) = L_{i+1}(t^+)$ and $L_i(t^+) = U_{i-1}(t^+)$. This clearly implies that the order of the desired sweeping arcs is never changed and that any two desired sweeping arcs can at most share a boundary.

We finally prove 3. The order of the beads can change only as a consequence of an impact. We will see that even after an impact the order of the beads is preserved. If beads i and $i + 1$ are involved in an impact of “head-head type”, after the impacts both beads will change their direction so clearly $\text{dist}_{\text{cc}}(\theta_{i-1}(t + s), \theta_i(t + s)) \leq \text{dist}_{\text{cc}}(\theta_{i-1}(t + s), \theta_{i+1}(t + s))$, with $0 \leq s < \bar{s}$ and $t + \bar{s}$ is the time at which i will impact again. If the impact is of “head-tail type” the directions of the two beads will not change, but their nominal velocities $\nu_i(t^+)$ and $\nu_{i+1}(t^+)$ will be equal because of equation (5.1). The impact can occur in $\mathcal{D}_i(t)$, or in $\mathcal{D}_{i+1}(t)$ or in neither, see Figure 5.2. If the impact occurs in $\mathcal{D}_i(t)$ and

$d_i(t) = d_{i+1}(t) = +1$, after the impact $u_i(t^+) = \nu_i(t^+)$ while $u_{i+1}(t^+) = h\nu_{i+1}(t^+)$. In fact, because of part 2, $i + 1$ is moving towards its desired sweeping arc. If the impact occurs in $\mathcal{D}_i(t)$ and $d_i(t) = d_{i+1}(t) = -1$ after the impact $u_i(t^+) = -\nu_i(t^+)$ and $u_{i+1}(t^+) = -f\nu_{i+1}(t^+)$ because $i + 1$ is moving away from its desired sweeping arc, again because of part 2. Recalling that $f < 1$ and $h > 1$ we have that, in both cases, $\text{dist}_{\text{cc}}(\theta_{i-1}(t+s), \theta_i(t+s)) \leq \text{dist}_{\text{cc}}(\theta_{i-1}(t+s), \theta_{i+1}(t+s))$ for any time $0 \leq s < \bar{s}$. An analogous reasoning will conclude that this holds also if the impact occurs in $\mathcal{D}_{i+1}(t)$. Now, if the impact occurs in neither $\mathcal{D}_i(t)$ nor $\mathcal{D}_{i+1}(t)$, the beads are both moving either towards or away their desired sweeping arcs therefore $u_i(t^+) = u_{i+1}(t^+) = h\nu_i(t^+)$ or $u_i(t^+) = u_{i+1}(t^+) = f\nu_i(t^+)$. Again $\text{dist}_{\text{cc}}(\theta_{i-1}(t+s), \theta_i(t+s)) \leq \text{dist}_{\text{cc}}(\theta_{i-1}(t+s), \theta_{i+1}(t+s))$ for any $0 \leq s < \bar{s}$. ■

Lemma 5.2 (Impacts in bounded interval) *Let $\nu_{\min} = \min_{i \in \{1, \dots, N\}} \nu_i(0)$.*

Along the trajectories of the SYNCHRONIZATION ALGORITHM, with

$\{(\theta_i(0), x_i(0))\}_{i \in \{1, \dots, N\}} \in \mathcal{A}$, for all $i \in \{1, \dots, N\}$ and for all $t_0 > 0$, bead i will impact at least once with both its neighbors $i - 1$ and $i + 1$ across the interval $[t_0, t_0 + \frac{2\pi}{f\nu_{\min}}]$.

Proof. Note that $\min_{i \in \{1, \dots, N\}} \nu_i(t) \geq \min_{i \in \{1, \dots, N\}} \nu_i(0) = \nu_{\min}$ because of equation (5.1). Therefore for any $t > 0$ the lowest possible speed at which a bead can travel is $f\nu_{\min}$. We first show that at most after $\frac{\pi}{f\nu_{\min}}$ any bead will have a “head-head type” impact one of its neighbors.

First, any bead i can only impact neighbors $i + 1$ and $i - 1$ because of Lemma 5.1, part 3. The necessary time for two beads $i, i + 1$ to impact depends on their positions, the directions of motion and the speeds they are traveling with.

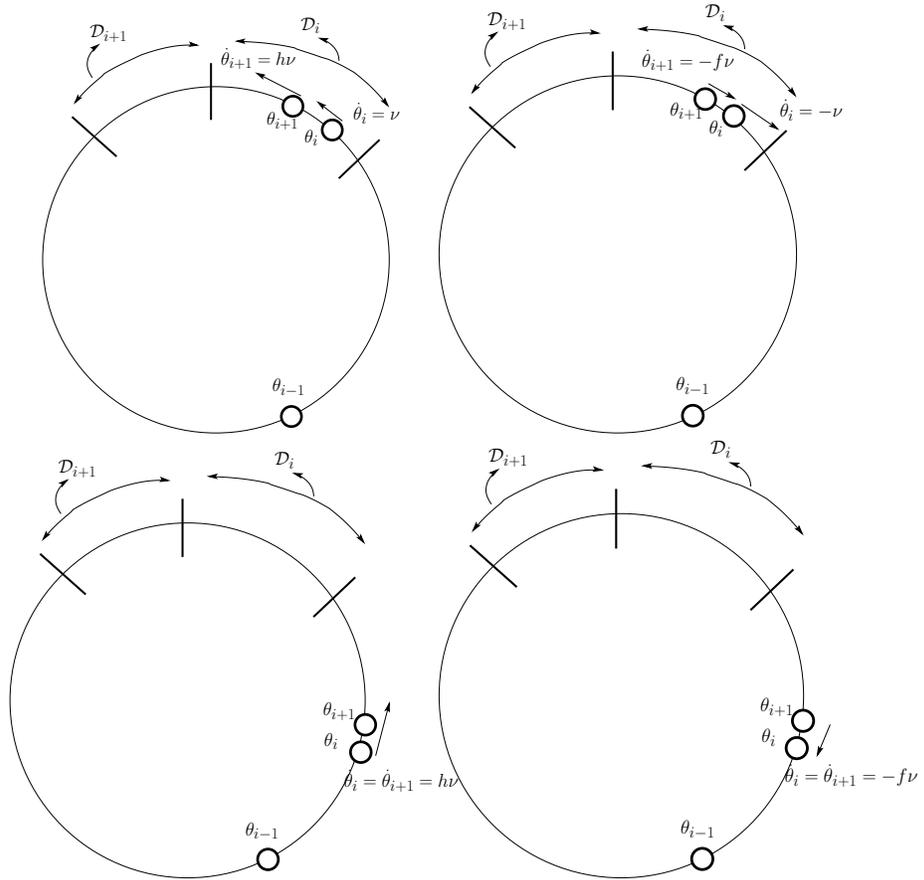


Figure 5.2. This figure shows that, regardless from where and with which velocities beads i and $i + 1$ impact, the order of the beads is preserved. The velocities in the figure are the velocities after the impact. The speed ν is just the average value of ν_i and ν_{i+1} before the impact.

In the worst possible case at a time $t = t_0$ all the beads are clustered in a small arc of \mathbb{T} of length ϵ , with i and $i + 1$ at the opposite ends of the arc (i.e., $\text{dist}_{\text{cc}}(\theta_{i+1}(t_0), \theta_i(t_0)) = \epsilon$), $d_i(t_0) = d_{i+1}(t_0)$, and the speeds have the smallest possible value $|u_i(t_0)| = |u_{i+1}(t_0)| = f\nu_{\min}$.

Let us suppose $d_i(t_0) = d_{i+1}(t_0) = +1$. That is, $i + 1$ is moving towards the cluster of beads and i is moving away from it. Because of Lemma 5.1, part 1, we have that $-\sum_{i=1}^n d_i(t_0) = D < N$ and this implies that $i + 1$ can travel at most for $\frac{\epsilon}{2f\nu_{\min}}$ before having a “head-head type” impact. So at $t_1 \leq t_0 + \frac{\epsilon}{2f\nu_{\min}}$, $d_{i+1}(t_1) = -1$, and $\text{dist}_{\text{cc}}(\theta_{i+1}(t_1), \theta_i(t_1)) \geq \epsilon$. This is true because by assumption $|u_i(t_0)| = |u_{i+1}(t_0)|$ and i could have had a “head-tail type” impact with $i - 1$ so that $|u_i(t_1)| \geq f\nu_{\min}$. Now, suppose that even after the impact $|u_{i+1}(t_1)| = f\nu_{\min}$, then beads i and $i + 1$ are moving towards each other and $\text{dist}_{\text{cc}}(\theta_i(t_1), \theta_{i+1}(t_1)) \leq 2\pi - \epsilon$. They will then meet at time $t_2 \leq t_1 + \frac{2\pi - \epsilon}{2f\nu_{\min}} \leq t_0 + \frac{\epsilon}{2f\nu_{\min}} + \frac{2\pi - \epsilon}{2f\nu_{\min}} = t_0 + \frac{\pi}{f\nu_{\min}}$.

After the impact with $i + 1$, $d_i(t_2) = -1$ and, therefore, in its next “head-head type” impact bead i will meet $i - 1$. Following the same reasoning, we have that at most after $\frac{\pi}{f\nu_{\min}}$ the two beads i and $i - 1$ will meet. Hence across the interval $[t_0, t_0 + \frac{2\pi}{f\nu_{\min}}]$ any bead will impact at least once with both its neighbors. ■

Proof.[of Proposition 5.1] Because of Lemma 5.2, for all i and for all t_0 there exist t_1 and $t_2 \in [t_0, t_0 + \frac{2\pi}{f\nu_{\min}}]$ such that $\mathcal{G}(t_1)$ and $\mathcal{G}(t_2)$ have respectively an edge between vertices i and $i + 1$ and between vertices i and $i - 1$. Then, clearly the graph $\bigcup_{t \in [t_0, t_0 + \frac{2\pi}{f\nu_{\min}}]} \mathcal{G}(t)$ is connected. ■

5.5 Convergence analysis

In this first part of this section we prove that the nominal speeds ν_i will asymptotically be equal to the average of their values, and that the desired sweeping arc will asymptotically have length $2\pi/N$. In the second and third part of this section we show that SYNCHRONIZATION ALGORITHM enables the beads to reach balanced synchrony if N is even and unbalanced synchrony if N is odd, respectively.

5.5.1 Convergence of nominal speed and desired sweeping arc

We start by proving that the nominal speeds ν_i will be all equal to the average of their values.

Lemma 5.3 (Speed convergence) *Let $\nu(t) = [\nu_1(t), \dots, \nu_n(t)]^T \in \mathbb{R}^{N \times 1}$.*

Along the trajectories of the SYNCHRONIZATION ALGORITHM, with

$\{(\theta_i(0), x_i(0))\}_{i \in \{1, \dots, N\}} \in \mathcal{A}$:

$$\lim_{t \rightarrow +\infty} \left\| \nu(t) - \frac{\mathbf{1}^T \nu(0)}{N} \mathbf{1} \right\| = 0.$$

Proof. For all $i \in \{1, \dots, N\}$, define $A_i \in \mathbb{R}^{N \times N}$ by:

$$[A_i]_{lm} = \begin{cases} \frac{1}{2}, & \text{if } l = m = i \text{ or } l = m = i + 1, \\ \frac{1}{2}, & \text{if } (l, m) \in \{(i, i + 1), (i + 1, i)\}, \\ \delta_{lm}, & \text{otherwise.} \end{cases}$$

Because of equation (5.1), if at time t an impact between i and $i + 1$ occurs, then:

$$\nu(t^+) = A_i \nu(t).$$

Therefore the dynamics of $\nu(t)$ is just the average consensus dynamics with matrices A_i and, because of Proposition 5.1, the consensus is asymptotically reached (see [30]). Clearly, because the matrices A_i are doubly stochastic, the consensus value is $\frac{1}{N} \sum_{i=1}^N \nu_i(0)$. \blacksquare

We now prove that the desired sweeping arc will asymptotically have length $2\pi/N$.

