UNIVERSITÀ DEGLI STUDI DI PALERMO

DEPARTMENT OF ENERGY, INFORMATION TECHNOLOGY ENGINEERING, AND MATHEMATICAL MODELLING

PHD COURSE IN: **Electrical, Electronics and Telecommunication Engineering, Mathematics and Automation**

SPECIALISM IN: **Mathematics and Automation for Scientific and Technological Innovation**

S.S.D. ING-INF/04

Complex Networked Systems: Convergence Analysis, Dynamic Behaviour, and Security

PhD Candidate
Eng. Damiano La Manna

PhD Course Coordinator
Prof. Maria Stella Mongiovì

Tutor
Eng. Adriano Fagiolini

Co-Tutor
Eng. Vincenzo Di Dio

CYCLE OF STUDY XXVI - YEAR 2016

To my family,
my true strength and point of reference,
for their trust and support, and
for always having been close to me.

" The progressive development of man is vitally dependent on invention. It is the most important product of his creative brain. Its ultimate purpose is the complete mastery of mind over the material world, the harnessing of the forces of nature to human needs . . . Science is but a perversion of itself unless it has as its ultimate goal the betterment of humanity! "

Nikola Tesla

Acknowledgements

I would like to express my gratitude to my tutor Adriano Fagiolini and my co-tutor Vincenzo Di Dio for their precious guidance to the development of this work. I deeply appreciate their continuous support, useful advices and complete availability. I am grateful with them for having contributed to my professional growth, both from the scientific and the human viewpoint. A special thanks goes to Gabriele Oliva for his valuable collaboration and significant contribution in achieving many of the results presented in this thesis. I would also like to thank Prof. Francesco Alonge for useful discussions on many aspects of the work and for being always present. Last but not least, I wish to acknowledge Fabio Pasqualetti, who advised me during the visiting scholar period at Riverside. I am thankful to him for his suggestions, for giving me a formal theoretic approach to solve problems in cyber-security, and for making the period abroad a wonderful experience.

Abstract

Complex networked systems are a modern reference framework through which very different systems from far disciplines, such as biology, computer science, physics, social science, and engineering, can be described. They arise in the great majority of modern technological applications. Examples of real complex networked systems include embedded systems, biological networks, large-scale systems such as power generation grids, transportation networks, water distribution systems, and social network. In the recent years, scientists and engineers have developed a variety of techniques, approaches, and models to better understand and predict the behaviour of these systems, even though several research and industrial challenges are still open.

This thesis addresses the study of different properties of complex networked systems and their applications. The main contribution of the work can be considered as threefold: the study of interaction among agents and the relative data clustering in small groups, the analysis of convergence conditions towards a common or multiple agreements, and the investigation of security aspects concerning the detection of perturbations that can propagate across network components and subnetworks. Firstly, a novel approach to solve data clustering problems within wireless sensor networks is proposed, including additional constraints on the distance among cluster centroids. A key feature of the presented algorithm is its ability to partition the original raw dataset into a suboptimal set of clusters, without the requirement of a priori specification of the desired cluster number. Secondly, after introducing a mathematical framework describing the dynamic model of a

complex network, a set of centralised and distributed conditions are determined, allowing the detection of the connectedness of the network's underlying topological structure, its convergence to a steady state, and even to an agreement. To this purpose, the so-called Hegselmann-Krause opinion dynamics model is adopted, which describes the way agents of a community dynamically influence with each other. Thirdly, the problem of optimal sensor location within a class of networked systems, which requires the detection of unknown input disturbance, is addressed. To this aim, a measure simultaneously based on the properties of controllability and observability of the network is used, which allows different sensor locations to be evaluated with respect to the location of the signal to be detected. These results inform the design of robust networks, and they suggest that sensor location methods based on the network topology alone may lead to poor detection performance.

Contents

List of Figures

List of Tables

Publications

Journal Papers

1. G. Oliva, D. La Manna, A. Fagiolini and R. Setola. Distributed Steady-State, Connectedness and Agreement Detection Criteria for HK Model. Automatica-Journal-Elsevier. Submitted, 2015.

2. G. Oliva, D. La Manna, A. Fagiolini and R. Setola. Distributed Data clustering via Opinion Dynamics. International Journal of Distributed Sensor Networks, 2015.

3. V. Li Vigni, D. La Manna, E. Riva San Severino, V. Di Dio, P. Romano, P. Di Buono, M. Pinto, R. Miceli and C. Giaconia. Proof of Concept of an Irradiance Estimation System for Reconfigurable Photovoltaic Arrays. Energies 8(7), 6641-6657, 2015.

4. D. La Manna, V. Li Vigni, E. Riva San Severino, V. Di Dio and P. Romano (2014). Reconfigurable Electrical Interconnection Strategies for Photovoltaic Arrays: A review. Renewable and Sustainable Energy, Elsevier 33: 412-426, 2014.

Conference Papers

5. D. La Manna, A. Fagiolini, F. Pasqualetti. Attack and Defense Location in Line Network Systems. International Conference on Decision and Control (CDC). December, Las Vegas, Nevada (USA). Submitted, 2016.

6. G. Oliva, D. La Manna, A. Fagiolini and R. Setola. Distance-Constrained Data Clustering by Combined k-means Algorithms and Opinion Dynamics Filters. 22nd Mediterranean Conference on Control and Automation. Palermo, Italy, 2014.

7. G. Cipriani, V. Di Dio, D. La Manna, R. Miceli, G. Ricco Galluzzo. Technical and Economical Comparison between Different Topologies of PV Plant Under Mismatch Effect. IEEE International Conference on Ecological Vehicles and Renewable Energies (EVER). Monte Carlo, France, 2014.

8. G. Cipriani, V. Di Dio, D. La Manna, F. Massaro, R. Miceli and R. Zizzo. Economic Analysis on Dynamic Photovoltaic Systems in New Italian, Feed in Tariffs Context. International Conference on Clean Electrical Power ICCEP. Alghero, Italy,, 2013.

Chapter 1

Introduction

Nowadays, the world where we live and operate is networked. We drive to work on a network of roads and fly across a web of cities connected by commercial airlines. We communicate using an increasingly elaborate set of devices ranging from phones, to computers or tablets, all of which are connected through Internet [1]. Our personal choices are often influenced by the opinions of other members belonging to our social community. We shop at stores and dine at restaurants connected through food and goods delivery networks. Our governments are networked as are our financial and societal infrastructures. Everything around us may be considered as a networked system whose complexity increases with the number of components as well as the interactions among them. Complex networked systems can be interpreted as large collections of entities that are interconnected by links, whose structure is irregular, complex and dynamically evolving in time [7]. Such networks arise in many applications involving, e.g., wireless sensor networks, swarms of robots, advanced communication systems, power distribution networks, social and economic networks, smart grids. They have been studied for the purpose of modelling, analysing, and controlling different kinds of real world systems. In this respect, to achieve a correct description of their behaviour, the specification of the dynamics of the individual entities, or agents, acting within the system, must be accompanied by a characterisation of their interaction, indicating, e.g., how they communicate with each others, namely the graph of their interaction topology.

1.1 Current Research and Challenges

The main goal of modern research on modelling, simulation and analysis of complex networked systems is to develop capabilities for understanding, designing and control of those networked systems [10]. To carry out this, it needs to better understand the structure and the dynamics of the network, predictive modelling and simulation of dynamic networks, and situational awareness and control. Despite the use of different analysis tools, network properties such as connectivity, efficiency, and robustness are critical to both control design and complex-network modelling [13, 19, 96].

Network theory provides tools that characterise the growth and topology of distributed networks in relation to their navigability, congestion, clustering, and robustness to failure [17, 27, 79]. Nevertheless, the need for new paradigms for control design is particularly evident in large-scale interconnected multi-agent systems, where signals need to how quickly and efficiently, but interconnected components may not be able to store and manipulate the complete state of the system. Although complexity barriers render the task of designing controllers for high-dimensional systems impractical, the ability to reason about global network properties based on locally available information enables the design of decentralised control laws.

To this regard, distributed systems and networks have received much attention in the last years because of their flexibility and computational performance. Frequently, the tasks that must be accomplished by a group of autonomous agents requires that they agree on the value of suitable parameters, which can be effectively achieved within the framework of so-called consensus algorithms. Several kinds of these algorithms have been proposed in the literature [55, 80, 102]. A large research effort in this direction is witnessed by pioneer researchers such as French [32] and DeGroot [38], who proposed first frameworks to model such problems with respect to groups of experts making decisions together. The study of agreement has been further fostered by the work of Olfati-Saber and Murray [71] were distributed procedures are devised to let a set of agents reach a common standpoint by means of only local interaction, without a central authority. Among the proposed approaches, the simplest ones are those based on linear consensus iterations [33, 99] where each node updates its state, at each time instant, according to a weighted combination of its own value and those received from its neighbours. As it is known, the combination weights determine the convergence velocity towards a steady state value.

A further important research direction is the study of opinion dynamics within interacting agent networks, whether consisting of humans, robots or sensors, which has always attracted large part of the scientific community, also for the challenge of capturing the major features of such complex processes. To this respect, the Hegselmann-Krause (HK) opinion dynamics model was first proposed and studied in [41]. Many works have later established properties of the HK model and showed their application [5, 6]. This model is based on the assumption that agents having far different opinions do not communicate, while agents with close enough opinions that is, the difference in opinions is less than an influence parameter ε will influence each other [43]. In spite of its simplicity, the HK model displays nontrivial characteristics, since the graph topology underlying the communication among the agents is timevarying and depends on the state in a nonlinear and discontinuous way [4, 14, 16]. There is, however, no guarantee to obtain an agreement, and several clusters of opinions can be obtained, based on the choice of ε [40, 85, 101]. Notwithstanding the large research effort focused on taming the complexity of such a model [18, 51, 60, 68], all the inherent features has not been fully understood yet.

Another crucial issue to be addressed for complex networked systems is linked to their security. Networked systems are prone to malicious attacks and faults against individual nodes and interconnection dynamics [15, 65, 86]. Perturbations propagate across components and subnetworks and may cascade into the failure of all interconnected parts [12]. Reliable operation of network systems relies on the prompt detection and remedy of malfunctions. The ability to detect perturbation in network systems depends on the intrinsic structure of the system, as well as on the location of sensors and monitors [77, 94]. New security approaches for complex networks, improve the design of networks to be more robust against failures or attacks, detecting, i.e., potential catastrophic failure on the power grid and preventing or mitigating its effects, understanding how populations will respond to the availability of new energy sources or changes in energy policy, and detecting subtle vulnerabilities in large software systems to intentional attacks [20, 30, 64].

1.2 Thesis Contributions

The main contribution of each chapter are as follows.

Chapter 2 introduces the general notation that will be adopted in the document. We first recall definitions from algebraic graph theory and notions from network science analysis. We provide mathematical models that are used in the literature, describing the dynamical behaviour of network agents, and the mechanism by which opinions evolve and lead to an agreement or multiple local consensus. We finally present structural properties of networked systems, such as controllability and observability, which will be used to detect external input attacks.

Chapter 3 gives centralised and distributed conditions for the HK model aimed at detecting at time instant t the convergence to a steady state, the connectedness of the underlying topological structure, and the convergence to an agreement at time instant $t+1$. This is obtained by using an alternative formulation of the HK model, which provides some insights on the complexity of the model. According to these distributed conditions, each agent i is assumed to known, at each time step, not only the opinion of its neighbours j, but also the size of the neighbourhood of the agents j, i.e., the number of agents they are connected to.

Chapter 4 provides a distributed method to partition a large set of data into clusters, characterised by small in-group and large out-group distances. We assume a wireless sensor network where each sensor is given a set of data, and the objective is to provide a way to group the sensor values in homogeneous clusters based on the type of information. In the previous literature, the desired number of clusters must be specified a priori by the user. In our approach, the clusters are constrained to have centroids with a distance at least ε among them, and the number of desired clusters can be unspecified. The inclusion of this type of constraint within the problem formulation allows better clustering performance. However, existing and available algorithms fail to cope with such a constraint. We show how a solution based on the HK model is able to find a sub-optimal admissible solution. Even though the HK model is a centralised algorithm, we provide in this work a distributed implementation, based on a combination of distributed consensus algorithms.

Chapter 5 discusses the general problem of network security and proposes an optimal sensor placement method to detect an external attack. We consider both the cases of Toeplitz line networks, where edge weights are specified by three parameters, and line networks with general weights. As a counterintuitive result we prove that, depending on the edge weights, the sensor should be placed as far as possible in the network from the origin of the signal. On the other hand, in certain regions of the parameters space, the sensor should be co–located with the signal to be detected. Our results suggest that sensor location methods based on the network topology alone may lead to poor detection performance in complex cyber-physical systems, due to the intricate relation between the system dynamics and the underlying network structure. The findings are illustrated on a class of electronic circuits.

Chapter 2

General Notions in Algebraic Graph Theory, Dynamic Models and Properties of Networked Systems

In this chapter we review some basic notions in graph theory and present some dynamical models describing the behaviour of networked systems. We also recall the known structural properties of networked control systems, such as controllability and observability, which are used within the attack detection framework as metrics to solve optimisation problems.

2.1 Preliminaries in Graph Theory

Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be a graph, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is the set of n *vertices* and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of *links* (v_i, v_j), also called *edges*. A link (v_i, v_j) is called a *self-loop* if $i = j$. Let us refer to a graph such that all the nodes have a self-loop as a *self-loop graph*. A graph is said to be *undirected* if $(v_i, v_j) \in \mathcal{E}$ whenever $(v_j, v_i) \in \mathcal{E}$, and *directed* otherwise. The *in-degree* of a node is the number of incoming edges, while the out-degree is the number of *out-going edges*; for undirected graphs the in-degree coincides with the out-degree and is referred to as the *degree* of the node. An undirected graph $\mathcal{G}$ is *connected* if for any $v_i, v_j \in V$ with $v_i \neq v_j$ there is a path that connects v_i and v_j. A graph $\mathcal{G}$ is *complete* if for any $v_i, v_j \in V$ s.t. $v_i \neq v_j$ the link $(v_i, v_j) \in \mathcal{E}$. A self-loop undirected graph $\mathcal{G}$ has an *m-path* if there is a path connecting m consecutive nodes, each with exactly degree m, and such path contains a node that is connected to each others node in the path (see Fig. 2.1). Let a unit segment graph be a graph embedded in $\mathbb{R}$ s.t. a couple of nodes are connected whenever their distance is smaller than a given threshold ρ. Note that unit segment graphs are the mono-dimensional analogue of unit disk graphs [3].

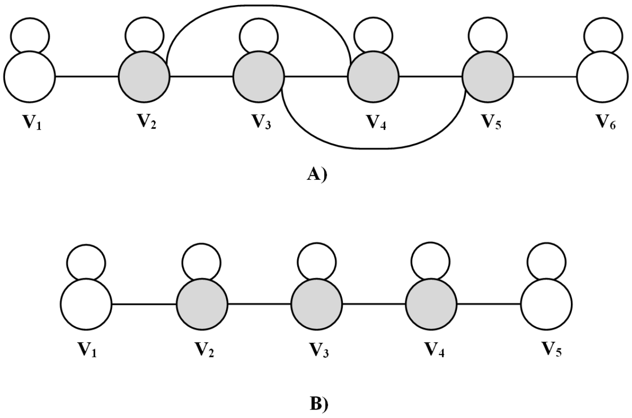


Figure 2.1: Example of two self-loop undirected graphs with m-paths: A) graph with a 4-path where each node in grey colour (V_2,V_3,V_4,V_5) has got four links with the other nodes, *B*) graph with five nodes and a 3-path.

As reported in [8], let us give some other graph definitions. A *cycle* is a closed path in which all nodes and all edges are distinct. With the term *neighbours* it is indicated the nearest of a node i, and they are the nodes to which it is connected directly by an edge, so the number of nearest neighbours of the node is equal to the node degree. A node is said to be *reachable* from another node if there exists a path connecting the two nodes, even if it goes through multiple nodes in between. The *shortest path length* is defined as the length of the shortest path going from nodes i to j. The *diameter* of network is defined as the maximum shortest path length in the network. That is, the diameter is the longest of all shortest paths among all possible node pairs in a graph. The *size* of a network is the average shortest path length defined as the average value over all the possible pairs of nodes in the network. The *density* of a graph is defined as the ratio of the number of edges in the graph to the square of the total number of nodes.

In the following, matrices are represented by uppercase literals (e.g., Γ) while their coefficients are represented by lower case literals (e.g., γ_{ij}). A $n \times n$ matrix Γ and a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with n nodes are *associated* if the coefficient $\gamma_{ij} > 0$ whenever $(v_i, v_j) \in \mathcal{E}$, and $\gamma_{ij} = 0$ otherwise. A matrix Γ whose rows sum all to one is said to be *row stochastic*, while it is said to be *column stochastic* when all the columns sum to one. If Γ is both *row stochastic* and *column stochastic* then it is called *double stochastic*.

2.2 Network Graph Topologies

In this section we report the main network topologies arising in many context of real applications. Generally, in communication networks and computer science [50, 81], a physical topology is a schematic description of the network layout, which includes nodes (vertices) and edges lines (links). With reference to Fig. 2.2, the most widespread topologies are the following:

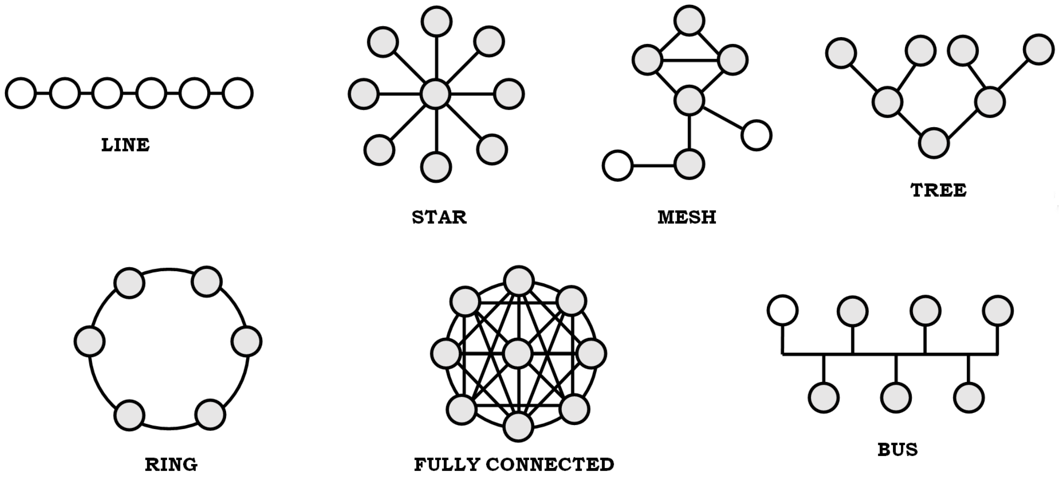


Figure 2.2: Different topologies of communication networks

- In the *line* network topology, the nodes are arranged in series each of which is linked to the predecessor and the followers, except for the first and the last node of the line.

- In the *ring* network topology, the nodes are connected in a closed loop configuration. Adjacent pairs of nodes are directly connected, the data flow passing through one or more intermediate nodes.

- In the *star* network topology, there is a central node to which all nodes are directly connected. Every node is indirectly connected to every other through the central node.

- In the *bus* network topology, every node is connected to a main cable called the bus. Therefore, in effect, each node is directly connected to every other node in the network.

- The *tree* network topology uses two or more star networks connected together. The central node of the star networks are connected to a main bus. Thus, a tree network is a bus network of star networks.

- In the *mesh* network topology the nodes are connected among them according to an undefined rule. In this case, different paths of graph can be determined.

- In the *fully connected* network topology each node is connected directly to each of the others.

2.3 Consensus in Multi-Agent Systems

Consensus issues in multi-agent systems have attracted a lot of interest in the last years [54, 80]. In such multi-agent systems, starting from some initial value, the agents communicate with each other, and tend to modify their values so that the difference with their neighbours decrease. These communications often take the form of agents averaging other agents values to update theirs. Such a system may reach or converge to consensus, that is, a situation where all agents hold the same value. The nature of the values on which consensus is sought, the way averages are performed and the communications that take place are part of each particular system definition. Many properties of these systems depend on their communication topology, which is usually represented by a sequence of graphs. The term *consensus* is usually indicated to describe the diffusion process taking place on the network that leads to the harmonization of the node initial conditions with their asymptotic evolutions converging onto a common equilibrium [22].

2.3.1 Consensus model for continuous-time systems

The interaction topology of a network of agents is represented using a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the set of nodes $\mathcal{V} = \{1, 2, \ldots, n\}$ and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. $\mathcal{G}$ is a graph with a non negative adjacency matrix $A = [a_{ij}]$ that specifies the interconnection topology of a network of dynamic systems, sensors, or agents [72]. The neighbours of agent i are denoted by the set $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$. A simple consensus algorithm to reach an agreement can be expressed in the form of a linear system:

$$\dot{x}_i(t) = \sum_{j \in \mathcal{N}_i} a_{ij}(x_j(t) - x_i(t)), \quad x(0) = c \in \mathbb{R}^n. \tag{2.1}$$

Given a connected network $\mathcal{G}$, all the solutions of system (2.1) converge to an aligned state $x^* = (\mu, \mu, ..., \mu)^T$ with identical elements equal to $\mu = x(0) = \frac{1}{n} \sum_i c_i$. In a more compact form, system (1) can be expressed as

$$\dot{x} = -L\,x \tag{2.2}$$

where L is the Laplacian Matrix of graph $\mathcal{G}$, defined as follows

$$L = \Delta - A, \tag{2.3}$$

where $\Delta = diag(\mathcal{A} \cdot 1)$ is the degree matrix of G with diagonal elements $d_i = \sum_j a_{ij}$. The vector $1 = (1, 1, ..., 1)^T \in \mathbb{R}^n$ denotes the vector of ones that is always a right eigenvector of L corresponding to $\lambda_1 = 0$. Instead, the second smallest eigenvalue λ_2 of L determines the speed of convergence of the algorithm.

2.3.2 Consensus algorithm for discrete-time systems

In this section we review the consensus algorithm for discrete-time systems, with a briefly focus on the: χ-consensus, *max*-consensus, and *average*-consensus problem .

In multi–agent systems each agent updates his current state based upon the information received from other agents. In the study of this system, the communication topology of information flow can be well represented by a weighted directed graph $\mathcal{G}(A)$ [98]. If agent i can receive information from agent j, then there exists an edge from vertex i to vertex j. Therefore, the neighbours of agent i just correspond to the set of neighbours $\mathcal{N}(v_i)$. Let $x_i(0) \in \mathbb{R}$ denote the value of vertex v_i (of agent i). Suppose every node i of $\mathcal{G}(A)$ at each iteration t, updates its own state as

$$x_i(t+1) = A\,x(t) \tag{2.4}$$

where $A \in \mathbb{R}^{n \times n}$ and $x = (x_1, x_2, \ldots, x_n)^T$, which is a discrete-time system. We say that the vertices v_i and v_j agree if and only if $x_i = x_j$, thus the vertices of $\mathcal{G}(A)$ have reach a consensus if and only if $x_i = x_j$ for all $i, j \in n$.

Eq.(2.4) can be expressed also as:

$$x_i(t+1) = \mathcal{U}_i(\{x_j(t) : v_j \in \mathcal{N}_i^{in} \cup \{i\}\}) \tag{2.5}$$

where $\mathcal{U}_i$ is a function of the current state of the node v_i and its in-neighbourhood. Let $\chi(x_1(0), \ldots, x_n(0)) \in \mathbb{R}$ be any function of the initial conditions of all the nodes the χ-*consensus problem* consists in finding a function $\mathcal{U}_i(\cdot)$, such that:

$$\lim_{t \to \infty} x_i(t) = \chi(x_1(0), \ldots, x_n(0)) \quad \forall i = 1, \ldots, n.$$

Let us now discuss the max-consensus problem, where the nodes are required to converge to the maximum of the initial conditions, i.e., $\chi(\cdot)$ is the maximum of its arguments.

2.3.3 Max-consensus

Assuming the graph is connected and undirected, the problem is known to have a solution in finite time [71] (and specifically in no more than n steps) by choosing

$$\mathcal{U}_i(\cdot) = \max_{h \in \mathcal{N}_i^{in} \cup \{i\}} x_h(t). \tag{2.6}$$

In the following we will denote by

$$\overline{x}_i = \text{max-consensus}_i(x_i(0), x_j(0) |\; j \neq i, \mathcal{G}, t_{\max}),$$

the execution of $t_{\max}$ iterations of the max-consensus procedure by the i-th agent in a network $\mathcal{G}$, starting from its own initial condition $x_i(0)$ and the "unknown" initial conditions of the other agents, while $\overline{x}_i$ is the state of the i-th agent at iteration $t_{\max}$. Such a formalism just represents the execution of the max-consensus or min-consensus by the i-th agent, and we assume that all other agents are executing the same algorithm in a synchronous manner, each with its own initial condition.

2.3.4 Average-consensus

In the *average*-consensus problem the nodes are required to converge to the average of their initial conditions, i.e.,

$$\chi(\cdot) = c^T \begin{bmatrix} x_1(0) & \dots & x_n(0) \end{bmatrix}^T$$

where $c^T = \frac{1}{n} 1_n^T$ and 1_n is a vector with n components all equal to 1. Let each node choose

$$\mathcal{U}_i(\cdot) = w_{ii} x_i(t) + \sum_{j=1}^{n} w_{ij} x_j(t) \tag{2.7}$$

where $w_{ij} = 0$ if $(v_i, v_j) \notin \mathcal{E}$. The update strategy for the entire system can be represented as

$$x(t+1) = W x(t),$$

where the $n \times n$ matrix W contains the terms w_{ij}.

According to [99], this choice of $\mathcal{U}_i(\cdot)$ yields an asymptotical solution if and only if : (I) W has a simple eigenvalue at 1 and all other eigenvalues have magnitude strictly less than 1; (II) the left and right eigenvectors of W corresponding to eigenvalue 1 are 1_n and c^T, respectively. The above condition implies that, if the underlying graph is undirected, it needs to be connected. A possible choice for W, assuming that each node knows n (or an upper bound for n), is that each node i chooses independently the terms w_{ij} as

$$w_{ij} = \begin{cases} \frac{1}{n}, & \text{if } v_j \in \mathcal{N}_i^{in} \\ 0, & \text{if } v_j \notin \mathcal{N}_i^{in} \\ 1 - \sum_{l \in \mathcal{N}_l^{in}} w_{il}, & \text{if } i = j \end{cases}$$

resulting in a matrix W that satisfies the conditions in [99]. Several other choices that yield to asymptotic consensus are possible (e.g., see [80]). As for the complexity, note that at each time step t each agent i calculates the contribution of each neighbour to the next state, hence we have $O(d\,n\,t_{\max})$. Notice that, however, the algorithm in its basic setting has asymptotic convergence, hence typically $t_{\max} \gg 1$ in order to obtain a good approximation of the asymptotic result. An alternative is to resort to finite-time average consensus algorithms like the one in [89] but we choose to omit the discussion for the sake of clarity. In the following we will denote by

$$\overline{x}_i = \text{average-consensus}_i(x_i(0), x_j(0)|\; j \neq i, \mathcal{G}, t_{\max}),$$

the execution of $t_{\max}$ iterations of the average-consensus procedure by the i-th agent in a network $\mathcal{G}$, starting from its own initial condition $x_i(0)$ and the "unknown" initial conditions of the other agents, while $\overline{x}_i$ is the state of the i-th agent at iteration $t_{\max}$. Again, such a formalism just represents the execution of the max-consensus or min-consensus by the i-th agent, and we assume that all other agents are executing the same algorithm in a synchronous manner, each with its own initial condition.

2.3.5 Network size calculation

Combining the max-consensus and the average consensus algorithms, it is possible to calculate the number of agents n in the network in a distributed way [84].

Specifically, suppose a *leader* is elected via max-consensus over $\mathcal{G}$ (e.g., the nodes each have a unique identifier and the node with maximum identifier is elected as leader via max-consensus). Now, let the nodes execute an average consensus algorithm with $\bar{x}_i(0) = 1$ if node v_i is the leader and $\bar{x}_i(0) = 0$ otherwise: the average-consensus yields

$$\lim_{t\to\infty} \bar{x}_i(t) = \frac{1}{n}$$

hence n can be calculated in a distributed way.

2.4 The Hegselmann-Krause Opinion Dynamics Model

The Hegselmann and Krause opinion dynamic model was designed to analyse the evolution and diffusion of opinions in a population of human beings [41]. The model represents social influence mechanisms in several contexts, such as political voting, commercial preferences, and social networks [61]. The main feature of this model is that agents with different opinions do not influence each others, while agents interact, influencing each other, if the opinions are close enough [42]. Let us consider a system composed of n agents, each characterized by an initial opinion $x_i(0)$ expressed by a real number. It is assumed that the process of opinion formation is discrete time $\mathcal{T} = \{0, 1, 2 \ldots\}$. Let $x_i(t) \in \mathbb{R}$ be the opinion of i–th agent at time instant t, the vector $x(t) = [x_1(t), \ldots, x_n(t)]^T$ in n dimensional space represents the *opinion profile* at the fixed time instant t. The i–th agent is influenced by opinions that differ from its own one no more than an influence parameter $\varepsilon \geq 0$. Fixing an agent i at generic time t, it can be define the set of *neighbours* of i, depending on $x(t)$ and ε, as:

$$\mathcal{N}_i(x(t), \varepsilon) = \{j \in \{1, \ldots, n\} : |x_i(t) - x_j(t)| \leq \varepsilon\} \tag{2.8}$$

The set $\mathcal{N}_i(x(t), \varepsilon)$ is always non-empty since it always contains the i-th agent itself. Indeed, in the HK model each agent takes into account also its current opinion to form a new one. The opinion of the i-th agent at time instant $t + 1$ is given by the average of the opinions of its neighbours, according to the iterative rule:

$$x_i(t+1) = \frac{1}{\#(\mathcal{N}_i(x(t), \varepsilon))} \sum_{j \in \mathcal{N}_i(x(t), \varepsilon)} x_j(t) \tag{2.9}$$

where $\#(\mathcal{N}_i(x(t), \varepsilon))$ is the cardinality of $\mathcal{N}_i(x(t), \varepsilon)$. The model (2) can be rewritten in compact form, as:

$$x(t+1) = A(x(t), \varepsilon)\, x(t), \quad x(0) = x_0 \tag{2.10}$$

where $A(x(t), \varepsilon)$ is a state-dependent $n \times n$ dynamic matrix whose entries are in the form:

$$a_{ij}(x(t), \varepsilon) = \begin{cases} \frac{1}{\#(\mathcal{N}_i(x(t),\varepsilon))} & \text{if } j \in \mathcal{N}_i(x(t), \varepsilon) \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

$A(x(t), \varepsilon)$ is a row stochastic matrix, that is, all elements of each rows sum to one. The entries of $A(x(t), \varepsilon)$ are either zero or are in the form $1/m$ (for $m = 1, \ldots, n$), and all the nonzero elements of a row are equal.

For every time instant t an undirected self-loop graph $\mathcal{G}(x(t), \varepsilon)$ can be associated to the matrix $A(x(t), \varepsilon)$.

In the following, where it is understood, we will refer to $A(x(t), \varepsilon)$, $\mathcal{G}(x(t), \varepsilon)$ and $\mathcal{N}_i(x(t), \varepsilon)$ as $A(t)$, $\mathcal{G}(t)$ and $\mathcal{N}_i(t)$, respectively.

Notice that the discrete model (2.10) has a high degree of discontinuity. As an example, Fig. 2.3 shows how a small variation in the magnitude of ε, of a quantity equal to 10^{-5}, leads to quite different evolutions for the state of the agents, due to different evolutions of the coefficient $a_{56}(\cdot)$.

Several works can be found in the literature attempting to characterise the properties of the HK model. Given the complexity of the HK model, most of the recently studies assume simple *initial opinion profiles*, i.e., the initial condition $x(0)$ and symmetric or asymmetric confidence intervals. Mainly, within the interval $[0, 1]$ two different classes are considered in the literature [41, 68]: the *deterministic equidistant profile*, where $x_i(0) = \frac{i-1}{n-1}$, and the *random profile* where opinions are uniformly distributed.

Fig. 2.4 shows an example of result of the HK model with $n = 100$ agents and initial

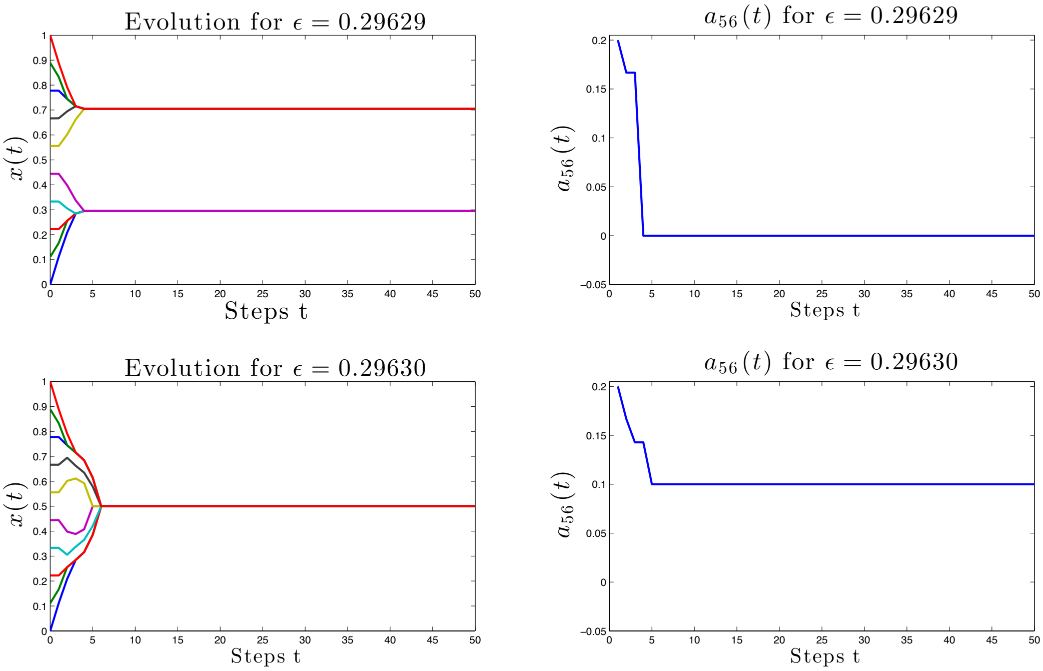


Figure 2.3: Example of completely different evolutions of the HK model for two extremely close values of ϵ (for $n = 10$ agents with deterministic equidistant initial opinion profile in $[0, 1]$ and ε in the two examples that differs of a factor 10^{-5}).