Lemma 5.4 (Desired sweeping arc convergence) *Let*

$\ell_i(t) = \text{dist}_{cc}(L_i(t), U_i(t))$ *be the length of the desired sweeping arc* $[L_i(t), U_i(t)]$ *for* $i \in \{1, \dots, N\}$, *and* $\ell(t) = [\ell_1(t), \dots, \ell_n(t)]^T \in \mathbb{R}^{N \times 1}$. *Along the trajectories of the SYNCHRONIZATION ALGORITHM, with* $\{(\theta_i(0), x_i(0))\}_{i \in \{1, \dots, N\}} \in \mathcal{A}$:

$$\lim_{t \rightarrow +\infty} \|\ell(t) - \frac{2\pi}{N} \mathbf{1}\| = 0.$$

Proof. From equations (5.3) and (5.4) we have that after the impact between i and $i + 1$:

$$\begin{aligned} \ell_i(t^+) &= \frac{3}{4}\ell_i(t) + \frac{1}{4}\ell_{i+1}(t), \\ \ell_{i+1}(t^+) &= \frac{1}{4}\ell_i(t) + \frac{3}{4}\ell_{i+1}(t). \end{aligned}$$

Now, define $B_i \in \mathbb{R}^{N \times N}$ by:

$$[B_i]_{lm} = \begin{cases} \frac{3}{4}, & \text{if } l = m = i \text{ or } l = m = i + 1, \\ \frac{1}{4}, & \text{if } (l, m) \in \{(i, i + 1), (i + 1, i)\}, \\ \delta_{lm}, & \text{otherwise.} \end{cases}$$

Then, if at time $t > t_0 + \frac{2\pi}{f\nu_{\min}}$ an impact between i and $i + 1$ occurs, the dynamics for $\ell(t)$ is simply:

$$\ell(t^+) = B_i \ell(t).$$

We recall that $\nu_{\min} = \min_{i \in \{1, \dots, N\}} \nu_i(0)$, and that at $t_0 + \frac{2\pi}{f\nu_{\min}}$ every impact between two consecutive beads has occurred. Once again, the dynamics of $\ell(t)$ is just the weighted average consensus dynamics with matrices B_i and, because of Proposition 5.1, the consensus is asymptotically reached (see [30]). Since $\sum_{i=1}^n \ell_i(t) = 2\pi$, or equivalently because the matrices B_i are doubly stochastic, we have that $\ell_i(t) \rightarrow \frac{2\pi}{N}$ asymptotically. \blacksquare

We have then proved that asymptotically the nominal velocities $\nu_i(t)$ will be equal to the average of the initial nominal velocities and the lengths of the desired sweeping arcs will asymptotically be equal to $2\pi/N$.

5.5.2 Balanced synchrony

We will now prove that the SYNCHRONIZATION ALGORITHM will steer the collection of beads to be in balanced synchrony for a set of initial conditions contained entirely in $\mathcal{A}_{0\text{-bal}}$.

Theorem 5.1 (Balanced synchrony convergence) *For all $i \in \{1, \dots, N\}$, let $\nu_i(0) = \nu_i(t) = \bar{\nu} > 0$, $L_i(t) = L_i(0)$, $U_i(t) = U_i(0)$, with $L_i(0) = U_{i-1}(0)$, and with $\text{dist}_{cc}(L_i(0), L_{i+1}(0)) = \frac{2\pi}{N}$. Let $d_i(0) = -d_j(0)$ for $j \in \{i-1, i+1\}$. Let $\gamma_i = \text{dist}_{cc}(C_i(0), \theta_i(0))$, $\delta_i = \min\{\gamma_i, 2\pi - \gamma_i\}$, and let $\delta = [\delta_1, \dots, \delta_N]^T \in \mathbb{R}^{N \times 1}$. Let T_i^k be the instant in which bead i passed by the center of its desired sweeping arc for the k -th time and $T^k = [T_1^k, \dots, T_N^k]^T \in \mathbb{R}^{N \times 1}$. If $\|\delta - \frac{\mathbf{1}^T \delta}{N} \mathbf{1}\|$ is sufficiently small, then along the trajectories of the SYNCHRONIZATION ALGORITHM:*

$$\lim_{k \rightarrow +\infty} \left\| T^k - \frac{\mathbf{1}^T T^k}{N} \mathbf{1} \right\| = 0.$$

Proof. Before tackling the proof it is useful to see that both the quantities $\|\delta - \frac{1}{N} \delta \mathbf{1}\|$ and $\|T^k - \frac{1}{N} T^k \mathbf{1}\|$ are measures of the asynchrony of the collection of beads. However, due to the switching nature of the dynamics of the beads, the asymptotic behavior of T^k is simpler to analyze. On the other hand δ is a more suitable quantity to describe the asynchrony at time 0.

For convenience let $\mathcal{D}_i = [L_i(t), U_i(t)]$. Let us suppose that at time t the beads i and $i + 1$, with directions $d_i(t) = -d_{i+1}(t) = +1$, are about to collide. We know that T_i^k and T_{i+1}^k , for some k , are the times at which they passed by the centers of their desired sweeping arcs. If $T_i^k < T_{i+1}^k$, that is bead i is *early* with respect to bead $i + 1$, the impact will occur in \mathcal{D}_{i+1} as shown in Figure 5.3, otherwise it will occur in \mathcal{D}_i . Without loss of generality we suppose that the impact will occur in \mathcal{D}_{i+1} , and precisely at $U_i + \Delta$.

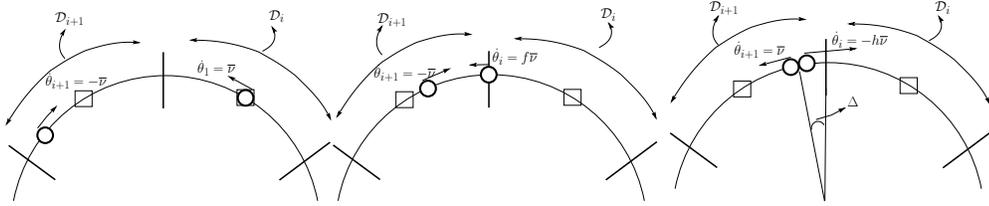


Figure 5.3. This figure shows how the speeds of bead i and $i + 1$ change they are traveling towards each other. Note that bead i is early with respect to bead $i + 1$.

In order to calculate where and when the beads will impact we need to impose that i and $i + 1$ reach simultaneously $U_i + \Delta$:

$$T_i^k + \frac{\pi}{N} \frac{1}{\bar{v}} + \frac{\Delta}{f\bar{v}} = T_{i+1}^k + \frac{\pi}{N} \frac{1}{\bar{v}} - \frac{\Delta}{\bar{v}}. \quad (5.5)$$

This clearly holds because $i + 1$ is traveling in its desired sweeping arc, therefore its speed is simply \bar{v} ; on the other hand bead i initially travels in its arc but

eventually leaves it and therefore its speed becomes $f\bar{v}$. Solving (5.5) for Δ we have:

$$\Delta = \bar{v} \frac{f}{1+f} (T_{i+1}^k - T_i^k). \quad (5.6)$$

After the impact the directions of both beads change because the impact is of “head-head type”, and they both head towards C_i and C_{i+1} , that they will reach at time T_i^{k+1} and T_{i+1}^{k+1} :

$$\begin{aligned} T_i^{k+1} &= T_i^k + \frac{2\pi}{N} \frac{1}{\bar{v}} + \frac{\Delta}{\bar{v}} \left(\frac{1}{f} + \frac{1}{h} \right), \\ T_{i+1}^{k+1} &= T_{i+1}^k + 2 \left(\frac{\pi}{N} - \Delta \right) \frac{1}{\bar{v}}. \end{aligned}$$

Recalling that $h = \frac{f}{2f-1}$ we have:

$$T_i^{k+1} = T_i^k + \frac{2}{\bar{v}} \left(\frac{\pi}{N} + \Delta \right), \quad (5.7)$$

$$T_{i+1}^{k+1} = T_{i+1}^k + \frac{2}{\bar{v}} \left(\frac{\pi}{N} - \Delta \right). \quad (5.8)$$

Substituting (5.6) in (5.7) and in (5.8):

$$\begin{aligned} T_i^{k+1} &= \frac{1-f}{1+f} T_i^k + \frac{2f}{1+f} T_{i+1}^k + \frac{2\pi}{N\bar{v}}, \\ T_{i+1}^{k+1} &= \frac{2f}{1+f} T_i^k + \frac{1-f}{1+f} T_{i+1}^k + \frac{2\pi}{N\bar{v}}. \end{aligned}$$

Note that $0 < \frac{1-f}{1+f} < 1/3$ and $2/3 < \frac{2f}{1+f} < 1$ since $f \in]0.5, 1[$. Now, let us define the matrices C^{even} and $C^{\text{odd}} \in \mathbb{R}^{N \times N}$ by:

$$\begin{aligned} [C^{\text{even}}]_{lm} &= \begin{cases} \frac{1-f}{1+f}, & \text{if } l = m, \\ \frac{2f}{1+f}, & \text{if } (l, m) \in \{(i, i+1), (i+1, i)\}, i \text{ even,} \end{cases} \\ [C^{\text{odd}}]_{lm} &= \begin{cases} \frac{1-f}{1+f}, & \text{if } l = m, \\ \frac{2f}{1+f}, & \text{if } (l, m) \in \{(i, i+1), (i+1, i)\}, i \text{ odd.} \end{cases} \end{aligned}$$

Then, if the first impact after $t = 0$ is between i and $i + 1$, and i is odd the vector T^k evolves as follows:

$$T^{k+1} = \begin{cases} C^{\text{odd}}T^k + \frac{2\pi}{N\nu}\mathbf{1}, & \text{if } k \text{ odd,} \\ C^{\text{even}}T^k + \frac{2\pi}{N\nu}\mathbf{1}, & \text{if } k \text{ even.} \end{cases} \quad (5.9)$$

If the first impact is between i and $i + 1$, and i is even, equation (5.9) is still valid as long as the definitions of C^{odd} and C^{even} are exchanged. In any case, the dynamics of T^k is just the weighted average consensus dynamics with matrices C^{odd} and C^{even} , and, because of Proposition 5.1, the consensus is asymptotically reached (see [30]). ■

Although Theorem 5.1 proves convergence to balanced synchronization only locally, simulations show that indeed the set of initial conditions for which the balanced synchronization is reached is quite large and may be equal to $\mathcal{A}_{0\text{-bal}}$. In the next remark we give some insight.

Remark 5.1 *The SYNCHRONIZATION ALGORITHM leads to a dynamical system that can be seen as a cascade of three dynamical systems: the dynamical systems of the nominal velocities $\nu_i(t)$, the dynamical systems of the desired sweeping arcs $\mathcal{D}_i(t)$, and the dynamical system of the synchrony T_i^k . The dynamical systems of the nominal velocities and of the desired sweeping arcs are independent from each other and independent from the dynamics of the synchrony, furthermore they act as disturbances on the latter. As proved in Lemma 5.3 and Lemma 5.4, $\lim_{t \rightarrow +\infty} \|\nu(t) - \frac{\mathbf{1}^T \nu(t)}{N} \mathbf{1}\| = 0$ and $\lim_{t \rightarrow +\infty} \|\ell(t) - \frac{\mathbf{1}^T \ell(t)}{N} \mathbf{1}\| = 0$ for all initial conditions in \mathcal{A} – the consensus of the nominal speeds and of the lengths of the desired sweeping arcs is guaranteed. Furthermore, since the convergence is uniform and the dynamics are linear the convergence is exponential. For the same reasons also*

the convergence of $\|T^k - \frac{\mathbf{1}^T T^k \mathbf{1}}{N} \mathbf{1}\|$ is exponential. If the inputs $\|\nu(t) - \frac{\mathbf{1}^T \nu(t) \mathbf{1}}{N} \mathbf{1}\|$ and $\|\ell(t) - \frac{\mathbf{1}^T \ell(t) \mathbf{1}}{N} \mathbf{1}\|$ enter linearly in the dynamics of T_i^k , then the local stability properties of the equilibrium $\|T^k - \frac{\mathbf{1}^T T^k \mathbf{1}}{N} \mathbf{1}\| = 0$ are not destroyed. This follows from Input-to-State Stability of exponentially stable systems [23]. If this holds, the restrictive assumptions for Theorem 5.1 are that $\|\delta - \frac{\mathbf{1}^T \delta \mathbf{1}}{N} \mathbf{1}\|$ is sufficiently small and that $d_i(0) = -d_j(0)$ for $j \in \{i-1, i+1\}$, while the assumptions that $\nu_i(0)$ have the same value and that $\text{dist}_{cc}(L_i(0), L_{i+1}(0)) = 2\pi/N$ are not restrictive.

5.5.3 Unbalanced synchrony

We now prove that the SYNCHRONIZATION ALGORITHM will steer the collection of beads to be in unbalanced synchrony for a set of initial conditions contained entirely in $\mathcal{A}_{D\text{-unbal}}$ with $D = \pm 1$, however we first start by proving that there exists an orbit along which the beads can reach unbalanced synchrony.

Theorem 5.2 (Exist. of periodic orbit for 1-unbal.: sufficiency) *Given $D \in \{-1, +1\}$, assume that $\{(\theta_i(0), x_i(0))\}_{i \in \{1, \dots, N\}} \in \mathcal{A}_{D\text{-unbal}}$, $\frac{1}{2} < f < \frac{N}{1+N}$, and that, for $i \in \{1, \dots, N\}$, $\nu_i(t) = \nu_i(0) = \bar{\nu}$, $L_i(t) = L_i(0)$, $U_i(t) = U_i(0)$ with $L_i(0) = U_{i-1}(0)$ and with $\text{dist}_{cc}(L_i(0), L_{i+1}(0)) = \frac{2\pi}{N}$. Then*

1. *there exists a periodic orbit for the SYNCHRONIZATION ALGORITHM in which the beads are in unbalanced synchrony with period $2\frac{2\pi}{N} \frac{1}{\bar{\nu}}$; and*
2. *along this orbit each bead i impacts its neighboring bead $i-1$ at position $L_i(0) + D\delta$, where $\delta = \frac{2\pi}{N^2} \frac{f}{1-f} < \frac{2\pi}{N}$.*

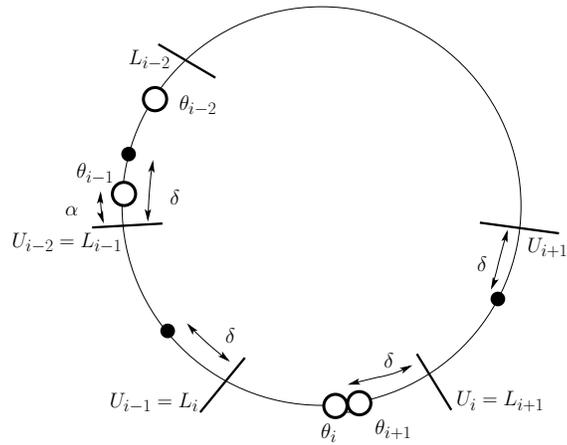


Figure 5.4. This figure shows the periodic orbit described in Theorem 5.2. The white circles are the positions of beads $i - 2$, $i - 1$, i , and $i + 1$ when i and $i + 1$ meet at $U_i - \delta$. The black dots are the locations of the impacts for any two neighboring beads. Note that bead $i - 1$ and $i - 2$ are moving towards each other. Because bead $i - 2$ is in its desired sweeping arc, its speed is \bar{v} while $i - 1$ is moving away from it and therefore its speed is $f\bar{v}$. The same holds for i and $i + 1$ respectively.

Remark 5.2 (Impacts order in 1-unbalanced synchrony) *It is useful to take note of the order in which the impacts happen in a D -unbalanced collection of beads that reached unbalanced synchrony, where $D \in \{-1, +1\}$. As we will see in the proof of Theorem 5.2, if $\sum_{i=1}^N d_i(0) = -1$ and i and $i + 1$ have just met, the next impact will be between $i - 1$ and $i - 2$ and so on until i meets $i + 1$ again and the periodic orbit is complete. More concisely, if the first two beads to impact are i and $i + 1$, then the k -th impact will happen between $(i - 3Dk) \bmod N$ and $(i + 1 - 3Dk) \bmod N$. Therefore if $\sum_{i=1}^N d_i(0) = -1$, then the impacts happen in a counterclockwise fashion, on the other hand, if $\sum_{i=1}^N d_i(0) = +1$, then the impacts happen in a clockwise fashion. Let us illustrate the idea using a the graph $\mathcal{G}(t)$ introduced in Proposition 5.1. We recall that the graph $\mathcal{G}(t)$ has as vertex set $\{1, \dots, N\}$ and edge from i to $i + 1$ if the beads i and $i + 1$ collide at time t . Figure 5.5 shows $\mathcal{G}(t)$ for $t \in [t_{1,2}, t_{1,2} + 2\frac{2\pi}{N}\frac{1}{v}]$ and the time at which the impacts happen for $N = 5$.*

Proof.[of Theorem 5.2] We will prove the theorem by constructing the periodic orbit. Without loss of generality let us suppose that $\sum_{i=1}^N d_i(0) = -1$. Let $t_{i,i+1}$ be the time at which bead i and bead $i + 1$ impact at $U_i(0) - \delta \equiv L_{i+1}(0) - \delta$. Let us suppose that $\theta_{i-1}(t_{i,i+1}) = L_{i-1}(0) - \alpha$ and that $\theta_{i-2}(t_{i,i+1})$ is such that:

$$t_{i-2,i-1} = t_{i,i+1} + \frac{\delta - \alpha}{f\bar{v}}, \quad (5.10)$$

with $\delta < \frac{2\pi}{N}$ and $\alpha < \delta$ (see Figure 5.4). Let $t_0 = t_{1,2}$ then, recalling (5.10) and by symmetry we have:

$$t_{2,3} = t_0 + \frac{N-1}{2} \frac{\delta - \alpha}{f\bar{v}} \quad (5.11)$$

$$t_{N,1} = t_0 + \frac{N+1}{2} \frac{\delta - \alpha}{f\bar{v}}. \quad (5.12)$$

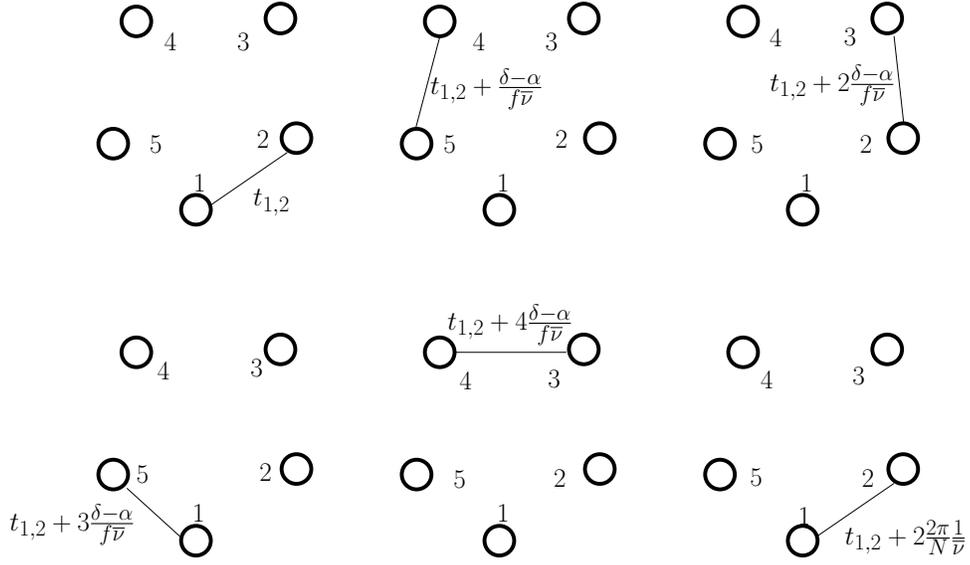


Figure 5.5. This figure illustrates $\mathcal{G}(t)$ for $t \in [t_{1,2}, t_{1,2} + 2\frac{2\pi}{N}\frac{1}{\bar{v}}]$ and the time at which each edge appears for $N = 5$ and $\sum_{i=1}^N d_i(0) = -1$ when unbalanced synchrony is reached.

For beads 1 and 2 to meet again at $U_1(0) - \delta \equiv L_2(0) - \delta$, the following must hold:

$$t_{2,3} + \left(\frac{2\pi}{N} - \delta\right) \frac{1}{\bar{v}} + \frac{\delta}{f\bar{v}} = t_{N,1} + \frac{\delta}{h\bar{v}} + \left(\frac{2\pi}{N} - \delta\right) \frac{1}{\bar{v}}. \quad (5.13)$$

In fact, after impacting with bead 3, bead 2 travels along the arc $[L_2(0), U_2(0) - \delta]$ with velocity $-\bar{v}$ since it is in its desired sweeping arc. After crossing $L_2(0)$, the speed of bead 2 becomes $-f\bar{v}$ because it is moving away from its arc. For bead 1 the dual is true. After impacting with bead N , bead 1 travels along the arc $[L_1(0) - \delta, L_1(0)]$ with velocity $h\bar{v}$ since it is moving towards its desired sweeping arc. After crossing $L_1(0)$, the speed of bead 1 becomes \bar{v} because it is in its arc (see Figure 5.6).

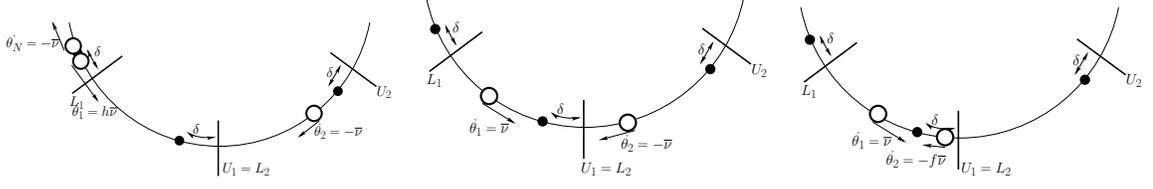


Figure 5.6. This figure shows how the speeds of bead 1 and 2 change as they are traveling towards each other, shortly after bead 1 meets bead N .

Recalling (5.11) and (5.12), we have:

$$t_0 + \frac{N-1}{2} \frac{\delta - \alpha}{f\bar{v}} + \left(\frac{2\pi}{N} - \delta\right) \frac{1}{\bar{v}} + \frac{\delta}{f\bar{v}} = t_0 + \frac{N+1}{2} \frac{\delta - \alpha}{f\bar{v}} + \frac{\delta}{h\bar{v}} + \left(\frac{2\pi}{N} - \delta\right) \frac{1}{\bar{v}},$$

rearranging all the terms:

$$\alpha = \delta(2f - 1). \quad (5.14)$$

In order to be a periodic orbit we need to impose that beads 1 and 2 meet again after a period:

$$t_0 + \frac{N-1}{2} \frac{\delta - \alpha}{f\bar{v}} + \left(\frac{2\pi}{N} - \delta\right) \frac{1}{\bar{v}} + \frac{\delta}{f\bar{v}} = t_0 + 2 \frac{2\pi}{N} \frac{1}{\bar{v}}. \quad (5.15)$$

If we substitute (5.14) in (5.15) and solve for δ we have:

$$\delta = \frac{2\pi}{N^2} \frac{f}{1-f}.$$

Recalling the assumption of f we have:

$$f < \frac{N}{1+N} \Rightarrow \delta = \frac{2\pi}{N^2} \frac{f}{1-f} < \frac{2\pi}{N}.$$

■

It turns out that $f < \frac{N}{1+N}$ is not only sufficient but also necessary for the existence of a periodic orbit along the trajectories of the SYNCHRONIZATION ALGORITHM and $\{(\theta_i(0), x_i(0))\}_{i \in \{1, \dots, N\}} \in \mathcal{A}_{D-\text{unbal}}$ with $D \in \{-1, +1\}$.