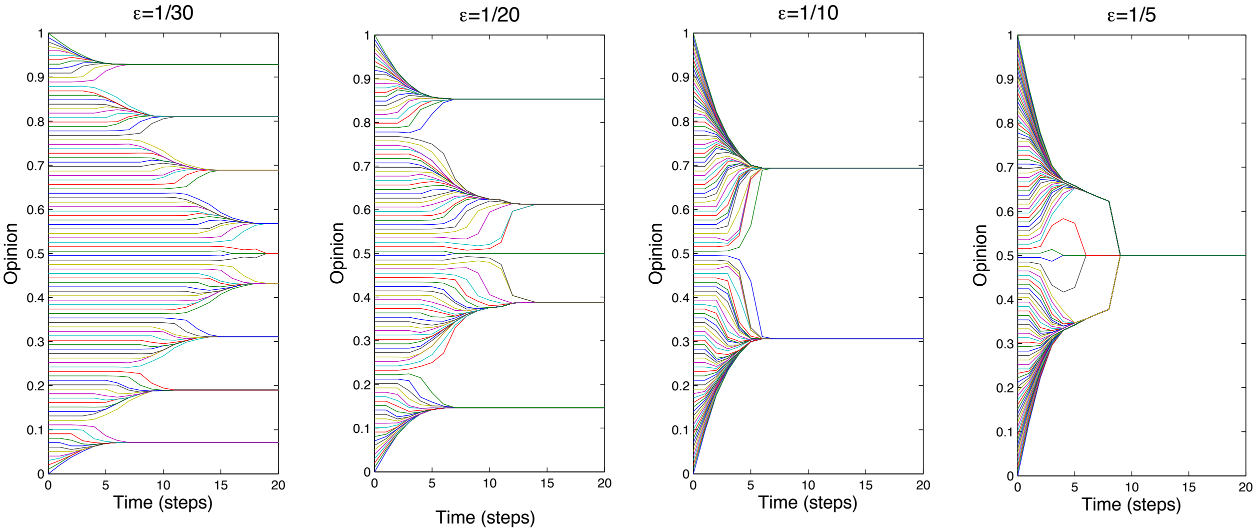


Figure 2.4: Simulation of HK model for $n = 100$ agents with deterministic equidistant initial opinion profile in $[0, 1]$ and for different values of ε.

opinions equidistantly distributed within $[0,1]$, plotted for several choices of the parameter ε. It is noteworthy that the number of clusters decreases when ε grows; for sufficiently large ε, the opinion of all the agents reach an agreement (i.e., a unique shared opinion for all the agents). In [41] it is conjectured that for every ε there must be a number of agents n s.t. the equidistant profile leads to an agreement, while in [68] it is shown that for any initial opinion profile, there exists a finite time $\bar{t}$ after which the graph $\mathcal{G}(t)$ underlying the matrix $A(t)$ remains fixed, i.e., $\mathcal{G}(\bar{t}) = \mathcal{G}(t)$ for all $t \geq \bar{t}$. In [4] it is proven that, during the evolution of the system, the order of the opinions is preserved, that is $x_i(0) \leq x_j(0) \Rightarrow x_i(t) \leq x_j(t)$ for all t. It is also proved that, if the initial opinion profile is sorted in ascending order, then the evolutions of the smallest opinion $x_1(t)$ and of the largest opinion $x_n(t)$ are non-decreasing and non-increasing, respectively. Moreover, if at any step $\bar{t}$ it holds $|x_i(\bar{t}) - x_{i+1}(\bar{t})| > \varepsilon$, this remains true for any $t \geq \bar{t}$, and the system splits into at least two independent subsystems and the graph $\mathcal{G}(x(t), \varepsilon)$ has become disconnected. As a consequence, for any time instant t, an agreement is possible if and only if

$$\min_{i=1,\ldots,n-1} \{|x_i(t) - x_{i+1}(t)|\} \leq \varepsilon$$

In [4, 26, 59] the convergence of the HK model is investigated. In [4], it is proved that the system convergences in finite time to a *steady state*. More precisely, a steady state is reached if there is a time instant t^* s.t. for all $t \geq t^*$ it holds $x(t) = x(t^*)$. However, the convergence to a single or different clusters is still under investigation. Recently, in [101] for a smoothed HK model, a sufficient condition on the initial states (opinions) s.t. the system will converge to exactly one cluster is presented.

2.5 Structural Properties of Networked Systems

2.5.1 Network model

The control problem of complex networks consists of the selection of a set of control nodes, and the design of a control law to steer the network to a target state [75]. Consider a network represented by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in which the set of nodes $\mathcal{V} = \{1, 2, \dots, n\}$ while the edges set is $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Let $A = [a_{ij}]$ be the weighted adjacency matrix associated to the graph $\mathcal{G}$. Assume that A is diagonalizable, i.e., A admits a basis of eigenvectors. We associate a real value (state) with each node, collect the nodes states into a vector (network state), and define the map $x : \mathbb{R} \to \mathbb{R}^n$ to describe the evolution (network dynamics) of the network state over time. We consider the discrete-time, linear, and time invariant network dynamics described by $\dot{x}(t) = A\,x(t)$. Controllability of the network $\mathcal{G}$ refers to the possibility of steering the network state to an arbitrary configuration by means of external controls. We assume that a set $\mathcal{K} := \{k_1, k_2, \dots, k_n\} \subseteq \mathcal{V}$ of nodes can be independently controlled and we let $B_k := \{e_{k_1}, e_{k_2}, \dots, e_{k_n}\}$ be the input matrix, where denotes the i–th canonical vector of dimension n. The network with control nodes k reads as

$$\dot{x}(t) = A\,x(t) + B_k\,u_k(t) \tag{2.12}$$

where is $u_k : \mathbb{R} \to \mathbb{R}^n$ the control signal injected into the network via nodes k.

2.5.2 Controllability of networked systems

A network is controllable by the set of control nodes $\mathcal{K}$ if and only if for every state $x_f(t) \in \mathbb{R}^n$ there exists an input u_k such that $x_f(t) = x(t)$ with initial condition $x(0) = 0$. Controllability of dynamical systems is a well understood property, and it can be ensured by different structural conditions [57]. For instance, let C_k be the controllability matrix defined as:

$$C_k := [B_k \;\; AB_k \;\; \dots \; A^{n-1}B_k] \tag{2.13}$$

The network system(2.12) is controllable by the nodes k, if and only if, rank $(C_k) = n$. The above notion of controllability is qualitative, and it does not quantify the difficulty of the control task as measured, for instance, by the control energy needed to reach a desired state. As a matter of fact, many controllable networks require very large control energy to reach certain states. To formalize this discussion, define the *controllability Gramian*

$$W_c = \int_0^\infty e^{At}\, B\, B^T\, e^{At}\, dt \tag{2.14}$$

and it is the solution of Lyapunov equation

$$-BB^T = A^T W_c + W_c A \tag{2.15}$$

The controllability Gramian defines an ellipsoid in the state space

$$\mathcal{E}_c(t) = \{x \in \mathbb{R}^n | x^T W_c(t)^{-1} x \leq 1\} \tag{2.16}$$

that contains the set of states reachable in t seconds with one unit or less of input energy. The eigenvectors and corresponding eigenvalues of W_c define the semi-axes and corresponding semi-axis lengths of the ellipsoid. Eigenvectors of W_c associated small eigenvalues (large eigenvalues of W_c^{-1} define directions in the state space that are less controllable (require large input energy to reach), and eigenvectors of W_c associated with large eigenvalues (small eigenvalues of W_c^{-1} define directions in the state space that are more controllable (require small input energy to reach).

An analysis of the eigenvalues and corresponding eigenvectors of the controllability Gramian reveals which directions the system can be easily steered to, and which are more energy–demanding. In particular, the direction of the eigenvector corresponding to an eigenvector zero cannot be reached. In other words a controllable system has a positive definite W_c [87]. It can be verified that the controllability Gramian W_c is positive definite if and only if the network is controllable at the time t by the nodes k. Different quantitative measures of controllability of dynamical systems have been considered in the last years [23]. In addition to the smallest eigenvalue of the controllability Gramian $\lambda_{min}(W_c)$, the trace of the inverse of the controllability Gramian $trace(W_c^{-1})$, and the determinant of the controllability Gramian $det(W_c)$, have been proposed. It can be shown that, while $trace(W_c^{-1})$ measures the average control energy over random target states, $det(W_c)$ is proportional to the volume of the ellipsoid containing the states that can be reached with a unit energy control input. Unlike the controllability metrics $\lambda_{min}(W_c)$, $trace({W_c}^{-1})$, $det(W_c)$ the selection of the control nodes to maximize the $trace(W_c)$ admits a closed-form solution. Unfortunately, the maximization of $trace(W_c)$ does not automatically ensure controllability and, it often leads to a poor selection of the control nodes with respect to the worst-case control energy to reach a target state.

2.5.3 Observability of networked systems

A quantitative description of a complex networked systems is inherently limited by our ability to estimate the systems internal state from experimentally accessible outputs. A system is called *observable* if we can reconstruct the systems complete internal state from its outputs [58]. More precisely, the network system (2.12) is observable if the observability matrix $O_k := [C_k^T \ C_k^T A \ \dots \ C_k^T A_{n-1}]^T$ is full column rank. The observability problem of complex networks consists of selecting a set of sensor nodes, and designing an estimation strategy to reconstruct the network state from measurements collected by the sensor nodes. Equivalently to the controllability case, observability can be analysed by using the *observability Gramian* W_o which is defined as

$$W_o = \int_0^\infty e^{At}\, C^T\, C\, e^{At}\, dt \tag{2.17}$$

The observability Gramian is symmetric and positive semi–definite and it is also the solution of Lyapunov equation:

$$-C^T C = A^T W_o + W_o A \tag{2.18}$$

The observability Gramian is a measure of the energy visible in the output signal when letting the system freely evolve from the initial state at time zero towards the steady state 0. Similarly to the controllability Gramian, the observability Gramian defines an ellipsoid in the state space

$$\mathcal{E}_o(t) = \{x \in \mathbb{R}^n | x^T W_o(t)^{-1} x \leq 1\} \tag{2.19}$$

The eigenvectors and eigenvalues of the observability ellipsoid provide information about the relative degree of observability of different directions in state space.

2.6 Attack Detection and Optimal Sensor Placement in Networked Systems

Controllability and observability have long been recognized as fundamental structural properties of dynamical systems [69], but have recently seen renewed interest in the context of large, complex networks of dynamical systems [88]. A basic problem is the sensor and actuator placement, namely, selecting a subset from a finite set of possible placements to optimize some real-valued controllability and observability metrics of the network [88]. In this section we model complex networked systems under attack as a time-invariant dynamical network models, in which the dynamics are given by subject to unknown inputs.

This modelling framework is very general, it includes many of the existing networked systems and describes various attack scenarios of interest [76]. The network's dynamics is thus described by the the following linear time–invariant dynamical system:

$$\begin{aligned} \dot{x}(t) &= A\,x(t) + B\,u(t), \\ y(t) &= C\,x(t) + D\,u(t), \end{aligned} \tag{2.20}$$

where $x : \mathbb{R} \to \mathbb{R}^n, y : \mathbb{R} \to \mathbb{R}^p,\ A \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times m}, C \in \mathbb{R}^{p\times n}$ and $D \in \mathbb{R}^{p\times m}$. We assume that the vector u models the attacks on the state x, and the vector y models sensor measurements. As discussed before in Section 2.5.1, A is the network matrix, and without affecting generality, a crucial assumption leading to a correct analysis for the system (2.20) is that A is Hurwitz stable [92], i.e., all its eigenvalues are contained in the open left half of the complex plane ($Re[\lambda_i] < 0$). In this description, the states are subjected to attacks through the input matrix B, while the system output is reconstructed by the sensors matrix C, describing where it is best to place available sensors in order to detect possible attacks to the network. Moreover, y can be influenced by unknown signals Du, modelling disturbances affecting the plant. Besides, reflecting the genuine failure of systems components, these disturbances model the effect of an attack against the networked system. For systems with sensors and measurements, the impact of an attack can be marginalized if one can detect the attack. Sensor and actuator placement problems can be formulated as set function optimization problems.

For a given finite set $\mathcal{V} = \{1, \dots, n\}$ a set function $f : 2^{\mathcal{V}} \to \mathbb{R}$ assigns a real number to each subset of $\mathcal{V}$. In our setting, the elements of $\mathcal{V}$ represent potential locations for the placement of sensors or actuators in a dynamical system, and the function f is a metric for how controllable or observable the system is for a given set of placements, which is to be maximized. We consider set function optimization problems of the form:

$$\begin{aligned} &\max && f(\mathcal{S}) \\ &\text{subject to} && |\mathcal{S}| = k,\ \mathcal{S} \subseteq \mathcal{V}. \end{aligned}$$

The problem is to select a k-element subset of $\mathcal{V}$ that maximizes f. This is a finite combinatorial optimization problem, so one way to solve it is by brute force: simply enumerate all possible subsets of size k, evaluate f, and pick the best subset. However, we are interested in cases arising from complex networks in which the number of possible subsets is very large. The number of possible subsets grows extremely fast as $\mathcal{V}$ increases, so the brute force approach quickly becomes infeasible as $\mathcal{V}$ becomes large.

Chapter 3

Distributed Steady-State, Connectedness and Agreement Detection Criteria for the Hegselmann-Krause Model

3.1 Alternative Formulation of the Hegselmann-Krause Model

Let us start this section with an alternative formulation of the HK model presented in Section 2.4, by providing insights on the complexity underlying the model. The update rule of the HK model (2.9) can be rewritten as follows:

$$x_i(t+1) = \frac{\sum_{j=1}^{n} \beta_{ij}(x_i(t), x_j(t), \varepsilon)\, x_j(t)}{\sum_{j=1}^{n} \beta_{ij}(x_i(t), x_j(t), \varepsilon)}\,, \quad i = 1, \ldots, n\,, \tag{3.1}$$

where $\beta_{ij}(x_i(t), x_j(t), \varepsilon) : \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ is a non–negative *weighting function*, describing the belief of the i–th agent about the opinion of the j–th agent, depending also on the parameter ε. Such a weighting function depends on the distance between its two input variables, i.e.,

$$\beta_{ij}(x_i(t), x_j(t), \varepsilon) = \beta_{ij}(|x_j(t) - x_i(t)|, \varepsilon).$$

The stack equation (2.10) of the HK model is obtained for

$\beta_{ij}(|x_i(t) - x_j(t)|, \varepsilon) = sign(sign(\varepsilon - |x_i(t) - x_j(t)|) + 1)$

setting:

$$a_{ij}(x(t), \varepsilon) = \frac{\beta_{ij}(|x_i(t) - x_j(t)|, \varepsilon)}{\sum_{h=1}^{n} \beta_{ih}(|x_i(t) - x_h(t)|, \varepsilon)} \tag{3.2}$$

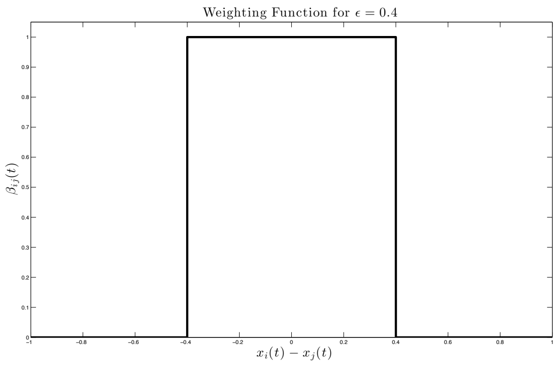


Figure 3.1: Weighting function $\beta_{ij}(|x_i - x_j|, \varepsilon)$ of the HK model for $\varepsilon = 0.4$.

The above formalization provides some insights on the challenging complexity of the HK model. Indeed, the model is intrinsically discontinuous: the agents cease abruptly to influence each other when the distance among the agent's opinion becomes larger than ε, as shown in Fig. 3.1. Notice that, while the term $\beta_{ij}(|x_i(t) - x_j(t)|, \varepsilon)$ only depends on the state of agent i and agent j, $a_{ij}(x(t), \varepsilon)$ depends on the state of all the agents. In the following, where understood, we will refer to $\beta_{ij}(|x_i(t) - x_j(t)|, \varepsilon)$ as $\beta_{ij}(t)$.

3.2 Criteria for Steady State Detection

It has been established in [4] that the HK model converges to a steady state from any initial opinion profile $x(0)$. In this section, we provide conditions, which can be verified in a distributed way, on the state $x(\bar{t})$ of the system at the time $\bar{t}$ ensuring that the system has reached a steady state at time instant $t+1$; such conditions can be used to determine when the HK algorithm can stop.

Let us present the following preliminary result, which it will use as mathematical tool to prove the convergence of the HK model in one step.

Lemma 3.2.1. ***(Unit segment graph)*** *Let the opinion profile $x(\bar{t})$ of an HK model for a given time instant $\bar{t}$. The graph $\mathcal{G}(\bar{t})$ associated to matrix $A(\bar{t})$ is a unit segment graph with $\rho = \varepsilon$.*

Proof. Each agent in the HK model has a scalar opinion $x_i(\bar{t})$ hence the graph $\mathcal{G}(\bar{t})$ is embedded in a mono-dimensional space. The definition of the neighboured (2.8) implies that $\mathcal{G}(\bar{t})$ is a unit segment graph with $\rho = \varepsilon$. □

Instead, the following theorem gives a necessary and sufficient condition to establish the convergence of the HK model.

Theorem 3.2.1. ***(Convergence of the HK model)*** *Given the opinion profile $x(\bar{t})$ of a HK dynamic model at a time instant $\bar{t}$, the dynamic model will converge in one step, i.e., $x(\bar{t}+1)$ is an equilibrium point, if, and only if, the dynamic matrix $A(\bar{t})$ is doubly stochastic.*

Proof. Let us first prove that $A(\bar{t})$ is doubly stochastic if, and only if, $A(\bar{t})$ is similar to a block diagonal matrix, with $\kappa \geq 1$ full blocks, i.e., there exists a permutation matrix P s.t.

$$P^{-1}\,A(\bar{t})\,P = \mathrm{diag}(D_1(\bar{t}), \ldots, D_\kappa(\bar{t}))\,, \tag{3.3}$$

where each sub–matrix $D_i(\bar{t})$ is full.

Suppose matrix $A(\bar{t})$ is symmetric and let P be the permutation matrix that sorts the components of $x(\bar{t})$ in ascending order, i.e., $Px(\bar{t}) = z$ s.t. $z_i \leq z_{i+1}$, for all $i = 1, \ldots, n-1$. Let $\zeta_1 = \#(\mathcal{N}_1(z(\bar{t}), \varepsilon))$ be the size of the neighboured of the agent represented by the variable z_1. Since by Lemma 3.2.1, $\mathcal{G}(\bar{t})$ is a unit segment graph, the neighbours of the agent represented by z_1 are also neighbours of each other, the subgraph underlying the agent represented by z_1 and its neighbours is complete. Matrix $A(\bar{t})$ is symmetric, therefore the first row $row_1(D(\bar{t}))$ and the first column $col_1(D(\bar{t}))$ of matrix $D(\bar{t})$ are given by:

$$row_1(D(\bar{t})) = col_1(D(\bar{t}))^T = \left[\begin{array}{c|c} \frac{1}{\zeta_1} \ldots \frac{1}{\zeta_1} & 0 \ldots 0 \end{array}\right].$$

Since the first ζ_1 elements in $col_1(D(\bar{t}))$ are equal to $\frac{1}{\zeta_1}$, the first ζ_1 agents have exactly ζ_1 neighbours, and thus

$$row_i(D(\bar{t})) = row_1(D(\bar{t})), \quad \forall i = 1, \ldots, \zeta_1$$

hence

$$D(\bar{t}) = \mathrm{diag}(D_1(\bar{t}), D_1^*(\bar{t}))$$

where $D_1(\bar{t})$ is a full matrix and $D_1^*(\bar{t})$ is a $(n-\zeta_1) \times (n-\zeta_1)$ matrix. The proof that $D_1^*(\bar{t})$ is a block diagonal matrix with complete blocks can be obtained by applying the above argument to the agent represented by z_{ζ_1+1}, and so on. Let us now prove that if $A(\bar{t})$ is composed of full block diagonals then $A(\bar{t})$ is column stochastic. Since $A(\bar{t})$ is similar to a full block diagonal matrix, all agents associated with the state variables in the same block have the same number of neighbours. Therefore the elements of each block take the same values and, since the rows sum to one, also the columns sum to one. To complete the proof, let us prove that if $A(\bar{t})$ is column stochastic then it is also symmetric. We can equivalently show that, if $A(\bar{t})$ is asymmetric, there is a j for which $\sum_{i=1}^{n} a_{i,j}(\bar{t}) \neq 1$. Let P

be the permutation matrix that sorts the state variables in ascending order and let ζ_1 be the number of neighbours of agent with smaller opinion. The first row $row_1(D(\bar{t}))$ and the first column $col_1(D(\bar{t}))$ of matrix $D(\bar{t})$ are:

$$\begin{aligned} row_1(D(\bar{t})) &= \left[\begin{array}{c|c} \frac{1}{\zeta_1} \dots \frac{1}{\zeta_1} & 0 \dots 0 \end{array} \right], \\ col_1(D(\bar{t}))^T &= \left[\begin{array}{c|c|c} \frac{1}{\zeta_1} & q_1 \dots q_{\zeta_1 - 1} & 0 \dots 0 \end{array} \right]. \end{aligned} \tag{3.4}$$

Since $A(\bar{t})$ is asymmetric, $D(\bar{t})$ is not a block diagonal matrix with full blocks, hence one of the following holds:

i) $D(\bar{t}) = \text{diag}(D_1(\bar{t}), D_1^*(\bar{t}))$, where $D_1(\bar{t})$ is a complete block and $D_1^*(\bar{t})$ is not a diagonal block matrix composed of full blocks;

ii) there is an index i s.t. q_{i-1} is different from the value of the i–th element in the first row. Moreover, since it must hold that $\frac{1}{n} \leq q_i \leq \frac{1}{\zeta_1}$, for $i = 1, \dots, \zeta_1 - 1$, it must also hold that $q_i < \frac{1}{\zeta_1}$.

Suppose case ii) is verified; the sum of the first column is s.t.

$$\frac{1}{\zeta_1} + \sum_{i=1}^{\zeta_1 - 1} q_i \leq \frac{1}{\zeta_1} + \frac{\zeta_1 - 2}{\zeta_1} + q_i^* < \frac{1}{\zeta_1} + \frac{\zeta_1 - 2}{\zeta_1} + \frac{1}{\zeta_1} = 1$$

hence the statement is proved.

If conversely, case i) is verified, the argument used for case ii) can be applied to the first column of $D_1^*(\bar{t})$ and so on, until a non complete block is found s.t. the sum of the first row is less than 1. We are guaranteed to find at least one of such incomplete blocks because $A(\bar{t})$ is asymmetric.

The above result shows that, when $A(\bar{t})$ is column stochastic, $\mathcal{G}(\bar{t})$ is either a complete graph or is decomposed in complete subgraphs, hence $x(\bar{t}+1)$ is the steady state.

Finally, we need to show that, once the matrix $A(\bar{t})$ has reached the diagonal form in Eq. (3.3), the state of the system will converge to an equilibrium in one step. To prove this, note that, once $A(\bar{t})$ has assumed such a structure, according the HK model, no further links can appear and no existing ones can disappear. Thus it must hold $x_i(\bar{t}^*) = x_i(\bar{t})$, for all $i = 1, \dots, n$, and for all $\bar{t}^* > t$. □

According to the above result, checking for column stochasticity is a stop criterion that can be verified in a centralized way. We now focus on understanding whether the above criterion can be translated into a local criterion, i.e., if the following claim holds true.

Criterion 3.2.1. ***(Distributed steady state achievement criterion)*** *Let the opinion profile $x(\bar{t})$ of a HK model* (2.10) *for a given time instant $\bar{t} \geq 0$. If the i-th column of $A(\bar{t})$ sums to one, then the state $x_i(\bar{t}+1)$ coincides with the steady state of agent i.*

Such a criterion is easily checkable in a distributed fashion if each node i provides to his neighbours at each time step t the value of $\#(\mathcal{N}_i(t))$; knowing such values, each node i is able to calculate sum of the entries of the corresponding column of $A(t)$.

There is, unfortunately, a counterexample to the above claim.

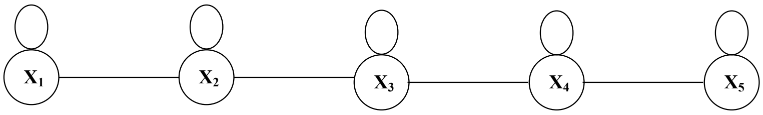


Figure 3.2: Graph topology relatives of the Counterexample 3.2.1 with five-agents system.

Counterexample 3.2.1. *Let us consider a five–agent system, represented by the HK model* (2.10) *for a fixed time instant $\bar{t}$, $\varepsilon = 0.2$ and $x(\bar{t}) = [0.1,\ 0.3,\ 0.5,\ 0.7,\ 0.9]^T$. The graph $\mathcal{G}(\bar{t})$ underlying the HK model at time instant $\bar{t}$ is given in Fig. 3.2, while the corresponding matrix $A(\bar{t})$ is:*

$$A(\bar{t}) = \begin{bmatrix} 1/2 & 1/2 & \mathbf{0} & 0 & 0 \\ 1/3 & 1/3 & \mathbf{1/3} & 0 & 0 \\ 0 & 1/3 & \mathbf{1/3} & 1/3 & 0 \\ 0 & 0 & \mathbf{1/3} & 1/3 & 1/3 \\ 0 & 0 & \mathbf{0} & 1/2 & 1/2 \end{bmatrix}$$

In this case $\mathcal{G}(\bar{t})$ has only non complete components, but there is a column (in boldface) whose sum is 1.

As it will be shown in the following, a case similar to the above counterexample can be obtained if several contiguous agents have the same number of neighbours. Let us characterize the above class of counterexamples, and let us show under which hypotheses Criterion 3.2.1 holds.

Lemma 3.2.2. ***(Equal column elements value)*** *suppose the opinion profile $x(\bar{t})$ of a HK dynamic model is given at a time instant $\bar{t}$. A column of $A(\bar{t})$ sums to* 1 *if, and only if, all the nonzero entries of that column have all the same value.*

Proof. The sufficiency trivially follows by observing that, if all entries of the i–th column are equal, say, to the value of the entry $a_{i,i}(\bar{t})$, they are also equal to the values of the i–th row, which, by the hypothesis of stochasticity of $A(\bar{t})$, also implies that the sum of the entries in the column is one.

To prove the necessity, let us suppose that $A(\bar{t})$ is in the form of Eq. (3.3). By absurd assume that the i–th column of $A(\bar{t})$ sums to one, i.e., $\sum_{j=1}^{n} a_{j,i}(\bar{t}) = 1$, while its nonzero entries are not equal. We need to show that in this case the number of nonzero entries of column i is different from the number of nonzero entries of row i, which is absurd because $\mathcal{G}(\bar{t})$ is symmetric.

Let the i-th column of $A(\bar{t})$ and suppose that it sums to one and that the nonzero entries are not all equal; the i-th row has exactly $\frac{1}{a_{i,i}(\bar{t})}$ nonzero elements, all equal to

$a_{i,i}(\bar{t})$. The i-th column does not have all nonzero entries equal to $a_{i,i}(\bar{t})$ for hypothesis, hence there is at least a nonzero element which is not equal to $a_{i,i}(\bar{t})$. Without loss of generality, suppose that the indices of the nonzero elements of column i are $1, \dots, 1/a_{i,i}(\bar{t})$. The sum of the element of the column is such that

$$\sum_{j=1, j\neq i}^{1/a_{i,i}(\bar{t})} a_{j,i}(\bar{t}) = 1 - a_{i,i}(\bar{t})$$

with the constraint that each $a_{j,i}(\bar{t})$ is positive and rational in the form $1/m$ with at least one $m \neq 1/a_{i,i}(\bar{t})$. It is easy to see that no choice of $a_{j,i}(\bar{t})$ satisfies the above constraints. We can therefore conclude that either the coefficients are all equal or the number of nonzero entries of the i-th row and column are different; both cases are absurd. □

Let us provide an example explaining Lemma 3.2.2.

Example 3.2.1. *The following matrix $A(\bar{t})$ has the first column that sums to one, but the nonzero elements are not all equal.*

$$A(\bar{t}) = \begin{bmatrix} 1/3 & 1/3 & 1/3 & \mathbf{0} & 0 & 0 & \dots \\ 1/3 & 1/3 & 1/3 & \mathbf{0} & 0 & 0 & \dots \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & \dots \\ \mathbf{1/6} & \mathbf{1/6} & 1/6 & 1/6 & 1/6 & 1/6 & \dots \\ 0 & 0 & 0 & \vdots & \vdots & \vdots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots \\ 0 & 0 & 0 & \vdots & \vdots & \vdots & \dots \end{bmatrix}$$

It is easy to show that matrix $A(\bar{t})$ does not represent a valid HK model: in fact, the first two elements of the 4*-th row (in boldface) are positive, while the first two elements of the* 4*-th column (in boldface) are equal to zero, hence the graph topology underlying the system is not undirected, as required for valid HK models.*

Let us state the following Theorem.

Theorem 3.2.2. ***(Convergence to a steady state)*** *Let the opinion profile $x(\bar{t})$ of an HK model for a given time instant $\bar{t}$. Suppose graph $\mathcal{G}(\bar{t})$ associated to matrix $A(\bar{t})$ is decomposed in $\kappa \geq 1$ subgraphs, and that an agent i belongs to a subgraph $\mathcal{G}_j(\bar{t})$ of $\mathcal{G}(\bar{t})$ with κ_j nodes. The i-th column of matrix $A(t)$ sums to one if and only if either one of the following propositions holds true:*

1. *$\mathcal{G}_j(\bar{t})$ is complete;*

2. *$\mathcal{G}_j(\bar{t})$ contains a m-path with $2 < m < \kappa_j$.*

Proof. The proof of necessity is trivial hence we will only prove sufficiency.

Without loss of generality, let us suppose that $A(\bar{t})$ is in the form of eq. (3.3). Suppose the i-th column of matrix $A(t)$ sums to one but $\mathcal{G}_j(x(\bar{t}), \varepsilon)$ does not contains a m-path nor $\mathcal{G}_j(x(\bar{t}), \varepsilon)$ is complete. By Lemma 3.2.2, the only way to obtain a column whose elements sum to one is that all the nonzero elements of the column take on the same value $1/\alpha$, with $\alpha \in \mathbb{N}$. Now suppose that the number of such elements is m. The sum of the column is thus $\frac{m}{\alpha}$ which must equal 1 by hypothesis. Hence, it must be $m = \alpha$. If the m entries are consecutive then there exists a path of length m, whose nodes have exactly $\alpha = m$ neighbours, and such a path has a node connected to each node in the path, that is $\mathcal{G}_j(x(\bar{t}), \varepsilon)$ has a m-path or is complete.

□

According to the above result, therefore, there is an intrinsic ambiguity while attempting to evaluate locally the convergence to a steady state by inspecting the column sums, because the criterion provided in Claim 3.2.1 does not hold in the presence of m-paths.

It is however possible to provide a distributed procedure to detect the convergence to a steady state, which is a corollary of Theorem 3.2.1 and Theorem 3.2.2.

Corollary 3.2.1. ***(Steady state criterion)*** *Let the opinion profile $x(\bar{t})$ of an HK model for a given time instant $\bar{t}$. The state $x_i(\bar{t}+1)$ of an agent i coincides with its steady state if and only if $\sum_{h=1}^{n} a_{jh}(\bar{t}) = 1$ for all $j \in \mathcal{N}_i(\bar{t})$.*

The above corollary can be used to define a distributed check for the convergence to the steady state: as a first step each node i provides to his neighbours the value of $\#(\mathcal{N}_i(t))$ of its neighbourhood; then, each node i calculates the sum of the corresponding column, and provides such value to the neighbours, hence the nodes are able to check whether the column associated to all their neighbours (including themselves) sums to one. Such a two-step procedure can be alternated with the execution of the HK model.

Notice that the previous results provide information on the convergence to a steady state but do not imply the convergence to an agreement or the fact the graph is connected; to this end we provide some conditions in the next Section.

3.3 Connectedness of Graph Network

In this section we provide some centralized and distributed conditions to detect whether the system has reached an agreement, and whether the graph network $\mathcal{G}$ at time instant $t+1$ is connected, given the state $x(t)$ and ε. Let us state the following Theorem.

Theorem 3.3.1. ***(Connectedness of graph)*** *Let the HK model given in eq.* (2.10) *and let an $(n-1)\times(n-1)$ matrix $\hat{L}^*(t)$ s.t.*

$$\hat{l}^*_{ij}(t) = \sum_{h=1}^{n} \hat{\beta}_{i+1,h}(t) - \hat{\beta}_{1j}(t), \quad i,j = 1,\ldots,n-1 \tag{3.5}$$

where

$$\hat{\beta}_{ij}(t) = \beta(|\sum_{h=1}^{n}(a_{ih}(t) - a_{jh}(t))x_h(t)|, \varepsilon). \tag{3.6}$$

The graph $\mathcal{G}(t+1)$ is connected if, and only if, the matrix $L^*(t)$ is invertible.

Proof. Let us consider matrix $L(t) = N(t)(I - A(t))$, where $N(t) = \text{diag}(\mathcal{N}_1(t) \ldots, \mathcal{N}_n(t))$.

It is easy to verify that it holds

$$l_{ij}(t) = \begin{cases} \sum_{h=1}^{n} \beta_{ih}(t) & \text{if } i = j \\ \\ -\beta_{ij}(t) & \text{if } i \neq j \end{cases} \tag{3.7}$$

therefore $L(t)$ is a Laplacian matrix. Such a matrix has only nonnegative eigenvalues and the multiplicity of the eigenvalue $\lambda_1 = 0$ coincides with the number of connected components of the graph $\mathcal{G}(t)$ (see, for instance, [97]). To prove that graph $\mathcal{G}(t)$ is connected, we need to show that $L(t)$ has a simple eigenvalue $\lambda_1 = 0$.

To this end, let us consider the following transformation matrix:

$$T = \left[\begin{array}{c|c} 1 & 1_{n-1}^T \\ \hline 1_{n-1} & -I_{n-1} \end{array}\right]$$

where 1_{n-1} is a vector with $n-1$ components, all equal to one and I_{n-1} is the $(n-1) \times (n-1)$ identity matrix.

It holds that $L(t)$ has a simple eigenvalue $\lambda_1 = 0$ if and only if $TL(t)T^{-1}$ has a simple eigenvalue $\lambda_1 = 0$.