Theorem 5.3 (Exist. of periodic orbit for 1-unbal.: necessity) *Given $D \in \{-1, +1\}$, assume that $\{(\theta_i(0), x_i(0))\}_{i \in \{1, \dots, N\}} \in \mathcal{A}_{D\text{-unbal}}$, and that, for $i \in \{1, \dots, N\}$, $\nu_i(t) = \nu_i(0) = \bar{\nu}$, $L_i(t) = L_i(0)$, $U_i(t) = U_i(0)$ with $L_i(0) = U_{i-1}(0)$ and with $\text{dist}_{\text{cc}}(L_i(0), L_{i+1}(0)) = \frac{2\pi}{N}$. If along the trajectories of the SYNCHRONIZATION ALGORITHM the unbalanced synchrony is reached, that is, beads i and $i - 1$ always meet at $L_i(t) + D\delta$ with $\delta < \frac{2\pi}{N}$ and the period of the orbit is $2\frac{2\pi}{N}\frac{1}{\bar{\nu}}$, then $f < \frac{N}{1+N}$.*

Proof. Let us assume, with no loss of generality, that $\sum_{i=1}^N d_i(0) = -1$. Let t^+ be the time spent by each bead traveling along the positive direction, and t^- be the time spent by each bead traveling along the negative direction in a period of the periodic orbit. In other words, if $\delta < \frac{2\pi}{N}$, then $t^- = (\frac{2\pi}{N} - \delta)\frac{1}{\bar{\nu}} + \frac{\delta}{f\bar{\nu}}$, and $t^+ = \frac{\delta}{h\bar{\nu}} + (\frac{2\pi}{N} - \delta)\frac{1}{\bar{\nu}}$, as in (5.13). Clearly $t^- + t^+ = 2\frac{2\pi}{N}\frac{1}{\bar{\nu}}$, which is the period of the orbit, and $t^- > t^+$, that is each bead spends more time traveling along the negative direction than along the positive. Every instant of time only one bead is unbalanced and $t^- - t^+$ is the time each bead is unbalanced during a period. By symmetry we can then conclude that $N(t^- - t^+)$ must be equal to a period:

$$2\frac{2\pi}{N}\frac{1}{\bar{\nu}} = N(t^- - t^+). \quad (5.16)$$

Recalling the expressions for t^- and t^+ , we have:

$$2\frac{2\pi}{N}\frac{1}{\bar{\nu}} = N2\frac{\delta}{\bar{\nu}}\frac{f}{1-f},$$

and solving for δ

$$\delta = \frac{2\pi}{N^2}\frac{f}{1-f}.$$

By assumption $\delta < \frac{2\pi}{N}$, therefore:

$$\delta = \frac{2\pi}{N^2}\frac{f}{1-f} < \frac{2\pi}{N} \Rightarrow f < \frac{N}{1+N}.$$

■

A natural question to ask is if there exists a periodic orbit for the SYNCHRONIZATION ALGORITHM when $\{(\theta_i(0), x_i(0))\}_{i \in \{1, \dots, N\}} \in \mathcal{A}_{D\text{-unbal}}$ and $|D| > 1$. To answer this question, we will extend the result of Theorem 5.3 to the more general case of D -unbalanced collections of beads.

Theorem 5.4 (Existence of a periodic orbit: necessity) *Let*

$\{(\theta_i(0), x_i(0))\}_{i \in \{1, \dots, N\}} \in \mathcal{A}_{D\text{-unbal}}$ and $|D| > 1$. If along the trajectories of the SYNCHRONIZATION ALGORITHM the unbalanced synchrony is reached and bead i meets bead $i - 1$ at location $L_i(t) + \frac{D}{|D|}\delta$ with $\delta < \frac{2\pi}{N}$, then $f < \frac{N/|D|}{1+N/|D|}$.

Proof. The proof parallels the one of Theorem 5.3. Without loss of generality let us assume $\sum_{i=1}^N d_i(t) = D < -1$. Every instant of time $|D|$ beads are unbalanced and $t^- - t^+$ is the time each bead is unbalanced during a periodic orbit. By symmetry we can then conclude that $N \frac{(t^- - t^+)}{|D|}$ must be equal to a period, therefore equation (5.16) becomes:

$$2 \frac{2\pi}{N} \frac{1}{\bar{\nu}} = N \frac{(t^- - t^+)}{|D|},$$

where $t^- - t^+ = 2 \frac{\delta}{\bar{\nu}} \frac{f}{1-f}$. Solving for δ we have:

$$\delta = |D| \frac{2\pi}{N^2} \frac{f}{1-f}.$$

Imposing the constraint $\delta < \frac{2\pi}{N}$ we can calculate the necessary condition for the existence of the periodic orbit in a D -unbalanced collection of beads:

$$f < \frac{N/|D|}{1 + N/|D|}.$$

Note that the higher the ratio $|D|/N$ is, the smaller f needs to be so that each bead spends enough time outside of its desired sweeping arc $[L_i(t), U_i(t)]$ but it does not get too far from it. ■

We will now prove that if $\{(\theta_i(0), x_i(0))\}_{i \in \{1, \dots, N\}} \in \mathcal{A}_{D\text{-unbal}}$ with $D \in \{-1, +1\}$, and the initial condition is such that the collection of beads is close to unbalanced synchrony, then the SYNCHRONIZATION ALGORITHM will asymptotically steer the collection of beads to unbalanced synchrony. In particular we will prove that the interval between two consecutive times each bead passes by a point while moving in the same direction asymptotically approaches $2\frac{2\pi}{N}\frac{1}{\bar{\nu}}$, which is the period of the periodic orbit. This is just a consequence of the definition of unbalanced synchrony.

Theorem 5.5 (1-unbalanced synchrony convergence) *Given*

$D \in \{-1, +1\}$, assume that $\{(\theta_i(0), x_i(0))\}_{i \in \{1, \dots, N\}} \in \mathcal{A}_{D\text{-unbal}}$, and that, for $i \in \{1, \dots, N\}$, $\nu_i(0) = \nu_i(t) = \bar{\nu} > 0$, $L_i(t) = L_i(0)$, $U_i(t) = U_i(0)$, with $L_i(0) = U_{i-1}(0)$, and with $\text{dist}_{\text{cc}}(L_i(0), L_{i+1}(0)) = \frac{2\pi}{N}$. Let $\delta = \frac{2\pi}{N^2} \frac{f}{1-f} < \frac{2\pi}{N}$, and \tilde{C}_i be the center of the counterclockwise arc $[L_i(0) + D\delta, U_i(0) + D\delta]$ for all $i \in \{1, \dots, N\}$, i.e., $\tilde{C}_i(t) = L_i(t) + D\delta + \frac{\pi}{N}$. Let T_i^k be the instant in which bead i passed by \tilde{C}_i for the k -th time and $T^k = [T_1^k, \dots, T_N^k]^T \in \mathbb{R}^{N \times 1}$. Let us suppose that the collection of beads is sufficiently close to be unbalanced synchronized, that is (i) after any impact each bead i passes by \tilde{C}_i before impacting again, (ii) the impacts occurs according to the impact sequence described in Remark 5.2. Then, along the trajectories of the SYNCHRONIZATION ALGORITHM:

$$\lim_{k \rightarrow +\infty} T^{2k} - T^{2(k-1)} = \mathbf{1} \frac{2}{\bar{\nu}} \frac{2\pi}{N},$$

that is, the collection of beads asymptotically reaches unbalanced synchrony.

Proof. *Case (i)* Let us suppose $\delta < \frac{\pi}{N}$, and $\sum_{i=1}^N d_i(0) = -1$. Let us suppose that bead $i-1$ and bead i are moving towards each other and let T_{i-1}^k and T_i^k be the last time they passed by \tilde{C}_{i-1} and \tilde{C}_i with directions $d_{i-1} = +1$ and $d_i = -1$.

If the two beads are not in unbalanced sync they will not meet at $U_{i-1} - \delta$ but at $U_{i-1} - \delta - \Delta$, as shown in Figure 5.7. In order to calculate where and when

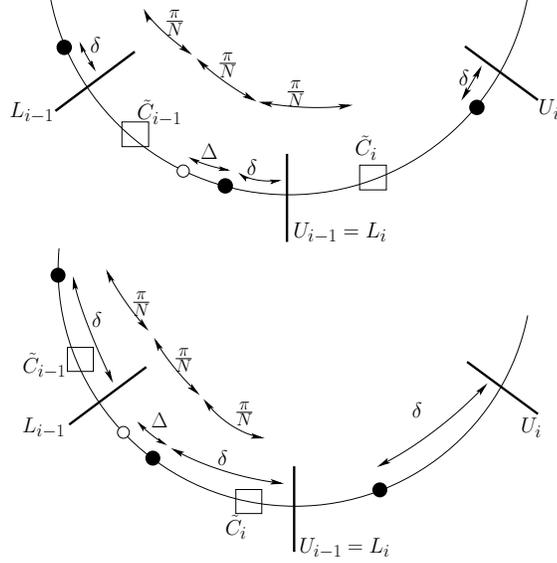


Figure 5.7. From top to bottom, the figure illustrates the position of \tilde{C}_{i-1} , \tilde{C}_i , and of $U_{i-1} - \delta - \Delta$ for $\delta < \frac{\pi}{N}$ and $\delta > \frac{\pi}{N}$.

the beads will impact we need to impose that i and $i - 1$ reach simultaneously $U_{i-1} - \delta - \Delta$:

$$T_{i-1}^k + \left(\frac{\pi}{N} - \Delta\right)\frac{1}{\bar{v}} = T_i^k + \left(\frac{\pi}{N} - \delta\right)\frac{1}{\bar{v}} + \frac{(\delta + \Delta)}{f\bar{v}},$$

This clearly holds because $i - 1$ is traveling in its desired sweeping arc, therefore its speed is simply \bar{v} ; on the other hand, bead i initially travels in its arc but eventually leaves it and therefore its speed becomes $f\bar{v}$.

Solving for Δ we have:

$$\Delta = \frac{-f}{f+1}\bar{v}(T_i^k - T_{i-1}^k) + \frac{f-1}{f+1}\delta. \quad (5.17)$$

Note that requiring i and $i - 1$ to be in unbalanced sync is equivalent to impose $\Delta = 0$ which implies $T_i^k - T_{i-1}^k = \frac{f-1}{f}\frac{\delta}{\bar{v}}$. After impacting at $U_{i-1} - \delta - \Delta$, beads

$i - 1$ and i will change directions and head back towards \tilde{C}_{i-1} and \tilde{C}_i , that they will reach at time T_{i-1}^{k+1} and T_i^{k+1} :

$$\begin{aligned} T_{i-1}^{k+1} &= T_{i-1}^k + 2\left(\frac{\pi}{N} - \Delta\right)\frac{1}{\bar{\nu}}, \\ T_i^{k+1} &= T_i^k + 2\left(\frac{\pi}{N} + \Delta\right)\frac{1}{\bar{\nu}}. \end{aligned}$$

Recalling equation (5.17) and rearranging the terms we have:

$$\begin{bmatrix} T_{i-1}^{k+1} \\ T_i^{k+1} \end{bmatrix} = M \begin{bmatrix} T_{i-1}^k \\ T_i^k \end{bmatrix} + \frac{2\delta}{\bar{\nu}} \frac{1-f}{f} \begin{bmatrix} 1 \\ -1 \end{bmatrix} + \frac{1}{\bar{\nu}} \frac{2\pi}{N} \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

where

$$M = \begin{bmatrix} 1 - \frac{2f}{f+1} & \frac{2f}{f+1} \\ \frac{2f}{f+1} & 1 - \frac{2f}{f+1} \end{bmatrix}. \quad (5.18)$$