Since

$$T^{-1} = \left[\begin{array}{c|c} \frac{1}{n} & \frac{1}{n}1_{n-1}^T \\ \hline \frac{1}{n}1_{n-1} & \frac{1}{n}1_{n-1}1_{n-1}^T - I_{n-1} \end{array}\right]$$

it is possible to show, by some algebra, that matrix $TL(t)T^{-1}$ has the following structure

$$TL(t)T^{-1} = \left[\begin{array}{c|c} 0 & 0_{n-1}^T \\ \hline 0_{n-1} & L^*(t) \end{array}\right]$$

where, by some algebra

$$L^*(t) = \begin{bmatrix} l_{22}(t) - l_{12}(t) & \cdots & l_{2n}(t) - l_{1n}(t) \\ \vdots & \ddots & \vdots \\ l_{n2}(t) - l_{12}(t) & \cdots & l_{nn}(t) - l_{1n}(t) \end{bmatrix}.$$

Matrix $TL(t)T^{-1}$ is block diagonal with a scalar block equal to 0, hence $\mathcal{G}(t)$ is connected if and only if $L^*(t)$ has no eigenvalue equal to 0.

Since the elements $l^*_{ij}(t)$ are given by

$$l^*_{ij}(t) = \sum_{h=1}^{n} \beta_{i+1,h}(t) - \beta_{1j}(t), \quad i, j = 1, \ldots, n-1.$$

we can conclude that $\mathcal{G}(t+1)$ is connected if and only if $\hat{L}^*(t)$ is invertible, and the proof is complete. □

3.4 Agreement Detection Conditions

According to Theorem 3.3.1, we can provide the following agreement condition.

Corollary 3.4.1. ***(Agreement condition achievement)*** *Let the HK model given in eq.* (2.10) *and suppose $\mathcal{G}(0)$ is connected. The HK model reaches an agreement if and only if for each time step t (until the steady state is reached) matrix $L^*(t)$ is invertible.*

Let us first provide a sufficient condition for the convergence to an agreement point:

Proposition 3.4.1. ***(Convergence to an agreement point)*** *Let the HK model given in eq.* (2.10) *and let*

$$\begin{aligned} \psi_i(x(t),\varepsilon) \quad = \quad & \textstyle\sum_{j=1}^{n} \hat{\beta}_{i+1,i+1}(t) + \hat{\beta}_{1,i+1}(t) + \\ & - \textstyle\sum_{j=2,j\neq i+1}^{n} |\hat{\beta}_{1j}(t) - \hat{\beta}_{i+1,j}(t)| + \\ & -|\hat{\beta}_{1,i+1}(t) - \textstyle\sum_{j=1}^{n} \hat{\beta}_{i+1,i+1}(t)|. \end{aligned} \tag{3.8}$$

where $i = 1, \ldots, n-1$.

If the state $x(t)$ and ε satisfy

$$\Psi(x(t),\varepsilon) = \min_{i=1,\ldots,n-1} \{\psi_i(x(t),\varepsilon)\} > 0 \tag{3.9}$$

then $\mathcal{G}(t+1)$ is connected.

Proof. By Gershgorin Circle's Theorem [37] the matrix $L^*(t)$ is invertible if $c_i - r_i > 0$ for all $i = 1, \ldots, n-1$, where

$$c_i := l^*_{ii}(t) = l_{i+1,i+1}(t) - l_{1,i+1}(t)$$

and

$$r_i := \sum_{j=1}^{n-1} |l_{ij}^*| = \sum_{j=2}^{n} |l_{i+1,j}(t) - l_{1j}(t)|.$$

In other words, $\hat{L}^*(t)$ is invertible if

$$l_{i+1,i+1}(t) - l_{1,i+1}(t) - \sum_{j=2}^{n} |l_{i+1,j}(t) - l_{1j}(t)| > 0\,,$$

and thus $\mathcal{G}(t+1)$ is connected if it holds

$$\hat{l}_{i+1,i+1}(t) - \hat{l}_{1,i+1}(t) - \sum_{j=2}^{n} |\hat{l}_{i+1,j}(t) - \hat{l}_{1j}(t)| > 0$$

or, by some algebra, if condition (3.9) is verified. □

By means of Proposition 3.4.1, we are provided with a function $\Psi(x(t), \varepsilon)$ that, if positive for a given time instant t, guarantees the graph topology of the HK model to be connected at time instant $t+1$.

Let us conclude the section providing a distributed way to detect at time instant t the convergence to an agreement at time instant $t+1$.

Proposition 3.4.2. ***(Distributed convergence criterion)*** *Let the HK model given in eq.* (2.10)*. An agreement is reached if and only if there is a time instant $\bar{t}$ s.t. a coefficient $a_{ij}(t) = 1/n$ and $\sum_{h=1}^{n} a_{hi}(t) = 1$. Such an agreement is reached at time instant $t+1$.*

Proof. If there is a coefficient $a_{ij}(t) = 1/n$ we can conclude that all the coefficients on the i-th row of $A(t)$ are equal to $1/n$ and that all the coefficients on the j-th column are nonzero, because the graph $\mathcal{G}(t)$ is undirected. Notice that this does not imply $\mathcal{G}(t)$ is complete, because the fact an agent is connected to each other agent does not imply each agent is connected with each other. If the i-th column sums to one, then for Theorem 3.2.2 the agent i belongs either to a connected component or to an m-path, but since

$a_{ij}(t) = 1/n$ we must conclude $\mathcal{G}(t)$ is complete, and from Corollary 3.2.1 a steady state is reached in time instant $t+1$. Such a steady state is the same for all the agents, because the graph $\mathcal{G}(t)$ is complete. □

Proposition 3.4.2 provides a distributed criterion for the verification of the convergence to an agreement. By knowing the value of n, each agent i is able to verify the convergence to an agreement by calculating the column sums of each node shares $\#(\mathcal{N}_i)$ with its neighbours and checking, for instance, if $a_{ii} = 1/n$. If, conversely, the value of n is not known, it is possible to resort to distributed algorithms such as the approach in [84] to calculate n based on distributed consensus algorithms. It can be noted that, although distributed, in the case an agreement is reached all agents will detect it at the same time instant t.

3.5 Illustrative Examples

In this section we provide an example of application of the proposed centralized criteria to detect at time instant t the convergence to a steady state in $t+1$ and to check the connectedness of the graph. Furthermore, we propose an example of application of the distributed criteria to detect the convergence to a steady state (Corollary 3.2.1) and an agreement (Proposition 3.4.2).

3.5.1 Numeric applications of centralized criteria

In Fig. 3.3, we report a simulation of the HK model for $n = 100$ agents with deterministic equidistant initial opinion profile. Column (a) and (b) and (c) show the results for $\varepsilon = 0.6$, $\varepsilon = 0.25$ and $\varepsilon = 0.15$, respectively. The upper plots show the evolution for the agents; the lower plots show the application of the criterion defined in Theorem 3.2.1 to detect the convergence to a steady state (red triangles facing downwards), of the criterion introduced

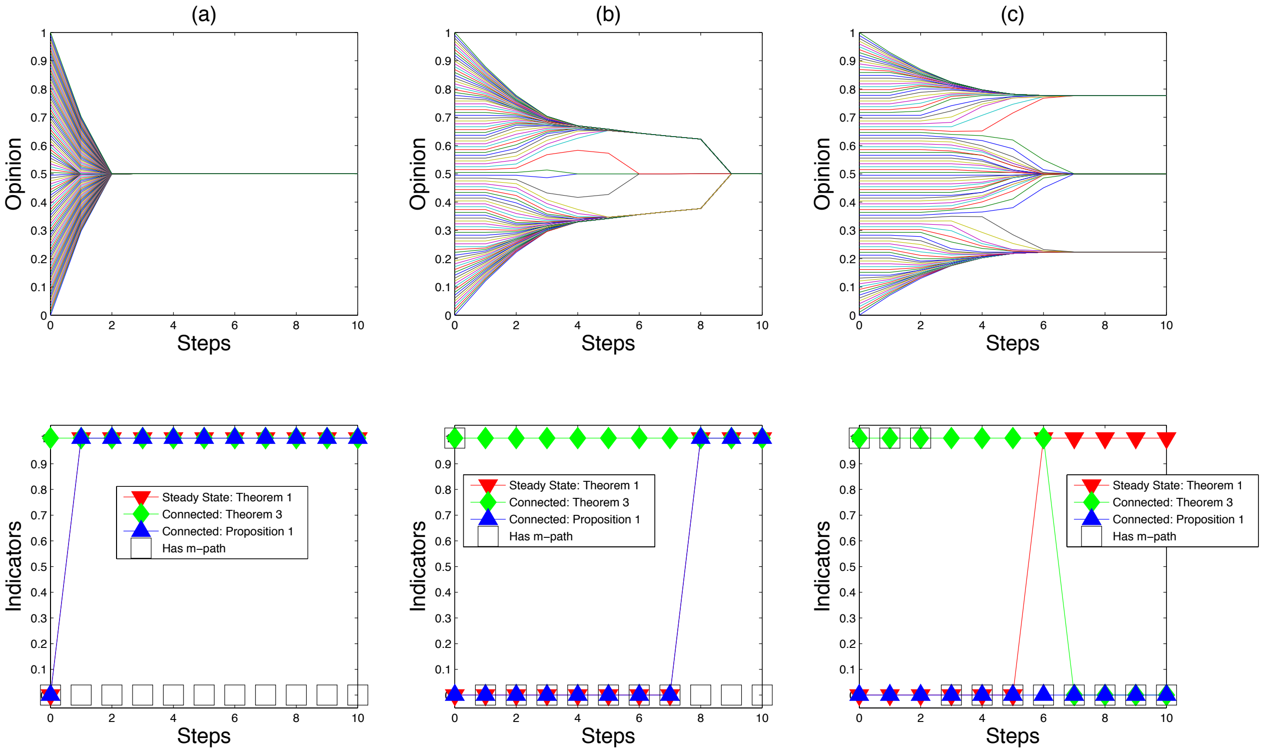


Figure 3.3: Example of application of the proposed centralized criteria for the convergence to a steady state and for the convergence to an agreement to a case with $n = 100$ agents with deterministic equidistant initial opinion profile. Column (a) and (b) and (c) show the results for $\varepsilon = 0.6$, $\varepsilon = 0.25$ and $\varepsilon = 0.15$, respectively. The first row of plots shows the evolution for the agents; the second row shows the metric for the convergence to a steady state (red triangles facing downwards), the connectedness metric of Theorem 3.3.1 (green diamonds), the connectedness metric of Proposition 3.4.1 and the presence or absence of an m-path (black empty boxes).

in Theorem 3.3.1 for the connectedness of the graph (green diamonds), the connectedness criterion of Proposition 3.4.1 (blue triangles facing upwards) and the presence or absence of an m-path (black empty boxes). Figures show that the indicator for the detection of the convergence to a steady state is indeed able to identify the time instant for which a steady state is reached; for instance in case (a) the indicator reaches one in $t = 2$, implying a steady state is obtained in $t = 3$, while for case (b) and (c) the detection instants are $t = 8$ and $t = 6$, respectively. As for the connectedness indicators, note that the one from Theorem 3.3.1 is equal to one for all time steps in the cases (a) and (b) where the HK model reaches an agreement, while in case (c) it goes to zero at $t = 6$, hence the graph gets disconnected at $t = 7$ and no agreement is reached. The indicator of Proposition 3.4.1, instead, is not always able to predict that the graph will be connected in the following time instant, because it is only a sufficient condition. Note further that in case (b) graph $\mathcal{G}(t)$ contains an m-path for $t = 0$, while for case b it contains an m-path for $t = \{0, 1, 2\}$, hence Criterion 3.2.1 cannot be applied.

3.5.2 Numeric examples of distributed criteria

In Fig. 3.4 numerical implementations of the distributed criteria are reported. The first and the second column of plots shows the results for $n = 150$ agents, and for ε equal to 0.3, and 0.11, respectively. In case (a) the agents reach an agreement in $t = 4$, hence all the agents detect at the same time in $t = 3$ that the agreement and the steady state will be reached in $t = 4$. In case (b) a steady state is reached in $t = 5$, but the network splits in 4 clusters of opinion, and such clusters converge in different time instants. Specifically the upper and lower cluster both reach a steady state at time instant $t = 4$, while the two central clusters reach both a steady state at time instant $t = 5$ and the four groups of agents detect the convergence to a steady state in two different time instants ($t = 3$ and $t = 4$, respectively); as for the agreement, since the network is split into clusters,

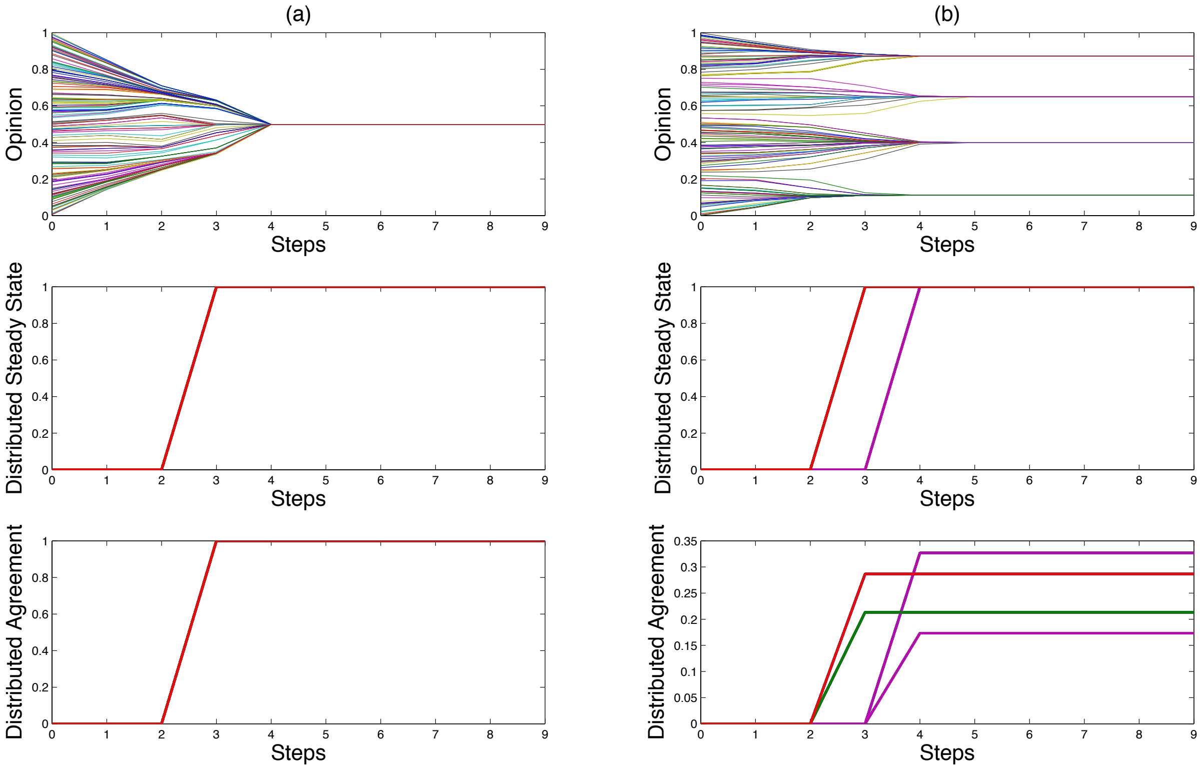


Figure 3.4: Example of application of the proposed distributed criteria for the convergence to a steady state and for the convergence to an agreement relating to the case with $n = 150$ agents with deterministic equidistant initial opinion profile. Column (a) and (b) show the results for $\varepsilon = 0.3$ and $\varepsilon = 0.11$, respectively. The upper row of plots shows the evolution for the agents; the central row shows the distributed metric for the convergence to a steady state defined in Corollary 3.2.1; the lower row of plots, finally, show the distributed metric for the convergence to an agreement, introduced in Proposition 3.4.2.

the criterion of Proposition 3.4.2 is never met, and none of the curves in the lower plot of column (b) reaches one. Note that, although the agents in the same cluster reach the same conclusions at the same time instant, the algorithms adopted are fully distributed.

Chapter 4

Distributed Data Clustering via Opinion Dynamics

In this section we present a distributed method to partition a large set of data in clusters, characterized by small in–group and large out–group distances. We take into account a wireless sensors network scenario where each sensor acquires a large set of data. The main goal is to provide a way to group the sensors in homogeneous clusters by information type. In previous literature, the desired number of clusters must be specified a priori by the user. In our approach, the clusters are constrained to have centroids with a distance at least ε between them and the number of desired clusters is not specified. We exploit the peculiarity of HK model to generate several clusters, with the aim to abstract from a large set of measurement data into few values (i.e, the opinion clusters). In this view, the HK model can be seen as a powerful methodology to determine the number of clusters while respecting the constraints on the distance among cluster centroids. Even though the HK model in its original setting arise as centralized algorithm [41], in this chapter we provide a distributed implementation based on a combination of distributed consensus algorithms for discrete-time systems. A comparison with k-means algorithm concludes the discussion.

4.1 Introduction to the Data Clustering Problem

Automatically determining the number of clusters has been one of the most difficult problems in data clustering [47]. Most methods for automatically determining the number of clusters cast it into the problem of model selection. Usually, clustering algorithms are run with different values of k; the best value of k is then chosen based on a predefined criterion. The problem of grouping large amounts of data into a small number of subsets with some common features among the elements (often referred to as the *data clustering problem*), has attracted the work of several researchers in different fields, ranging from statistics, to imagine analysis and bioinformatics [25, 34, 70].

Data clustering techniques are developed to partition an initial set of observation data into collections with small in–group distances and big out–group distances. Among the existing techniques, one of the most used is the k–means algorithm, or its successive extensions (e.g., fuzzy c–means [28], mixture of Gaussians algorithms [23], etc.). Given a set of initial observation data and a number k of desired clusters, the k-means algorithm computes a sub–optimal placement of k cluster centroids and assigns the observations to such centroids, alternating between an assignment phase, where each observation point is assigned with its nearest centroid, and refinement phase, where each centroid position is updated as the center of mass of all observations belonging to that centroid.

A well-known limitation of data clustering algorithms, such as the k–means algorithm, is that the number of clusters has to be specified beforehand, based, e.g., on subjective evaluations or a priori analysis. Since this assumption is typically not feasible in practice, a typical solution consists of running several times the algorithm with a different number of clusters, and then deciding on the best obtained solution based on a–posteriori evaluation [48]. Another issue of traditional algorithms is that there is no guarantee that the clusters are sufficiently far from each other. To this respect, distance constrained data clustering approaches have been devised in the literature: in [9, 95] the considered constraints are the so-called *must–links* (i.e., an observation i must belong to a cluster j) and *cannot–link* (i.e., an observation i can not belong to a cluster j); in [21] the feasibility of a constrained problem involving the so–called δ*-constraints* (i.e., any two observations must have a distance greater than δ) and the ε*-constraints* (i.e., for any observation i in cluster j there must be at least another observation h in cluster j such that the distance between i and j is less than ε) is given. To the best of our knowledge, nowadays there is no methodology to specify a constraint on the distance between cluster centroids, while this class of constraints might help finding a choice for the number k when such value is a–priori unknown. This problem has a particular relevance in a distributed setting, where a network of sensors has to classify information provided by several sensors without a central authority, but using only local data exchange among neighbours in the network. In this section, based on the preliminary results in [73] a novel approach to solve the data clustering problem with a distance constraint among cluster centroids is provided, that does not require the specification of the initial number k of clusters and that is based on a extension of the HK model [35, 41],which handles scalar data, in order to to handle data in $\mathbb{R}^d$. Such a model, similarly to consensus [29, 71] represents how a set of agents interact in order to reach a local agreement, but the agents may split in several clusters depending on their "opinions".

4.2 Data Clustering in Complex Networks

One of the modern challenge in complex networks and algebraic graph theory research is to partition a large graph (i.e., network) into cohesive subgraphs based on their link structure and node attributes [47]. The problem becomes even more complicated when the links, (which represent relations between objects) are allowed to have diverse types. One of the key issues is to define an appropriate clustering criterion for relational data, as shown in Fig. 4.1.

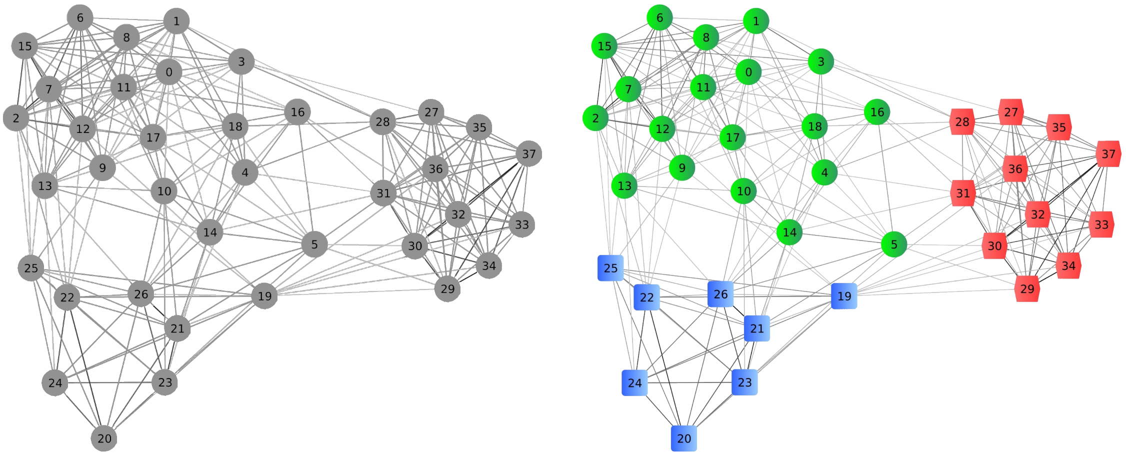


Figure 4.1: Example of data clustering in a complex network composed by 31 nodes. To the left side the network with grey nodes before the clustering operation is reported, while in the right side it is highlighted the network split up in three different clusters (green, blue and red colors).

The goal of data clustering is to discover the natural grouping(s) of a set of patterns, points, or objects. Clusters can differ in terms of their shape, size, and density, farther the presence of noise in the data makes the detection of the clusters even more difficult. An ideal cluster can be defined as a set of points that is compact and isolated, actually, a cluster is a subjective entity that is in the eye of the beholder and whose significance and interpretation requires domain knowledge. Generally, humans are excellent cluster seekers in two and three dimensions, while it needs automatic algorithms for high-dimensional data when complex systems are considered. It is this challenge along with the unknown number of clusters for the given data that has resulted in thousands of clustering algorithms that have been published and that continue to appear. An operational definition of clustering can be stated as follows: given a representation of n objects, find k groups based on a measure of similarity such that the similarities between objects in the same group are high while the similarities between objects in different groups are low. More precisely, let us formalize the following problem.

Problem 4.2.1. ***(Standard data clustering problem)*** *Given $k \leq n$, the data clustering problem consists in finding $r_{ij} \in \mathbb{B}$, $c_j \in \mathbb{R}^d$ for $i = 1, \dots, n$ and $j = 1, \dots, k$ that minimize*

$$D = \textstyle\sum_{i=1}^{n} \sum_{j=1}^{k} r_{ij} ||x_i - c_j||^2$$

$$\text{Subject to} \qquad (4.1)$$

$$\begin{matrix} (I) \\ (II) \end{matrix} \begin{cases} \sum_{j=1}^{n} r_{ij} = 1 & \forall i = 1, \dots, n \\ r_{ij} \in \{0, 1\} & \forall i = 1, \dots, n; \forall j = 1, \dots, k \end{cases}$$

The problem (4.1) is hard to solve, and in the literature several iterative algorithms have been devised.

4.2.1 The k-means Algorithm

Among the data clustering algorithms, the well-known *k-means algorithm* [62] is the most popular and the simplest partitional algorithm. Specifically, starting with a random set of k centroids $\{c_1(0), \dots, c_k(0)\}$, the algorithm alternates for each step an *assignment* and a *refinement* phase.

During the assignment phase, each observation x_i is assigned to the set characterized by the nearest centroid, i.e.,

$$r_{ih}(t) = \begin{cases} 1 & \text{if } h = \ \text{argmin}_j ||x_i - c_j(t)|| \\ 0 & \text{else} \end{cases} \tag{4.2}$$

Within the refinement phase each centroid c_j is updated as the centroid of the observations associated to the cluster $C_j(t)$, according the following rule:

$$c_j(t+1) = \frac{\sum_{i=1}^{n} r_{ij}(t) x_i}{\sum_{i=1}^{n} r_{ij}(t)} \tag{4.3}$$

The two steps are iterated until convergence or up to a maximum of M iterations.

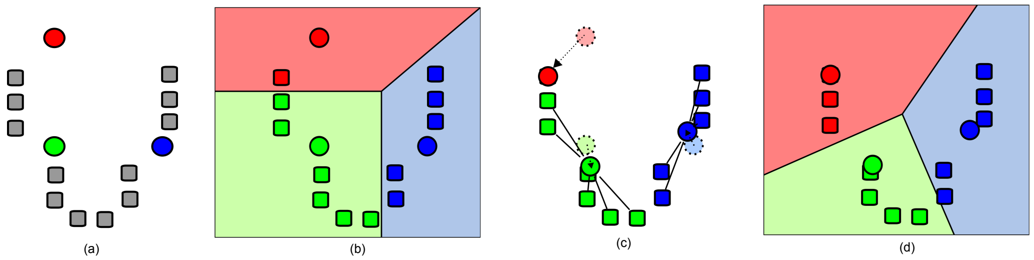


Figure 4.2: Example of execution of k-means algorithm (source; Wikimedia Commons available under GNU Free Documentation License v. 1.2).

Fig. 4.2 reports a simulation run of the algorithm for a set of $n = 12$ observations in $\mathbb{R}^2$ and for $k = 3$. To this regard, Fig. 4.2.(a) shows with circles the initial centroids, Fig. 4.2.(b) and Fig. 4.2.(c) report the assignment and refinement phases for the first step, while Fig. 4.2.(d) depicts the assignment phase for the second step. The k-means algorithm is granted to converge to a local optimum value, while there is no guarantee to converge to the global optimum [11, 62]. Since there is a strong dependency on the initial choice of the centroids, a common practice is to execute the algorithm several times and select the best solution. The algorithm, moreover, is extremely sensitive to outliers, which can significantly alter the results; to cope with this issue, the outliers have to be identified and excluded prior to the execution of the algorithm. Note that for each step, each of the n observations and for each of the d components of the observations, the algorithm calculates the difference with each of the k centers; hence the computational complexity is $O(d\,k\,n\,M)$, where M are the total number of iterations [11]. Notice that in [74] a distributed implementation of the k-means algorithm has been provided, with a computational complexity of $O(d\,k\,n^2\,M)$ for each agent. Note further that, unfortunately, the k-means algorithm is not able to solve Problem 4.4, hence we need to seek for other solutions.

4.3 Problem Setup

Consider a collection of n sensors, each equipped with a d-dimensional measurement or piece of information $\{x_1, \ldots, x_n\}$, with $x_i \in \mathbb{R}^d$. We want to select a number k of groups or *clusters* $\{C_1, \ldots, C_k\}$, $k \leq n$. Every cluster C_j is assigned with a cluster centroid $c_j \in \mathbb{R}^d$, which represents the centroid of the observations allocated to that cluster. A data point $x_i \in \mathbb{R}^d$ is said to belong to cluster C_j if its distance from the centroid c_h, with $h \neq j$, of every other cluster C_h is larger than its distance from c_j. If x_i belongs to C_j we can set a binary assignment variable $r_{i,j} \in \mathbb{B}$ to 1, and to 0 otherwise. The solution of a data clustering problem with distance constraints involves the computation of the optimal choice of cluster centroids c_j, (for $j = 1, \ldots, k$) that minimizes the distance of every measurement data point x_i from the cluster it belongs to. More formally, we need to solve the following:

Problem 4.3.1. ***(Data clustering with distance-constrains)*** *We want to find the number of clusters k, the cluster centroids, $c_j \in \mathbb{R}^d$, and measurement data assignments, $r_{i,j} \in \mathbb{B}$ that minimize the index*

$$D = \sum_{i=1}^{n} \sum_{j=1}^{k} r_{i,j} \left|\left| x_i - c_j \right|\right|^2 , \tag{4.4}$$

subject to the constraints

$$\begin{array}{c} (I) \\ (II) \\ (III) \end{array} \begin{cases} \sum_{j=1}^{k} r_{ij} = 1 \quad \forall i = 1, \ldots, n \\ ||c_h - c_j||^2 \geq \varepsilon \quad \forall i, j = 1, \ldots, k; \quad h \neq j \\ r_{ij} \in \mathbb{B} \quad \forall i = 1, \ldots, n; \quad \forall j = 1, \ldots, k \end{cases}$$

The first set of constraints (I) along with the third one implies that each observation is assigned exactly with one cluster; the constraints (II) imply that ε is a lower bound for the distance between any pair of centroids; finally constraints (III) imply that r_{ij} are binary decision variables.

This problem is novel, since in the literature the set of constraints (II) are typically not considered. Including such constraints, however, is quite useful, since it may improve the algorithm's ability to detect homogeneous clusters. The problem is very hard to solve exactly and traditional methods as the k-means algorithm [62] may fail even at finding an admissible solution, as it will be shown in the next section.

On the other side, the proposed algorithm uses an algorithm based on the HK model to choose an admissible, although sub-optimal solution, and it has a modest increase in computational complexity with respect to the k-means algorithm, while allowing several advantages:

- the algorithm proposed always finds an admissible, although sub-optimal solution to the problem by means of the HK model, while the k-means algorithm may fail;
- the proposed approach does not require the user to define a priori the number of clusters, but it finds automatically a suitable number of clusters based on the parameter ε;
- outliers can be automatically isolated, without any a priori data processing; this feature can be obtained by dropping out the clusters whose cardinality is significantly less than the others;
- the solution provided is deterministic, i.e., for fixed ε and fixed observations the result is always the same, while the k–means algorithm depends on the initial random choice of the centroids;
- while traditional approaches are very computationally expensive when applied in a decentralized and distributed setting [74] (i.e., for a sensors network), this method can be distributed with a modest increase in computational complexity [66].

4.4 Data Clustering with Distance Constraints via Opinion Dynamics Model and k–means Algorithm

In this section we propose an algorithm for data clustering with distance constraints, based on a generalization of the HK model which consists of a two–phase process (Fig. 4.3).

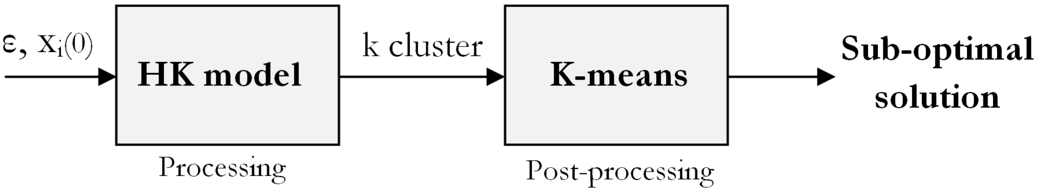


Figure 4.3: Representation of the two phases of the proposed approach: in the first phase, the number k of clusters is obtained by the processing with HK model, in the second one, post–processing of clusters is carried out by the k–means algorithm.

Specifically, measurement data is processed by an HK-like opinion dynamics "filter", which eventually segment the data into κ clusters. Moreover, based on the weight of each cluster (namely represented by the number of measurement data that has converged to that cluster), data outliers can be filtered out from the original set. More precisely, we assume that each agent i is provided with an initial measurement represented by a vector $x_i \in \mathbb{R}^d$. Every agent has an initial state $x_i(0) \in \mathbb{R}^d$, which is updated according to the following set of iterative rules:

$$\begin{pmatrix} x_{1,i}(t+1) \\ \vdots \\ x_{n,i}(t+1) \end{pmatrix} = A(x(t), \varepsilon) \begin{pmatrix} x_{1,i}(t) \\ \vdots \\ x_{n,i}(t) \end{pmatrix},$$

for $i = 1, \ldots, d$, where the adjacency matrix $A(x(t), \varepsilon)$ is computed based on the following definition of neighbourhood:

$$\mathcal{N}_i^*(t) = \{j \in \{1, \ldots, n\}\, s.t.\, ||x_i(t) - x_j(t)|| \leq \varepsilon\},$$

where $||\cdot||$ is the Euclidean norm. Note that for $d = 1$ the standard HK model is obtained.

Let us state the following conjecture.

Conjecture 4.4.1. ***(HK multi-dimensional extension)*** *The extension of the HK model to opinions in $\mathbb{R}^d$ using the Euclidean metric converges to a steady state in finite time.*

In order to provide evidence that supports this conjecture, we provide the simulation results of Fig. 4.4, where the average instant in which a steady state is reached and the number of clusters obtained are reported for several choices of $n = 50, 100, 200$ agents and $\varepsilon \in [0.1,\, 0.5]$; for each choice of n, ε, the average of 100 runs with random initial opinions in $[0, 1]$ are reported. Notice that all the executions reached an exact agreement in finite time. The proposed approach, therefore, does not require the user to specify a value for the parameter k, but it finds a suitable number of clusters based on the parameter ε. A high value of ε means very large and sparse clusters (eventually also very few of them) while a small value of ε means very compact and small clusters (eventually, many of them). One of the biggest problems of the k-means-like algorithms, is that the outliers have to be preprocessed and excluded, otherwise they would influence considerably the quality of the clustering. In the proposed approach, depending on the choice of ε, very far observation are not influenced by the others, and are assigned to a singleton (or more in general to a cluster composed of very few elements). It should be noted that it is always possible to execute a k-means algorithm with κ clusters in order to attempt to refine the solution found, but this can be done only if the solution of the k-means algorithm does not violate the constraints on the distance among cluster centroids. Therefore,

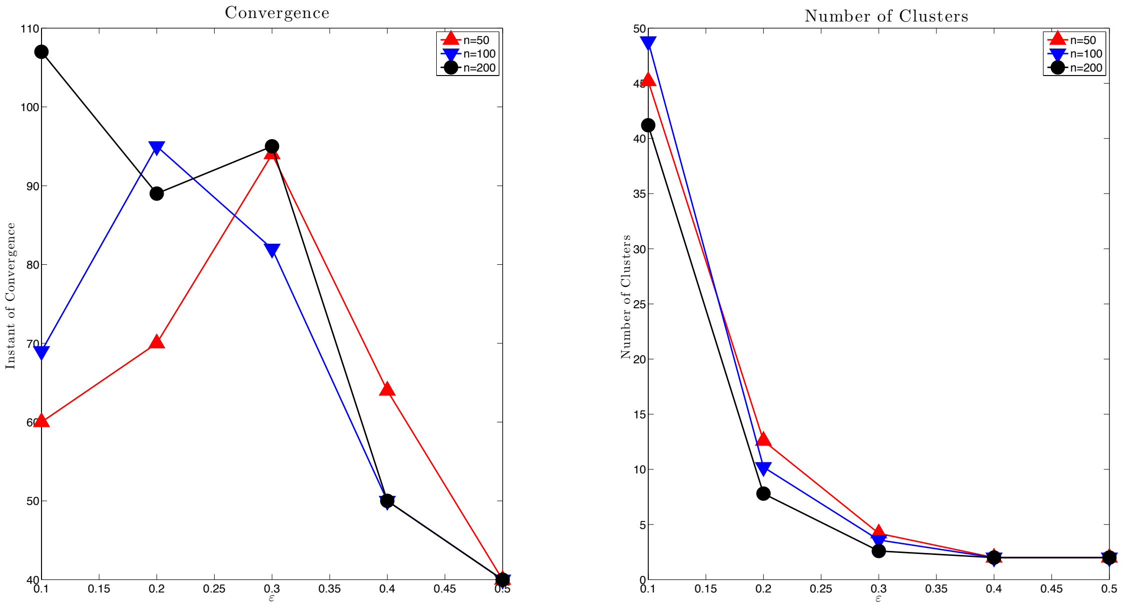


Figure 4.4: Average instant at which a steady state is reached (left) and average number of obtained clusters (right). Simulation runs are obtained with $n = 50, 100, 200$ agents, and $\varepsilon \in [0.1,\, 0.5]$. In particular, for each pair (n, ε), the average values are computed over 100 runs with random initial opinions in the interval $[0, 1]$.

the proposed algorithm appears as a good candidate to allow the clustering of a set of sensors or mobile robots, based on perceived information such as position or other sensorial information (temperature, humidity, etc.). As for the computational complexity of the extension of the HK model to opinions in $\mathbb{R}^d$, it can be noted that such complexity is the same of executing d scalar HK models, hence it is $O(d\,n^2\,M)$; since the complexity of the centralized k-means algorithm is $O(d\,k\,n\,M)$ [62], the proposed approach determines an increase in computational complexity with respect to the k-means algorithm of a factor $\frac{n}{k} \geq 1$.