Note that the dynamics matrix M is doubly stochastic since $f \in]0.5, 1[$. Let

$T^k = [T_1^k, \dots, T_N^k]^T$, any time an impact between $i - 1$ and i occurs we have:

$$\begin{bmatrix} T_1^k \\ \vdots \\ T_{i-1}^{k+1} \\ T_i^{k+1} \\ \vdots \\ T_N^k \end{bmatrix} = A_{i-1} \begin{bmatrix} T_1^k \\ \vdots \\ T_{i-1}^k \\ T_i^k \\ \vdots \\ T_N^k \end{bmatrix} + \frac{2\delta}{\bar{\nu}} \frac{1-f}{f} u_{i-1} + \frac{1}{\bar{\nu}} \frac{2\pi}{N} w_{i-1},$$

where

$$A_{i-1} = \begin{bmatrix} 1 & 0 & \dots & & 0 \\ 0 & \ddots & & & 0 \\ \vdots & & M_{11} & M_{12} & \vdots \\ \vdots & & M_{21} & M_{22} & \vdots \\ & & & & \ddots \\ 0 & 0 & \dots & & 1 \end{bmatrix}, \quad u_{i-1} = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ -1 \\ \vdots \\ 0 \end{bmatrix}, \quad w_{i-1} = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 1 \\ \vdots \\ 0 \end{bmatrix},$$

and M_{ij} are the entries of the matrix M defined in equation (5.18). After any bead has met both its two neighbors we have:

$$T^{k+2} = \tilde{A}T^k + \frac{2\delta}{\bar{\nu}} \frac{1-f}{f} \tilde{U} + \frac{1}{\bar{\nu}} \frac{2\pi}{N} \tilde{W}, \quad (5.19)$$

where $\tilde{A} = \prod_{m=1}^N A_{j_m}$, $j_m \in \{1, \dots, N\}$ (the value of j_m depends on the order of the impacts), $\tilde{U} = \sum_{r=1}^N \left(\prod_{m=1+r}^N A_{j_m} \right) u_{j_r}$, and $\tilde{W} = \sum_{r=1}^N \left(\prod_{m=1+r}^N A_{j_m} \right) w_{j_r}$. Note that the matrix \tilde{A} is ergodic doubly stochastic because the associated graph is connected and it is the product of doubly stochastic matrices. Furthermore, for all $k \in \mathbb{N}$ the dynamics matrix \tilde{A} is actually constant because by assumption the order of the impacts is just like in Figure 5.5. Since the dynamics (5.19) is time invariant we can write the trajectory in closed-form:

$$T^{2k+1} = \tilde{A}^k T^1 + \left(\sum_{j=1}^{k-1} \tilde{A}^j \right) \left(\frac{2\delta}{\bar{\nu}} \frac{1-f}{f} \tilde{U} + \frac{1}{\bar{\nu}} \frac{2\pi}{N} \tilde{W} \right).$$

We can then calculate:

$$T^{2k+1} - T^{2(k-1)+1} = (\tilde{A}^k - \tilde{A}^{k-1})T^1 + \tilde{A}^{(k-1)} \left(\frac{2\delta}{\bar{\nu}} \frac{1-f}{f} \tilde{U} + \frac{1}{\bar{\nu}} \frac{2\pi}{N} \tilde{W} \right).$$

Since \tilde{A} is ergodic and doubly stochastic, then $\lim_{k \rightarrow +\infty} \tilde{A}^k = \frac{\mathbf{1}\mathbf{1}^T}{N}$ (see [30]), and therefore:

$$\begin{aligned} \lim_{k \rightarrow +\infty} T^{2k+1} - T^{2(k-1)+1} &= \left(\frac{\mathbf{1}\mathbf{1}^T}{N} - \frac{\mathbf{1}\mathbf{1}^T}{N} \right) T^1 + \frac{\mathbf{1}\mathbf{1}^T}{N} \left(\frac{2\delta}{\bar{\nu}} \frac{1-f}{f} \tilde{U} + \frac{1}{\bar{\nu}} \frac{2\pi}{N} \tilde{W} \right), \\ &= \frac{\mathbf{1}\mathbf{1}^T}{N} \sum_{r=1}^N \left(\frac{2\delta}{\bar{\nu}} \frac{1-f}{f} \prod_{m=1+r}^N A_{j_m} u_{j_r} + \frac{1}{\bar{\nu}} \frac{2\pi}{N} \prod_{m=1+r}^N A_{j_m} w_{j_r} \right), \\ &= \frac{2\delta}{\bar{\nu}} \frac{1-f}{f} \sum_{r=1}^N \left(\frac{\mathbf{1}\mathbf{1}^T}{N} u_{j_r} \right) + \frac{1}{\bar{\nu}} \frac{2\pi}{N} \sum_{r=1}^N \left(\frac{\mathbf{1}\mathbf{1}^T}{N} w_{j_r} \right), \\ &= 0 + \frac{1}{\bar{\nu}} \frac{2\pi}{N} \sum_{r=1}^N 2 \frac{\mathbf{1}}{N}, \\ &= \frac{2}{\bar{\nu}} \frac{2\pi}{N}. \end{aligned}$$

The third equality holds because $\mathbf{1}^T A_{j_m} = \mathbf{1}^T$ for all $j_m \in \{1, \dots, N\}$ since A_{j_m} is doubly stochastic, while the fourth equality holds because $\mathbf{1}^T u_{j_r} = 0$ and $\mathbf{1}^T w_{j_r} = 2$ for all $j_r \in \{1, \dots, N\}$.

Case (ii) Let us now suppose $\delta \geq \frac{\pi}{N}$. To calculate where beads $i - 1$ and i will impact we need to solve (see Figure 5.7):

$$T_{i-1}^k + \left(\delta - \frac{\pi}{N}\right) \frac{1}{h\bar{\nu}} + \left(\frac{2\pi}{N} - \delta - \Delta\right) \frac{1}{\bar{\nu}} = T_i^k + \left(\frac{\pi}{N} + \Delta\right) \frac{1}{f\bar{\nu}},$$

solving for Δ we have:

$$\Delta = \frac{-f}{f+1} \bar{\nu} (T_i^k - T_{i-1}^k) + \frac{f-1}{f+1} \delta, \quad (5.20)$$

just like for case (i). After impacting at $U_{i-1} - \delta - \Delta$ beads $i - 1$ and i will change directions and head back towards \tilde{C}_{i-1} and \tilde{C}_i . We can now calculate T_{i-1}^{k+1} and T_i^{k+1} :

$$\begin{aligned} T_{i-1}^{k+1} &= T_i^k + 2\left(\frac{\pi}{N} + \Delta\right) \frac{1}{\bar{\nu}} \\ T_i^{k+1} &= T_i^k + 2\left(\frac{\pi}{N} - \Delta\right) \frac{1}{\bar{\nu}}. \end{aligned}$$

The dynamics of T_{i-1} and T_i are just like in case (i), therefore the analysis and conclusion of case (i) are valid also for case (ii). ■

5.6 Simulations

In this section we present simulations obtained by implementing the SYNCHRONIZATION ALGORITHM on balanced and unbalanced collection of beads.

5.6.1 Balanced collection of beads

As we have seen in Section 5.5.2, it can be proved that the SYNCHRONIZATION ALGORITHM would allow the beads to get in sync if for all $i \in \{1, \dots, N\}$, $\nu_i(0) = \bar{\nu} > 0$, $\text{dist}_{\text{cc}}(L_i(0), L_{i+1}(0)) = \frac{2\pi}{N}$, $\text{dist}_{\text{cc}}(L_i(0), U_i(0)) = \frac{2\pi}{N}$, and $d_i(0) = -d_j(0)$ for $j \in \{i-1, i+1\}$. But, extensive simulations have suggested the possibility that the basin of attraction of the periodic orbit is indeed much larger.

Conjecture 1 (Balanced collection: global basin of attraction) *Let*

$\{(\theta_i(0), x_i(0))\}_{i \in \{1, \dots, N\}} \in \mathcal{A}_{0\text{-bal}}$. *Let* T_i^k *be the last instant in which bead* i *passed by the center of its desired sweeping arc before time* t *and* $T^k = [T_1^k, \dots, T_N^k]^T \in \mathbb{R}^{N \times 1}$, *then, along the trajectories of the SYNCHRONIZATION ALGORITHM:*

$$\lim_{k \rightarrow +\infty} \left\| T^k - \frac{\mathbf{1}^T T^k}{N} \mathbf{1} \right\| = 0.$$

In what follows we present the simulation results obtained by implementing the SYNCHRONIZATION ALGORITHM with $N = 8$ beads, beads are randomly positioned on \mathbb{T} , $\nu_i(0)$ uniformly distributed $\in]0, 1]$, $d_1(0) = d_2(0) = d_4(0) = d_6(0) = +1$ and $f = 0.7$.

Figure 5.8 shows the positions of the eight beads vs time. Clearly, asymptotically each bead meets its neighbor at the same location on the circle, reaching synchrony.

Figure 5.9 shows $\max_i \nu_i(t) - \min_i \nu_i(t)$, which is a measure of disagreement of the nominal speeds. As expected the disagreement goes to zero asymptotically.

In Figure 5.10, the positions and the desired sweeping arc boundaries for bead $i = 5$ are illustrated. The solid line represents $\theta_5(t)$, the dash-dot line represents $L_i(t)$, and the thicker solid line represents $U_5(t)$. The distance $\text{dist}_{\text{cc}}(L_5(t), U_5(t))$

asymptotically approaches $360/N = 45$ degrees.

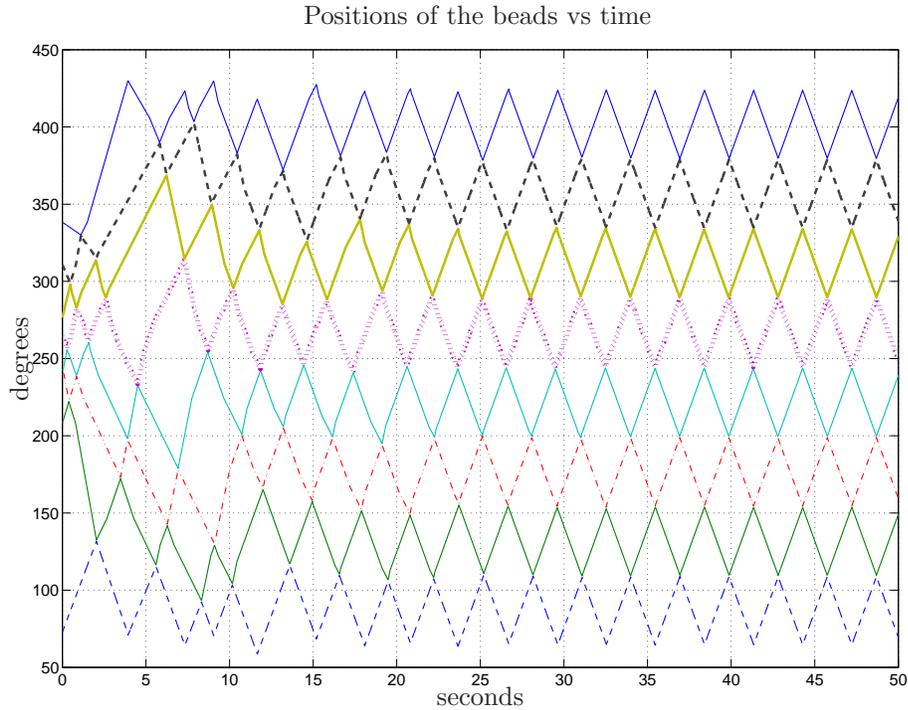


Figure 5.8. This figure shows θ_i vs time, obtained by implementing the SYNCHRONIZATION ALGORITHM with $N = 8$ beads, the beads are randomly positioned on \mathbb{T} , $\nu_i(0)$ uniformly distributed $\in]0, 1]$, $d_1(0) = d_2(0) = d_4(0) = d_6(0) = +1$, and $f = 0.7$. The positions of the beads 2, 4, 6, 8 are represented by solid lines, while the dash line, dash-dot line, point line, and thicker dash line represent the positions of beads 1, 3, 5, 7.