4.5 Distributed Implementation of the Hegselamann-Krause Model

As shown in the previous sections, the HK opinion dynamics model can be used to provide an admissible, although sub-optimal solution to the distance–constrained data clustering problem. However, since the topology underlying the HK model is indeed a state-dependent topology, theoretically each agent may exchange information with each other agent, depending just on the difference in their opinions.

In order to adopt the HK model in a distributed perspective, therefore, we need to provide a different implementation, as provided in the following Algorithm.

Algorithm 1: Distributed HK Opinion Dynamics Algorithm

```
for $t = 1, \dots, M$ do
  for $h = 1, \dots, n$ do
    /*Transmit the state of agent $h$ to each other*/
    $\delta^i_h = \begin{cases} x_h(t) & \text{if } i = h \\ 0 & \text{else} \end{cases};$
    $\delta_h = \text{max-consensus}_i(\delta^i_h, \delta^j_h | \ j \neq i, \mathcal{G}_c, n);$
    /*Calculate $x_i(t+1)$*/
    $e^i_h = \begin{cases} |x_i(t) - \delta_h| & \text{if } |x_i(t) - \delta_h| \leq \varepsilon \\ 0 & \text{else} \end{cases};$
    $x^i_h(t+1) = n \cdot \text{average-consensus}_i(e^i_h, e^j_h | \ j \neq i, \mathcal{G}_c, t_{\max});$
    if $i == h$ then
      $x_i(t+1) = x^i_h(t+1);$
    end if
  end for
end for
```

Since the agents have a unique identifier $h = 1, \dots, n$, for each time step and for each agent h a distributed procedure is executed by all agents in order to calculate $x_h(t+1)$. More specific, for an agent h, each agent i selects $\delta^i_k = x_h(t)$ if $i = j$ and $\delta^i_k = 0$ otherwise. Then the agents execute a *max*–consensus procedure using δ^i_k as initial condition; as a result of such an operation, each agent i knows $x_h(t)$ and is able to determine whether

$||x_h(t) - x_i(t)|| \leq \varepsilon$ or not. Such knowledge is stored in a variable e_i for each agent, and using e_i as initial condition for an average-consensus algorithm, the agents obtain

$$e = \frac{\sum_{h=1}^{n} ||x_h(t) - x_i(t)||}{n}$$

which, multiplied by n calculated according to [84], yields the value $x_h(t)$. As for the computational complexity of the distributed version, note that for each step $t = 1, \ldots, M$, and for each agent $h = 1, \ldots, n$, the agents execute a *max*-consensus ($O(d\, n)$ steps) and an *average*–consensus ($O(d\, t_{\max})$ steps), both with initial conditions in $\mathbb{R}^d$, and the complexity is $O(d\, n\, M\ \max\{t_{\max}, n\})$ where $t_{\max}$ is the number of iterations of the average consensus algorithm. Since, typically, $t_{\max} > n$, the distributed setting has $\frac{t_{\max}}{n}$ times the complexity of the centralized algorithm. Moreover, since the computational complexity of the distributed k-means algorithm is $O(d\, k\, n\, M)$ the proposed distributed algorithm has $\frac{t_{\max}}{k\, n}$ times the complexity of the distributed k-means algorithm, therefore the complexity is reduced for $t_{max} < k\, n$.

4.6 Simulation Results

As discussed above, the k–means algorithm is generally unable to solve the data clustering problem with distance constraints. In this section, in order to show the effectiveness of the proposed approach some examples are reported. A comparative simulation between the HK model and k–means algorithm is first addressed. Afterwards the potentiality of the proposed mixed approach is showed, then the distributed implementation is discussed.

Fig. 4.5 shows an example in $\mathbb{R}^2$ with $\varepsilon = 0.6$ and $n = 200$ observations. The application of the HK opinion dynamics model yields $k = 63$ clusters and $D \approx 0.35$. Unfortunately, the k-means algorithm finds a solution for $k = 63$ which, although having $D \approx 0.29$, is not feasible for Problem 4.4 (violations of the constraints are shown with red

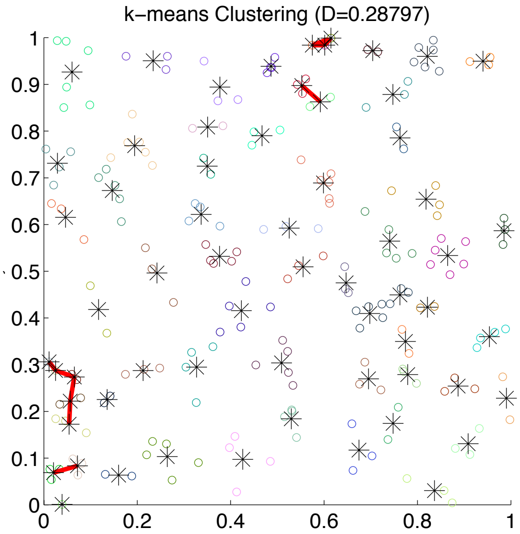


Figure 4.5: Clustering for $n = 200$ observations that are uniformly distributed in the interval $[0, 1]$ and $\varepsilon = 0.06$: the HK opinion dynamics model finds $k = 63$ clusters and $D \approx 0.35$. The solution of the k-means algorithm for $k = 63$ is better in terms of the objective function ($D \approx 0.29$), but it does not respect the distance constraints (red thick lines represent violations).

lines). However, this is not always verified, and the k-means algorithm may represent a quite good post processing algorithm to be applied, after that the HK opinion dynamics algorithm has selected a number k of clusters.

Fig. 4.6 shows a case where $n = 200$, $\varepsilon = 0.18$ and the HK opinion dynamics model gives $k = 5$ clusters. Using the k-means algorithm for $k = 5$ a better solution is obtained, and the constraints are not violated, hence in this case post-processing the result of the HK model via k-means algorithm yields a better result.

Fig. 4.7, shows the ability of the proposed methodology to isolate the outliers. For $n = 200$ and $\varepsilon = 0.1$ the HK model finds $k = 7$ clusters, two of which are singletone. Executing a k-means algorithm for $k = 7$ gives a worse results in terms of the objective function with respect to the HK approach. If, conversely, the two outliers are removed (k becomes equal to 5), then the k-means algorithm has better results in terms of the objective function.

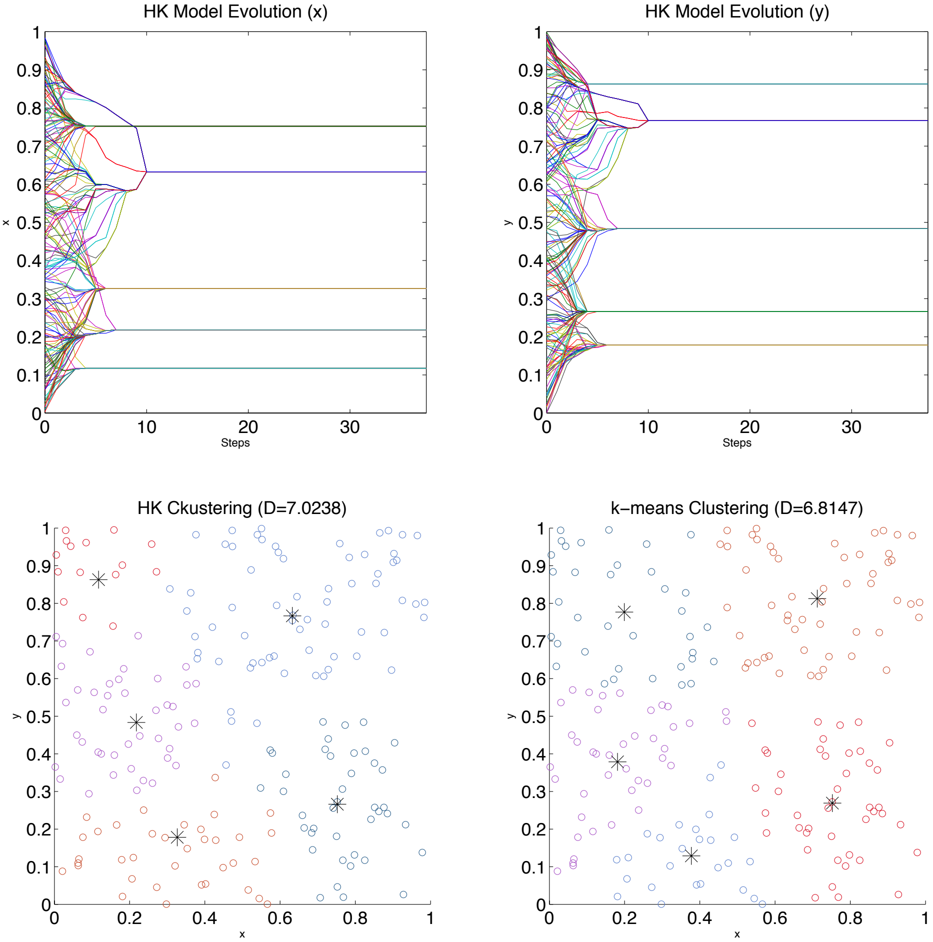


Figure 4.6: Clustering for $n = 200$ observations that are uniformly distributed in the interval $[0, 1]$ and $\varepsilon = 0.18$: the HK opinion dynamics model finds $k = 5$ clusters. The solution of the k-means algorithm for $k = 5$ is better in terms of the objective function D without violating the distance constraints.

Fig. 4.8, shows an example of application of the distributed HK model provided in Algorithm 1, in a case where $n = 40$ agents, each with a random position in $[0, 1]^2$, have to be clustered depending on their positions in a way that the centroids are not closer than $\varepsilon = 0.1$. The simulation was executed for $M = 15$ iterations, and the average-consensus algorithms were executed each $t_{\max} = 100$ steps. The topology of the network of agents is given in the upper left plot, while the lower left plots show the results of the distributed HK model for the x and y coordinates. The consensus steps performed by each agent during one iteration of the distribute HK algorithm are reported in the upper central plot, where

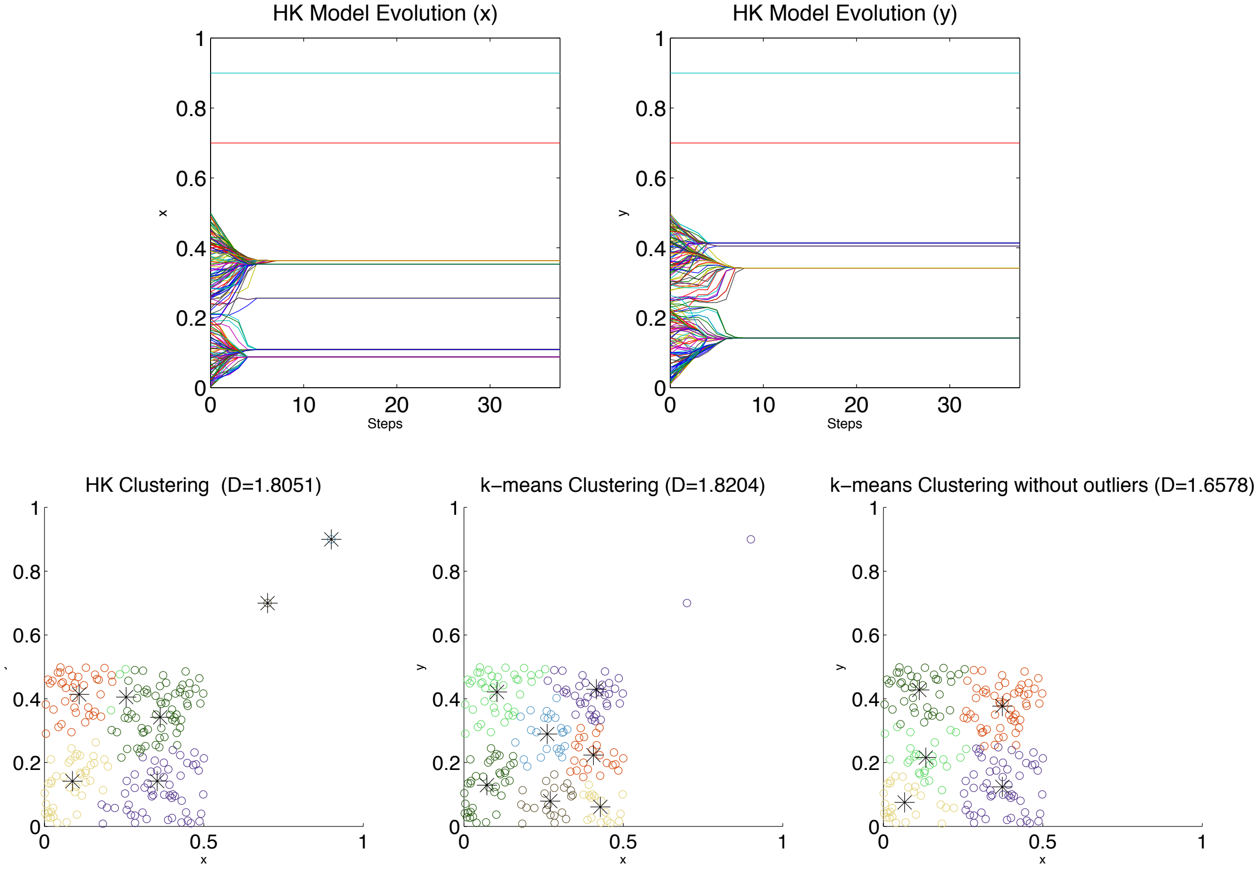


Figure 4.7: Clustering for $n = 200$ observations with $\varepsilon = 0.1$. The HK opinion dynamics model finds $k = 7$ clusters, of which 2 are singleton clusters containing one outlier each. The solution of the k-means algorithm for $k = 7$ is worse than the one found by HK model. If, however, the 2 outliers are removed (thus $k = 5$), the k-means algorithm has better results in terms of the objective function.

"AVG" stands for average consensus and "MAX" stands for max-consensus, while the distribution of max-consensus and average-consensus steps over the entire execution of the algorithm are reported in the upper rightmost plot. The lower right plots, eventually, show the results of the distributed HK model in terms of the clustering in $[0, 1]^2$: the agents are divided in $k = 20$ groups, and the objective function has a value $D \approx 0.06$. Notice that, in this case, the k-means algorithm with $k = 20$ yields both a worst solution in terms of the objective function $D \approx 0.12$, and in terms of violation of the constraints (5 constraints are violated, and are highlighted with red thick segments).