5.6.2 Unbalanced collection of beads

In Theorem 5.5 we have proved that if $\{(\theta_i(0), x_i(0))\}_{i \in \{1, \dots, N\}} \in \mathcal{A}_{D\text{-unbal}}$ with $D \in \{-1, +1\}$, and if the collection of beads is close to be in unbalanced synchrony, then the SYNCHRONIZATION ALGORITHM will steer the collection to

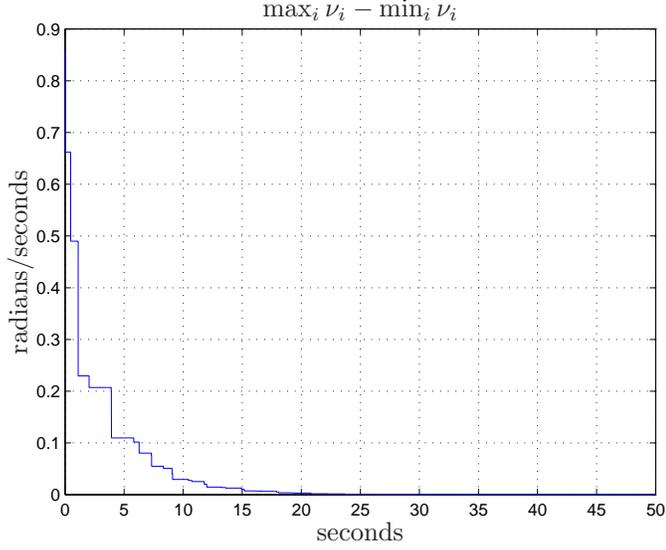


Figure 5.9. This figure shows $\max_i \nu_i - \min_i \nu_i$ vs time, obtained by implementing the SYNCHRONIZATION ALGORITHM with $N = 8$ beads, the beads are randomly positioned on \mathbb{T} , $\nu_i(0)$ uniformly distributed $\in]0, 1]$, $d_1(0) = d_2(0) = d_4(0) = d_6(0) = +1$, and $f = 0.7$.

synchrony. Also in this case, extensive simulations have suggested that the basin of attraction of the periodic orbit might be much larger.

Conjecture 2 (1-unbalanced collection: global basin of attraction) *Let $\{(\theta_i(0), x_i(0))\}_{i \in \{1, \dots, N\}} \in \mathcal{A}_{D\text{-unbal}}$ with $D \in \{-1, +1\}$. Let $\delta = \frac{2\pi f}{N^2(1-f)} < \frac{2\pi}{N}$, and $\tilde{C}_i(t)$ be the center of the counterclockwise arc $[L_i(t) + D\delta, U_i(t) + D\delta]$ for all $i \in \{1, \dots, N\}$, i.e., $\tilde{C}_i(t) = L_i(t) + D\delta + \frac{1}{2} \text{dist}_{cc}(L_i(t) + D\delta, U_i(t) + D\delta)$. Let T_i^k be the instant in which bead i passed by \tilde{C}_i for the k -th time and $T^k = [T_1^k, \dots, T_N^k]^T \in \mathbb{R}^{N \times 1}$, then, along the trajectories of the SYNCHRONIZATION ALGORITHM:*

$$\lim_{k \rightarrow +\infty} T^{2k} - T^{2(k-1)} = \mathbf{1} \frac{2 \ 2\pi}{\bar{\nu} \ N},$$

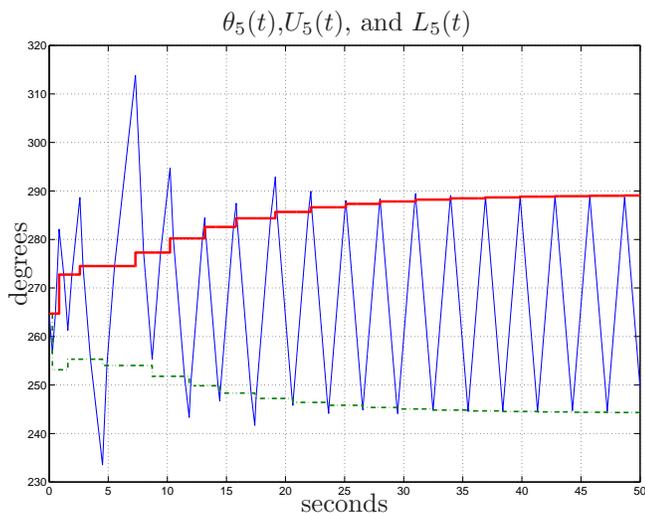


Figure 5.10. This figure shows $\theta_5(t)$ (solid line), $U_5(t)$ (thicker solid line), and $L_5(t)$ (dash-dot line), obtained by implementing the SYNCHRONIZATION ALGORITHM with $N = 8$ beads, the beads are randomly positioned on \mathbb{T} , $\nu_i(0)$ uniformly distributed in $]0, 1]$, $d_1(0) = d_2(0) = d_4(0) = d_6(0) = +1$, and $f = 0.7$. *that is, the collection of beads asymptotically reaches unbalanced synchrony.*

In what follows we present the simulation results obtained by implementing the SYNCHRONIZATION ALGORITHM with $N = 7$ beads, the beads are randomly positioned on \mathbb{T} , $\nu_i(0)$ uniformly distributed $\in]0, 1]$, $d_1(0) = d_4(0) = d_5(0) = d_7(0) = -1$, that is the collection of beads is D unbalanced with $D = -1$, and $f = 0.6$. Note that $f < \frac{N}{1+N} = \frac{7}{8}$.

Figure 5.11 shows the positions of the seven beads vs time. Clearly, asymptotically each bead meets its neighbor at the same location on the circle, reaching synchrony.

In Figure 5.12, the positions and the desired sweeping arc boundaries for bead $i = 3$ are illustrated. The solid line represents $\theta_3(t)$, the dash-dot line represents $L_3(t)$, and the thicker solid line represents $U_3(t)$. The distance $\text{dist}_{\text{cc}}(L_3(t), U_3(t))$ asymptotically approaches $360/N \approx 51.42$ degrees.

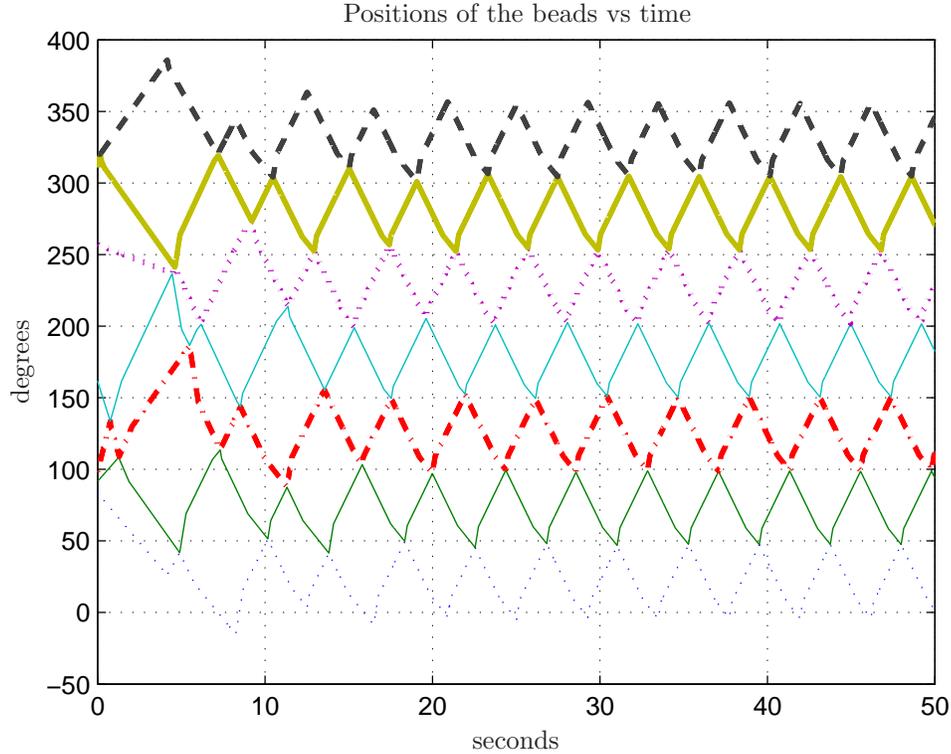


Figure 5.11. This figure shows θ_i vs time, obtained by implementing the SYNCHRONIZATION ALGORITHM with $N = 7$ beads, the beads are randomly positioned on \mathbb{T} , $\nu_i(0)$ uniformly distributed $\in]0, 1]$, $d_1(0) = d_4(0) = d_5(0) = d_7(0) = -1$, and $f = 0.6$. The positions of the beads 2, 4, 6 are represented by solid lines, while the dash line, dash-dot line, point line, and thicker dash line represent the positions of beads 1, 3, 5, 7.

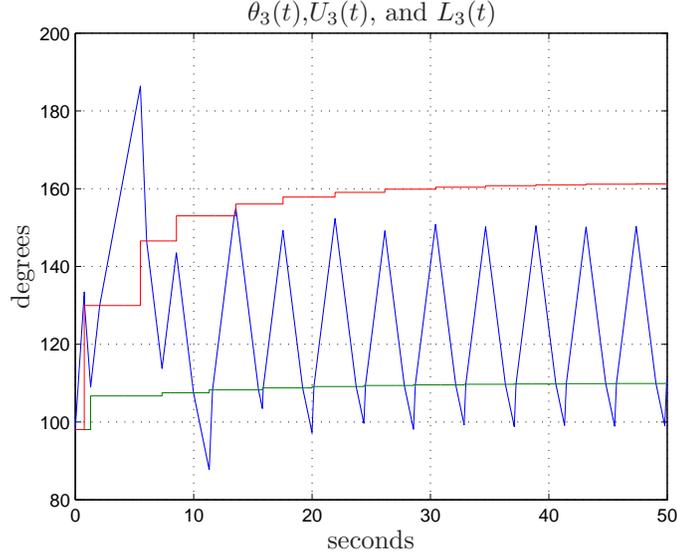


Figure 5.12. This figure shows $\theta_3(t)$ (solid line), $U_3(t)$ (thicker solid line), and $L_3(t)$ (dash-dot line), obtained by implementing the SYNCHRONIZATION ALGORITHM with $N = 7$ beads, the beads are randomly positioned on \mathbb{T} , $\nu_i(0)$ uniformly distributed in $]0, 1]$, $d_1(0) = d_4(0) = d_5(0) = d_7(0) = -1$, and $f = 0.6$.

For the more general case of D -unbalanced collections with $N > |D| > 1$, Theorem 5.4 states that $f < \frac{N/|D|}{1+N/|D|}$ is just a necessary condition for the existence of a period orbit, along which, i and $i-1$ meet always at $L_i + \frac{D}{|D|}\delta$, with $\delta < \frac{2\pi}{N}$. We conjecture that (i) $f < \frac{N/|D|}{1+N/|D|}$ is also sufficient for the existence of a periodic orbit in the most general case of $|D| > 1$, and (ii) the SYNCHRONIZATION ALGORITHM will steer the collection of D -unbalanced beads to synchrony.

Conjecture 3 (D-unbalanced collection: existence of periodic orbit) *Let $\{(\theta_i(0), x_i(0))\}_{i \in \{1, \dots, N\}} \in \mathcal{A}_{D\text{-unbal}}$, $\nu_i(t) = \bar{\nu}$, $\text{dist}_{cc}(L_i(0), L_{i+1}(0)) = \frac{2\pi}{N}$ for all $i \in \{1, \dots, N\}$. If and only if $\frac{1}{2} < f < \frac{N}{1+N}$, then there exists a periodic orbit, for the SYNCHRONIZATION ALGORITHM, along which each bead i always impacts with $i-1$ at $L_i(0) + D\delta$, where $\delta = \frac{2\pi}{N^2} \frac{f}{1-f} < \frac{2\pi}{N}$.*

Conjecture 4 (D-unbal. collection: global basin of attraction) *Let*

$\{(\theta_i(0), x_i(0))\}_{i \in \{1, \dots, N\}} \in \mathcal{A}_{D\text{-unbal}}$ *with* $N > |D| > 1$, $\delta = \frac{2\pi}{N^2} \frac{f}{1-f} < \frac{2\pi}{N}$, *and* $\tilde{C}_i(t)$ *be the center of the counterclockwise arc* $[L_i(t) + D\delta, U_i(t) + D\delta]$ *for all* $i \in \{1, \dots, N\}$, *i.e.,* $\tilde{C}_i(t) = L_i(t) + D\delta + \frac{1}{2} \text{dist}_{\text{cc}}(L_i(t) + D\delta, U_i(t) + D\delta)$. *Let* T_i^k *be the instant in which bead* i *passed by* \tilde{C}_i *for the* k -*th time and* $T^k = [T_1^k, \dots, T_N^k]^T \in \mathbb{R}^{N \times 1}$, *then, along the trajectories of the SYNCHRONIZATION ALGORITHM:*

$$\lim_{k \rightarrow +\infty} T^{2k} - T^{2(k-1)} = \mathbf{1} \frac{2}{\nu} \frac{2\pi}{N},$$

that is, the collection of beads asymptotically reaches unbalanced synchrony.

In what follows we present the results of two simulations (figures 5.13, 5.14, and 5.15, 5.16) obtained by implementing the SYNCHRONIZATION ALGORITHM with a collection of $N = 12$ beads which are D -unbalanced with $D = -2$, the beads are randomly positioned on \mathbb{T} , $\nu_i(0)$ uniformly distributed $\in]0, 1]$. Note that according to our conjectures $f < \frac{N/|D|}{1+N/|D|} = \frac{6}{7} \approx 0.857$ has to hold in order to reach unbalanced synchrony. In the first simulation $f = 0.84$, while in the second simulation $f = 0.87$, therefore we expect to the collection of beads to be in sync asymptotically in the first simulation but not in the second one.

Figure 5.13 shows the positions of the seven beads vs time. Clearly, asymptotically each bead meets its neighbor at the same location on the circle, reaching synchrony, because $f = 0.84 < \frac{6}{7}$.

In Figure 5.14, the positions and the desired sweeping arc boundaries for bead $i = 3$ are illustrated. The solid line represents $\theta_3(t)$, the dash-dot line represents $L_3(t)$, and the thicker solid line represents $U_3(t)$. The distance $\text{dist}_{\text{cc}}(L_3(t), U_3(t))$ asymptotically approaches $360/N = 30$ degrees.

Figure 5.15 shows the positions of the seven beads vs time. Clearly synchrony is not reached.

In Figure 5.16, the positions and the desired sweeping arc boundaries for bead $i = 3$ are illustrated. The solid line represents $\theta_3(t)$, the dash-dot line represents $L_3(t)$, and the thicker solid line represents $U_3(t)$. The distance $\text{dist}_{\text{cc}}(L_3(t), U_3(t))$ asymptotically approaches $360/N = 30$ degrees.

5.7 Summary

We presented and analyzed an algorithm that synchronizes a collection of N agents or beads, moving on a ring, so that each bead patrols only a sector of the ring. The algorithm is distributed and requires that two agents exchange information only when they meet. We proved that the proposed algorithm allows the agents to reach the desired steady state for certain initial conditions. Simulations show convergence to the desired steady state for a larger set of initial conditions.

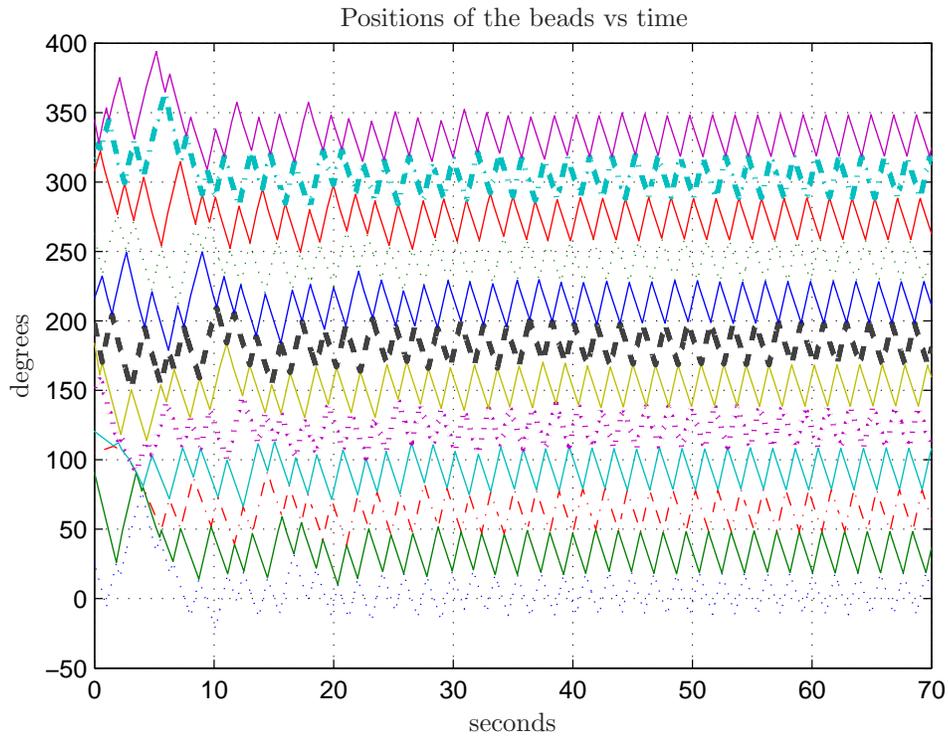


Figure 5.13. This figure shows θ_i vs time, obtained by implementing the SYNCHRONIZATION ALGORITHM with $N = 12$ beads, the beads are randomly positioned on \mathbb{T} , $\nu_i(0)$ uniformly distributed $\in]0, 1]$, $d_1(0) = d_2(0) = d_4(0) = d_6(0) = d_7(0) = d_9(0) = d_{12}(0) = -1$, and $f = 0.84$. The positions of the beads 2, 4, 6, 8, 10, 12 are represented by solid lines, while the dash line, dash-dot line, point line, and thicker dash line represent the positions of beads 1, 3, 5, 7, 9, 11.

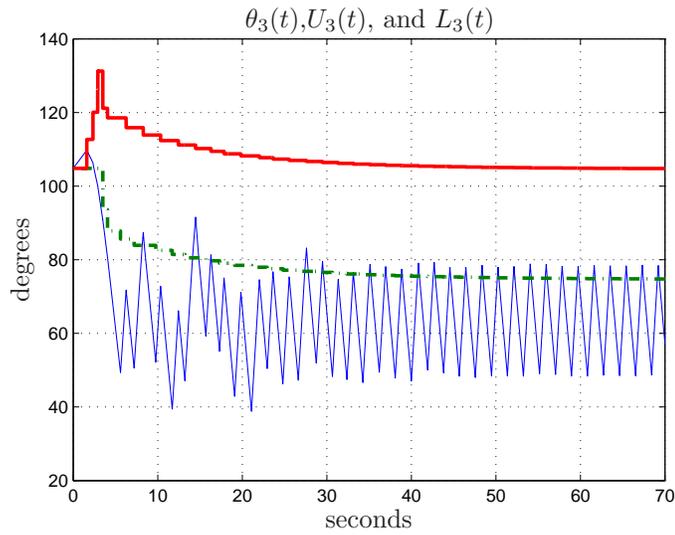


Figure 5.14. This figure shows $\theta_3(t)$ (solid line), $U_3(t)$ (thicker solid line), and $L_3(t)$ (dash-dot line), obtained by implementing the SYNCHRONIZATION ALGORITHM with $N = 12$ beads, the beads are randomly positioned on \mathbb{T} , $\nu_i(0)$ uniformly distributed in $]0, 1]$, $d_1(0) = d_2(0) = d_4(0) = d_6(0) = d_7(0) = d_9(0) = d_{12}(0) = -1$, and $f = 0.84$.

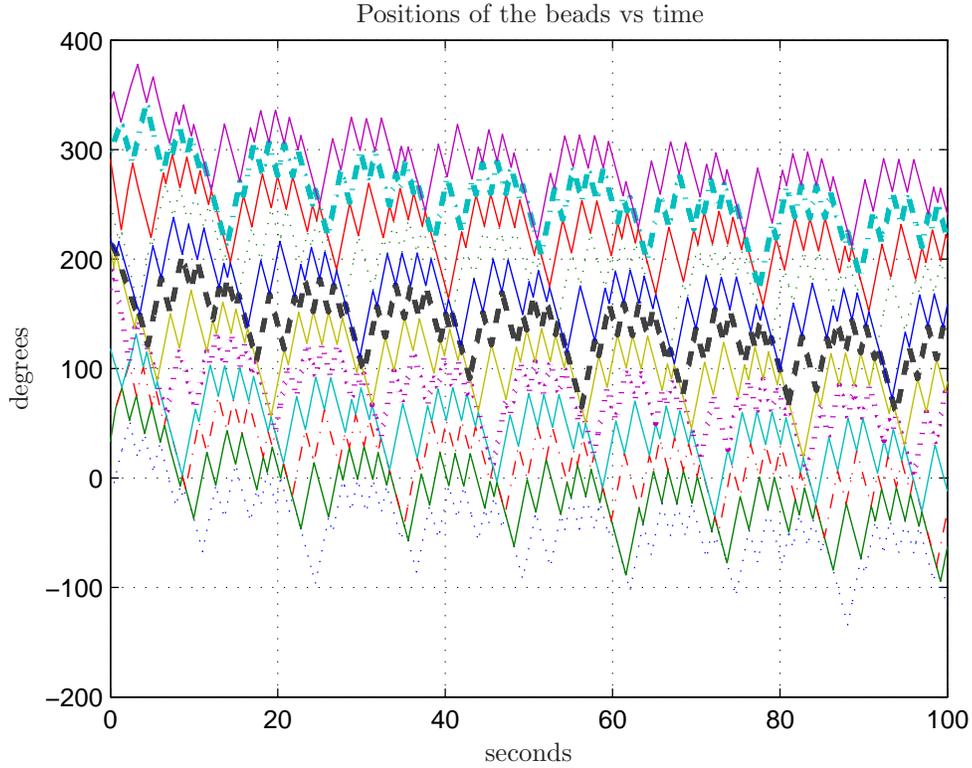


Figure 5.15. This figure shows θ_i vs time, obtained by implementing the SYNCHRONIZATION ALGORITHM with $N = 12$ beads, the beads are randomly positioned on \mathbb{T} , $\nu_i(0)$ uniformly distributed $\in]0, 1]$, $d_1(0) = d_4(0) = d_6(0) = d_7(0) = d_8(0) = d_9(0) = d_{10}(0) = -1$, and $f = 0.87$. The positions of the beads 2, 4, 6, 8, 10, 12 are represented by solid lines, while the dash line, dash-dot line, point line, and thicker dash line represent the positions of beads 1, 3, 5, 7, 9, 11.

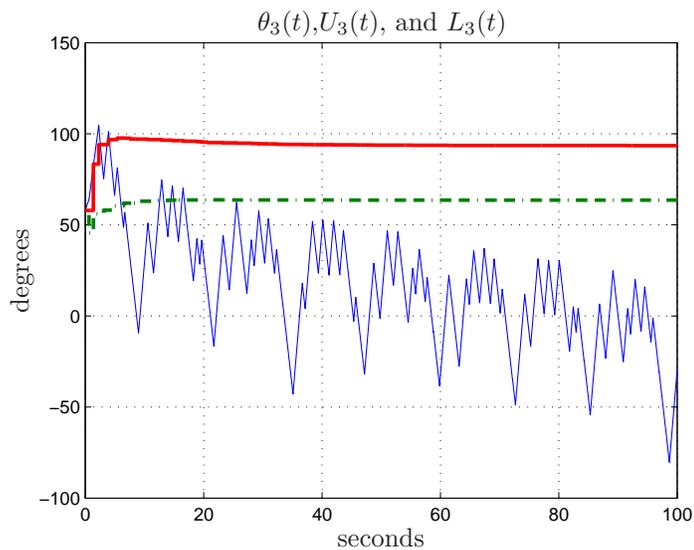


Figure 5.16. This figure shows $\theta_3(t)$ (solid line), $U_3(t)$ (thicker solid line), and $L_3(t)$ (dash-dot line), obtained by implementing the SYNCHRONIZATION ALGORITHM with $N = 12$ beads, the beads are randomly positioned on \mathbb{T} , $\nu_i(0)$ uniformly distributed in $]0, 1]$, $d_1(0) = d_4(0) = d_6(0) = d_7(0) = d_8(0) = d_9(0) = d_{10}(0) = -1$, and $f = 0.87$.

Chapter 6

Conclusions

In this thesis we addressed the problem of approximating and patrolling a planar contour by a fleet of autonomous vehicles with limited communication capabilities. We proposed and proved the correctness of distributed algorithms that will enable the autonomous vehicles to perform such tasks.

Bibliography

- [1] P.-A. Absil, R. Mahony, and B. Andrews. Convergence of the iterates of descent methods for analytic cost functions. *SIAM Journal on Control and Optimization*, 6(2):531–547, 2005.
- [2] D. Angeli. Intrinsic robustness of global asymptotic stability. *Systems & Control Letters*, 38(4-5):297–307, 1999.
- [3] D. Angeli and P.-A. Bliman. Stability of leaderless discrete-time multi-agent systems. *Mathematics of Control, Signals and Systems*, 18(4):293–322, 2006.
- [4] D. Bauso, L. Giarré, and R. Pesenti. Distributed consensus in networks of dynamic agents. In *IEEE Conf. on Decision and Control and European Control Conference*, pages 7054–7059, Seville, Spain, 2005.
- [5] A. L. Bertozzi, M. Kemp, and D. Marthaler. Determining environmental boundaries: Asynchronous communication and physical scales. In V. Kumar, N. E. Leonard, and A. S. Morse, editors, *Cooperative Control*, volume 309 of *Lecture Notes in Control and Information Sciences*, pages 25–42. Springer Verlag, 2004.
- [6] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis. Conver-

- gence in multiagent coordination, consensus, and flocking. In *IEEE Conf. on Decision and Control and European Control Conference*, pages 2996–3000, Seville, Spain, December 2005.
- [7] D. W. Casbeer, D. B. Kingston, R. W. Beard, T. W. McLain, S.-M. Li, and R. Mehra. Cooperative forest fire surveillance using a team of small unmanned air vehicles. *International Journal of Systems Sciences*, 37(6):351–360, 2006.
- [8] J. Clark and R. Fierro. Mobile robotic sensors for perimeter detection and tracking. *ISA Transactions*, 46(1):3–13, 2007.
- [9] B. Cooley and P. K. Newton. Iterated impact dynamics of N -beads on a ring. *SIAM Review*, 47(2):273–300, 2005.
- [10] J. Cortés. Distributed algorithms for reaching consensus on arbitrary functions. *Automatica*, October 2006. Submitted.
- [11] J. A. Fax and R. M. Murray. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9):1465–1476, 2004.
- [12] M. Fielder. *Special Matrices and their Applications in Numerical Mathematics*. Martinus Nijhoff Publishers, 1986.
- [13] P. M. Gruber. Approximation of convex bodies. In P. M. Gruber and J. M. Willis, editors, *Convexity and its Applications*, pages 131–162. Birkhäuser Verlag, 1983.
- [14] P. M. Gruber. Aspect of approximation of convex bodies. In P. M. Gruber

- and J. M. Willis, editors, *Handbook of Convex Geometry*, volume A, pages 319–345. Elsevier, Oxford, UK, 1993.
- [15] Y. Hatano and M. Mesbahi. Agreement over random networks. In *IEEE Conf. on Decision and Control*, pages 2010–2015, Paradise Island, Bahamas, December 2004.
- [16] U. Helmke and J. B. Moore. *Optimization and Dynamical Systems*. Springer Verlag, New York, 1994.
- [17] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, UK, 1985.
- [18] S. Huang, M. R. James, D. Nesic, and P. M. Dower. Analysis of input to state stability for discrete time nonlinear systems via dynamic programming. *Automatica*, 41(12):2055–2065, 2005.
- [19] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.
- [20] Z.-P. Jiang and Y. Wang. Input-to-state stability for discrete-time nonlinear systems. *Automatica*, 37:857–869, 2001.
- [21] H. H. Johnson and A. Vogt. A geometric method for approximating convex arcs. *SIAM Journal on Applied Mathematics*, 38(2):317–325, 1980.
- [22] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.

- [23] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, Englewood Cliffs, NJ, 2 edition, 1995.
- [24] D. B. Kingston, W. Ren, and R. W. Beard. Consensus algorithms are input-to-state stable. In *American Control Conference*, pages 1686–1690, Portland, OR, June 2005.
- [25] J. P. LaSalle. *The Stability and Control of Discrete Processes*, volume 62 of *Applied Mathematical Sciences*. Springer Verlag, New York, 1986.
- [26] J. Lin, A. S. Morse, and B. D. O. Anderson. The multi-agent rendezvous problem: An extended summary. In V. Kumar, N. E. Leonard, and A. S. Morse, editors, *Proceedings of the 2003 Block Island Workshop on Cooperative Control*, volume 309 of *Lecture Notes in Control and Information Sciences*, pages 257–282. Springer Verlag, New York, 2004.
- [27] S. Lojasiewicz. Sur les trajectoires du gradient d’une fonction analytique. *Seminari di Geometria 1982-1983*, pages 115–117, 1984. Istituto di Geometria, Dipartimento di Matematica, Università di Bologna, Bologna, Italy.
- [28] D. Marthaler and A. L. Bertozzi. Tracking environmental level sets with autonomous vehicles. In S. Butenko, R. Murphey, and P. M. Pardalos, editors, *Recent Developments in Cooperative Control and Optimization*, pages 317–330. Kluwer Academic Publishers, 2003.
- [29] D. E. McLure and R. A. Vitale. Polygonal approximation of plane convex bodies. *Journal of Mathematical Analysis and Applications*, 51(2):326–358, 1975.

- [30] L. Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2):169–182, 2005.
- [31] R. Olfati-Saber, E. Franco, E. Frazzoli, and J. S. Shamma. Belief consensus and distributed hypothesis testing in sensor networks. In P.J. Antsaklis and P. Tabuada, editors, *Network Embedded Sensing and Control. (Proceedings of NESC'05 Worskhop)*, volume 331 of *Lecture Notes in Control and Information Sciences*, pages 169–182. Springer Verlag, New York, 2006.
- [32] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.
- [33] S. Patterson, B. Bamieh, and A. E. Abbadi. Brief announcement: Convergence analysis of scalable gossip protocols. In *International Symposium on Distributed Computing (DISC 2006)*, pages 540–542, September 2006.
- [34] W. Ren, R. W. Beard, and E. M. Atkins. A survey of consensus problems in multi-agent coordination. In *American Control Conference*, pages 1859–1864, Portland, OR, June 2005.
- [35] A. Savvides, J. Fang, and D. LyMBERopoulos. Using mobile sensing nodes for boundary estimation. In *Workshop on Applications of Mobile Embedded Systems (Held in conjunction with MobiSys 2004)*, Boston, MA, June 2004.
- [36] L. Scardovi, A. Sarlette, and R. Sepulchre. Synchronization and balancing on the N -torus. *Systems & Control Letters*, 56(5):335–341, 2007.

- [37] E. D. Sontag. Input to state stability: Basic concepts and results. In P. Nistri and G. Stefani, editors, *Nonlinear and Optimal Control Theory*, Lecture Notes in Mathematics, pages 163–220. Springer Verlag, 2006.
- [38] D. P. Spanos, R. Olfati-Saber, and R. M. Murray. Approximate distributed Kalman filtering in sensor networks with quantifiable performance. In *Symposium on Information Processing of Sensor Networks (IPSN)*, pages 133–139, Los Angeles, CA, April 2005.
- [39] S. Susca, F. Bullo, and S. Martínez. Synchronization of beads on a ring. In *IEEE Conf. on Decision and Control*, pages 4845–4850, New Orleans, LA, December 2007.
- [40] H. Tanner, A. Jadbabaie, and G. J. Pappas. Stable flocking of mobile agents, Part I: Fixed topology. In *IEEE Conf. on Decision and Control*, pages 2010–2015, Maui, HI, December 2003.
- [41] E. Trost. Über eine Extremalaufgabe. *Nieuw Archief voor Wiskunde*, 2:1–3, 1949.
- [42] L. Xiao, S. Boyd, and S.-J. Kim. Distributed average consensus with least-mean-square deviation. *Journal of Parallel and Distributed Computing*, 67(1):33–46, 2007.
- [43] F. Zhang and N. E. Leonard. Generating contour plots using multiple sensor platforms. In *IEEE Swarm Intelligence Symposium*, pages 309–316, Pasadena, CA, June 2005.

Appendix A

Metzler matrices

We begin by introducing some definitions and notations taken from Chapter 5 in [12]. For $A \in \mathbb{R}^{n \times m}$, we let $A \succ (\succeq)0$ denote that the elements of A are positive (resp. non-negative). For $A \in \mathbb{R}^{n \times n}$, we let $A > (\geq)0$ denote that the matrix A is positive definite (resp. positive semidefinite) and we let $\rho(A)$ be the spectral radius of A .

Definition A.1 1. A square matrix is said to be of class Z if all of its off-diagonal elements are non-positive.

2. A square matrix A of class Z is said to belong to class K if there exists a matrix $C \succeq 0$ and a number $k > \rho(C)$ such that $A = kI - C$.

3. A square matrix A of class Z is said to belong to class K_o if there exists a matrix $C \succeq 0$ and a number $k \geq \rho(C)$ such that $A = kI - C$.

Matrices of class K and K_o are called M-matrices (Metzler matrices). Note that equivalent definitions of matrices of class K and K_o are provided in [12].

We briefly recall that a matrix A is irreducible if and only if its associated directed graph is strongly connected. The following theorem presents a few useful properties of these matrices.

Theorem A.1 1. If $A \in Z$ and if there exists $x \succ 0$ such that $Ax \succeq 0$, then $A \in K_o$.

2. If $A \in \mathbb{R}^{n \times n}$ belongs to the class K_o , is irreducible and singular, then there exists $u \succ 0$ such that $Au = 0$. Moreover, $\text{rank}(A) = n - 1$.

3. If $A \in K_o$, then every eigenvalue of A has nonnegative real part.

Theorem A.1 is proved in Theorem 5.11, 5.8 and 5.3 in [12], respectively.

Next we present an application of these concepts. For $\beta \in]0, 1]$, let $c_i \in [\beta, 1]$ for all $i \in \{1, \dots, n\}$, and define the $n \times n$ square matrices

$$A(c_1, \dots, c_n) = \begin{bmatrix} -c_1 - c_2 & c_2 & 0 & \dots & c_1 \\ c_2 & -c_2 - c_3 & c_3 & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ c_1 & 0 & \dots & c_n & -c_n - c_1 \end{bmatrix},$$

These matrices have the following useful properties.

Lemma A.1 The matrices $A(c_1, \dots, c_n)$ have rank $n - 1$ and their eigenvalues have non-positive real part.

Proof. Let $M = -A(c_1, \dots, c_n)$ so that $M \in Z$. By Theorem A.1(i) with $x = \mathbf{1}$, it can be seen that $M \in K_o$. Moreover M is irreducible because represents a strongly connected graph, and singular because $M\mathbf{1} = 0$. We can then apply

Theorem A.1(ii) which proves that the origin is an eigenvalue of M with multiplicity 1. With Theorem A.1(iii) finally it can be proved that M is positive semidefinite which implies that $A(c_1, \dots, c_n)$ is negative semidefinite. ■