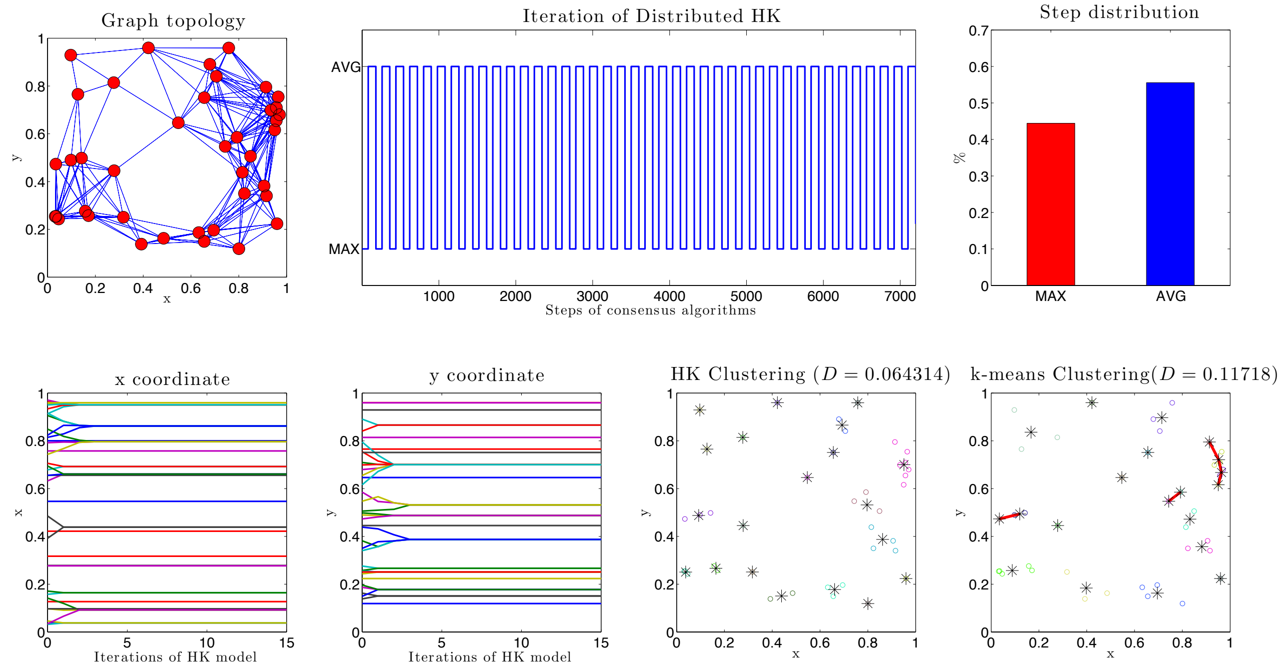


Figure 4.8: Distributed clustering on the positions of $n = 40$ agents with $\varepsilon = 0.1$: the distributed HK opinion dynamics model finds $k = 21$ clusters. The distributed implementation of the algorithm requires to alternate between max-consensus algorithms and average-consensus algorithms, as shown by the upper rightmost figure.The solution of the k-means algorithm for $k = 21$ is both worse in terms of the objective function D and does not respect the distance constraints (red thick lines in the lower rightmost figure represent violations).

Chapter 5

Attack and Defense Location in Line Network Systems

In this chapter we focus on identify an optimal network location for the detection of an unknown perturbation signal. In particular, we define detection metrics based on the *static gain* of the dynamical network system [49]. Among the different communication topologies (as showed in previous Section 2.2), we consider a class of network systems with line interconnection structure and we identify optimal sensor locations based on the network weights and the origin of the perturbation signal. Surprisingly we find that, in some cases, sensors should be located as far as possible from the origin of the perturbation while, in other cases, the location between the sensors and the origin of the perturbation should be minimized. Although our results pertain a specific, and simple, interconnection structure, we conjecture that a similar behaviour may appear in more complex interconnection structures, and that novel techniques are necessary to relate the structure with the dynamical properties of the associated network system.

5.1 Motivation

The problem of selecting sensors and actuators in dynamical systems has received considerable attention in the controls community. Typically, sensors and actuators are selected to maximize, respectively, certain observability and controllability metrics, often quantified by Gramian matrices [2, 36, 63]. For small-scale problems, the maximization of observability and controllability metrics often relies on combinatorial optimization procedures, which do not offer any particular insight into the structure of the problem, and become computationally infeasible for large problems.

Motivated by a renewed interest in network systems, and particularly by the need for a deepened understanding of the relation between network structure and network dynamics, recent studies have focused on determining suitable optimization metrics for sensor and actuator placement in large-scale systems [88], as well as on highlighting tradeoffs and relations between network structure and the associated Gramians [75,93,100]. In a related fashion, the controllability Gramian has been analysed for security and synchronization problems in network systems [24, 90]. In this thesis, we continue the work along these directions by considering the trace of the cross-Gramian [49] as a metric for joint sensor and actuator location in network systems, and by providing explicit results for a class of network systems.

5.2 Problem Statement

In this section we detail our setup and introduce preliminary concepts that will be used throughout the technical treatment. Consider a network represented by an interconnection graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, n\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denote the vertex and edge sets, respectively, and a weighted adjacency matrix $A \in \mathbb{R}^{n \times n}$ containing the weights of the network interconnection edges. That is, $A = [a_{ij}]$, where $a_{ij} = 0$ if $(i, j) \notin \mathcal{E}$, and a_{ij}

equals the weight of the edge (i, j) otherwise. Assume that a subset of vertices $\mathcal{U} \subseteq \mathcal{V}$, with $|\mathcal{U}| = m$, is affected by an external and unknown signal representing genuine disturbances or malicious attacks, and that a subset of nodes $\mathcal{Y} \subseteq \mathcal{V}$, with $|\mathcal{Y}| = p$, is equipped with sensors capable of measuring the nodes activity. Let $B \in \mathbb{R}^{n\times m}$ be the submatrix of the n-dimensional identity matrix with columns indexed by $\mathcal{U}$, and let $C \in \mathbb{R}^{p\times n}$ the submatrix of the n-dimensional identity matrix with rows indexed by $\mathcal{Y}$. Let $x_i \in \mathbb{R}$ be the state associated with node i. As previously introduced in Section 2.6, we assume that the network systems evolves according to linear, continuous-time dynamics described by its weighted adjacency matrix, that is,

$$\begin{aligned} \dot{x}(t) &= A\,x(t) + B\,u(t), \\ y(t) &= C\,x(t), \end{aligned} \tag{5.1}$$

where $x = (x_1, \ldots, x_n)^T$, $u = (u_1, \ldots, u_m)^T$, and $y = (y_1, \ldots, y_p)^T$ are the network state vector, the unknown disturbance input, and the measured output vector, respectively. For simplicity of analysis, the output disturbance affecting the plant has been neglected.

Two dual problems are of interest. On the one hand, based on the knowledge of the network dynamics A and sensor locations $\mathcal{Y}$, an *attacker* aims to find optimal input vertices $\mathcal{U}$ to maximally disrupt the network system while preventing observability from the sensor nodes. Conversely, based on the knowledge of the network dynamics A and input locations $\mathcal{U}$, the objective of a security system is to determine the sensor nodes $\mathcal{Y}$ ensuring optimal detectability of an unknown disturbance. The two problems can be addressed independently by defining quantitative notions of controllability and observability for the network system [36]. Yet, an approach combining both structural measures may lead to more robust results [69]. To this aim, we adopt the notion of *cross-Gramian* that was first introduced for single-input single-output linear systems in [31], and then extended for the multi-input multi-output case in [46]:

Definition 5.2.1. ***(Cross-Gramian)*** *For a stable network system described by the triple* (A, B, C)*, the cross-Gramian is defined as*

$$W_{co} = \int_0^\infty e^{At} \, B \, C \, e^{At} \, dt \, ,$$

or, equivalently, as the solution to the Sylvester's equation

$$A \, W_{co} + W_{co} \, A = -B \, C \, .$$

As shown in [31], the cross-Gramian matrix W_{co} carries information about both controllability and observability of the network. For symmetric systems, the cross-Gramian is indeed related to the controllability and observability Gramians, W_c and W_o, by the relation $W_{co} = \sqrt{W_c \, W_o}$ [49, 52]. Moreover, for single–input single–output systems, the trace of the cross–Gramian is related to the network steady-state gain $g = - C \, A^{-1} \, B$ by the equation [46]

$$\text{Trace} \, (W_{co}) = 1/2 \, g \, . \tag{5.2}$$

Eq. (5.2) suggests that Trace(W_{co}) can in fact be used to evaluate the amplification or attenuation of a network signal that is slowly varying with respect to the network dynamics. Motivated by the above discussion, we focus our attention on the following:

Problem 5.2.1. ***(Optimal sensor placement)*** *Given a network* $\mathcal{G}$ *with adjacency matrix* A*, input nodes* $\mathcal{U}$*, and dynamics as in* (5.1)*, determine the sensor locations* $\mathcal{Y}$*, with* $|\mathcal{Y}| = p$*, that maximize the trace of the associated cross-Gramian, that is,*

$$\begin{aligned} \max\nolimits_{\mathcal{Y}} \quad & \textit{Trace}(W_{co}), \\ \textit{subject to} \quad & |\mathcal{Y}| = p. \end{aligned} \tag{5.3}$$

Addressing the above problem for generic networks, with possibly multiple input nodes and with several sensors, is a daunting task. In this treatment we focus on a *single–attack, single–sensor* scenario, where $|\mathcal{U}| = 1$ and $|\mathcal{Y}| = 1$. Under this hypothesis, it can be shown that the trace of the cross-Gramian becomes

$$\text{Trace}\,(W_{co}) = -\frac{1}{2} A_{ij}^{-1} \,,$$

where A_{ij}^{-1} is (i,j)-th entry of the inverse of the adjacency matrix A. Thus, for the single-input single-sensor case, Problem 5.2.1 can be addressed by characterizing the inverse of the network adjacency matrix, so as to select the node i that maximizes the element A_{ij}^{-1}. This solution strategy is used in the next section to determine optimal sensor location in line networks, and to show that the network structure and dynamics may enforce counterintuitive relations regarding the location of sensors and actuators in a complex system.

5.3 Optimal Sensor Placement for Line Networks

5.3.1 Toeplitz line networks

Consider a Toeplitz line network with a tridiagonal adjacency matrix:

$$A = \begin{pmatrix} a & b & 0 & \cdots & 0 & 0 \\ c & a & b & \cdots & 0 & 0 \\ 0 & c & a & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a & b \\ 0 & 0 & 0 & \cdots & c & a \end{pmatrix}, \tag{5.4}$$

where the coefficients $a, b, c \in \mathbb{R}_{\neq 0}$. It is known that the inverse $A^{-1} = [A^{-1}_{ij}]$ of a tridiagonal Toeplitz matrix A is given by [53]:

$$A^{-1}_{ij} = \frac{1}{\theta_n} \begin{cases} (-1)^{i+j} b^{j-i}\, \theta_{i-1}\, \phi_{j+1}, & \text{for } i \leq j\,, \\ (-1)^{i+j} c^{i-j}\, \phi_{i+1}\, \theta_{j-1}, & \text{for } i > j\,, \end{cases} \tag{5.5}$$

where the coefficients θ_k are obtained through the forward iteration

$$\theta_k = a\,\theta_{k-1} - bc\,\theta_{k-2} \text{ for } k = 2, \dots, n\,, \tag{5.6}$$

with initial conditions $\theta_0 = 1$ and $\theta_1 = a$, while the coefficients ϕ_k are computed through the backward iteration

$$\phi_k = a\,\phi_{k+1} - bc\,\phi_{k+2} \text{ for } k = n-1, \dots, 1\,, \tag{5.7}$$

with final conditions $\phi_{n+1} = 1$ and $\phi_n = a$. It can be shown that $\theta_n = \det(A)$.

In this framework we are interested in characterizing the behaviour of the sequences (5.6) and (5.7) for large networks. For this reason, even though the network cardinality is finite, we adopt the following asymptotic definitions:

Definition 5.3.1. ***(Decreasing sequence)*** *A sequence $m(j)$, with $j = 1, 2, \dots,$ is* decreasing *if there exist j^*, $\gamma \in \mathbb{R}_{>0}$ and $\rho \in (0, 1)$ satisfying $|m(j)| \leq \gamma\rho^j$ for $j \geq j^*$.*

Definition 5.3.2. ***(Increasing sequence)*** *A sequence $m(j)$, with $j = 1, 2, \dots,$ is* increasing *if there exist j^*, $\gamma \in \mathbb{R}_{>0}$ and $\rho \in \mathbb{R}_{>1}$ satisfying $|m(j)| \geq \gamma\rho^j$ for $j \geq j^*$.*

The following theorem characterizes the behaviour of the entries A^{-1}_{ij} of the inverse of a tridiagonal Toeplitz matrix A.

Theorem 5.3.1. ***(Inverse of Toeplitz matrix)*** *Let $A \in \mathbb{R}^{n \times n}$ be a tridiagonal Toeplitz matrix with parameters $a, b, c \in \mathbb{R}_{\neq 0}$. Let*

$$\mu_+ = -a + \sqrt{a^2 - 4bc}, \text{ and } \mu_- = -a - \sqrt{a^2 - 4bc}.$$

As the dimension n grows, the entries A^{-1}_{ij} satisfy the following conditions:

1. *For all rows i,*

   (a) *if $|\mu_+| < 2|b|$ and $|\mu_-| < 2|b|$, then the sequence A^{-1}_{ij}, with $j \geq i$, is increasing;*

   (b) *if $|\mu_+| > 2|b|$ or $|\mu_-| > 2|b|$, then the sequence A^{-1}_{ij}, with $j \geq i$, is decreasing;*

2. *For all rows i,*

   (a) *if $|\mu_+| < 2|c|$ and $|\mu_-| < 2|c|$, then the sequence A^{-1}_{ij}, with $j < i$, is decreasing;*

   (b) *if $|\mu_+| > 2|c|$ or $|\mu_-| > 2|c|$, then the sequence A^{-1}_{ij}, with $j < i$, is increasing.*

Proof. To analyse the behaviour of the entries of A^{-1}, based on (5.5), one can conveniently fix a row index i and study the behaviour of all column entries j by distinguishing the cases with $i \leq j$ and $i > j$.

If $i \leq j$, the entry A^{-1}_{ij} can be factorized as

$$y^i(j) := A^{-1}_{ij} = \alpha(i)\,(-1)^j\,b^j \phi(j+1)\,,$$

where $\alpha(i) = (-1)^i\,b^{-i}\,\theta_n^{-1}\,\theta_{i-1}$. Starting from the last two column entries, which are given by

$$y^i(n+1) = \alpha(i)\,(-1)^{n+1}\,b^{n+1}\,,$$
$$y^i(n) = \alpha(i)\,(-1)^n\,b^n a\,,$$

one can recursively obtain all other entries of the i-th row through a backward recurrence relation that is derived below. Consider expanding the expression of the entry $y^i(j-1)$

as follows: for $j = n, \ldots, i+1$,

$$\begin{aligned}
y^i(j-1) &= \alpha(i)\,(-1)^{j-1}\,b^{j-1}\,\phi(j) \\
&= \alpha(i)\,(-1)^{j-1}\,b^{j-1}\,(a\,\phi(j+1) - bc\,\phi(j+2)) \\
&= -\frac{a}{b}\alpha(i)\,(-1)^{j}\,b^{j}\,\phi(j+1) \\
&\quad - \frac{c}{b}\alpha(i)\,(-1)^{j+1}\,b^{j+1}\,\phi(j+2) \\
&= -\frac{a}{b}y^i(j) - \frac{c}{b}y^i(j+1)\,.
\end{aligned}$$

Translating of a step backward the above relation yields:

$$y^i(j-2) = -\frac{c}{b}y^i(j) - \frac{a}{b}y^i(j-1),\ \text{for } j = n+1, \ldots, i+2\,.$$

To analyse the behaviour of the above difference equation, one can define the vector state $x(j) = \left(y^i(j), y^i(j-1)\right)^T$, whose backward evolution is described by the state form

$$x(j-1) = H_\phi\,x(j)\,,\ \text{for } j = n, \ldots, i+2\,, \tag{5.8}$$

with

$$H_\phi = \begin{pmatrix} 0 & 1 \\ -c/b & -a/b \end{pmatrix},$$

and initial condition $x(n) = \alpha(i)\,(-1)^n(-b^{n+1}, b^n a)^T$. Based on the expression of H_ϕ's eigenvalues,

$$\frac{\mu_+}{2\,b},\ \frac{\mu_-}{2\,b},$$

one can conclude the following. 1-a) If $|\mu_+| < 2\,|b|$ and $|\mu_-| < 2\,|b|$, both eigenvalues lay inside the unit circle, the backward evolution of the system in (5.8) is decreasing, while the sequence $\{y^i(j)\}$, for $j = i, i+1, \dots, n$, is increasing. 1-b) If $|\mu_+| > 2\,|b|$ and $|\mu_-| > 2\,|b|$, at least one eigenvalue lays outside the unit circle, the backward evolution of the system in (5.8) is increasing, and the sequence $\{y^i(j)\}$ is decreasing. If $i > j$, the entry of A^{-1}_{ij} can be factorized as

$$y^i(j) := A^{-1}_{ij} = \beta(i)\,(-1)^j\,c^{-j}\,\theta(j-1)\,,$$

where $\beta(i) = (-1)^i\,c^i\,\theta_n^{-1}\,\phi_{i+1}$. Starting from the first two column entries, $y^i(1) = 1$ and $y^i(2) = a$, one can recursively obtain all other entries of the i-th row through a forward recurrence relation that is derived below. Consider expanding the expression of the entry $y^i(j+1)$ as follows: for $j = 2, \dots, n$,

$$\begin{aligned} y^i(j+1) &= \beta(i)\,(-1)^{j+1}\,c^{-j-1}\,\theta(j) \\ &= \beta(i)\,(-1)^{j+1}\,c^{-j-1}\,(a\,\theta(j-1) - bc\,\theta(j-2)) \\ &= -\frac{a}{c}\beta(i)\,(-1)^j\,c^{-j}\,\theta(j-1) \\ &\quad -\frac{b}{c}\beta(i)\,(-1)^{j-1}\,c^{-j+1}\,\theta(j-2) \\ &= -\frac{a}{c}y^i(j) - \frac{b}{c}y^i(j-1)\,. \end{aligned}$$

Translating of a step forward the above relation yields:

$$y^i(j+2) = -\frac{b}{c}y^i(j) - \frac{a}{c}y^i(j+1), \text{ for } j = 1, \dots, n-1\,.$$

One can define the vector state $x(j) = (\theta(j), \theta(j+1))^T$ and study the evolution of the state form

$$x(j+1) = H_\theta\,x(j)\,, \text{ for } j = 1, \dots, i-1\,, \tag{5.9}$$

with

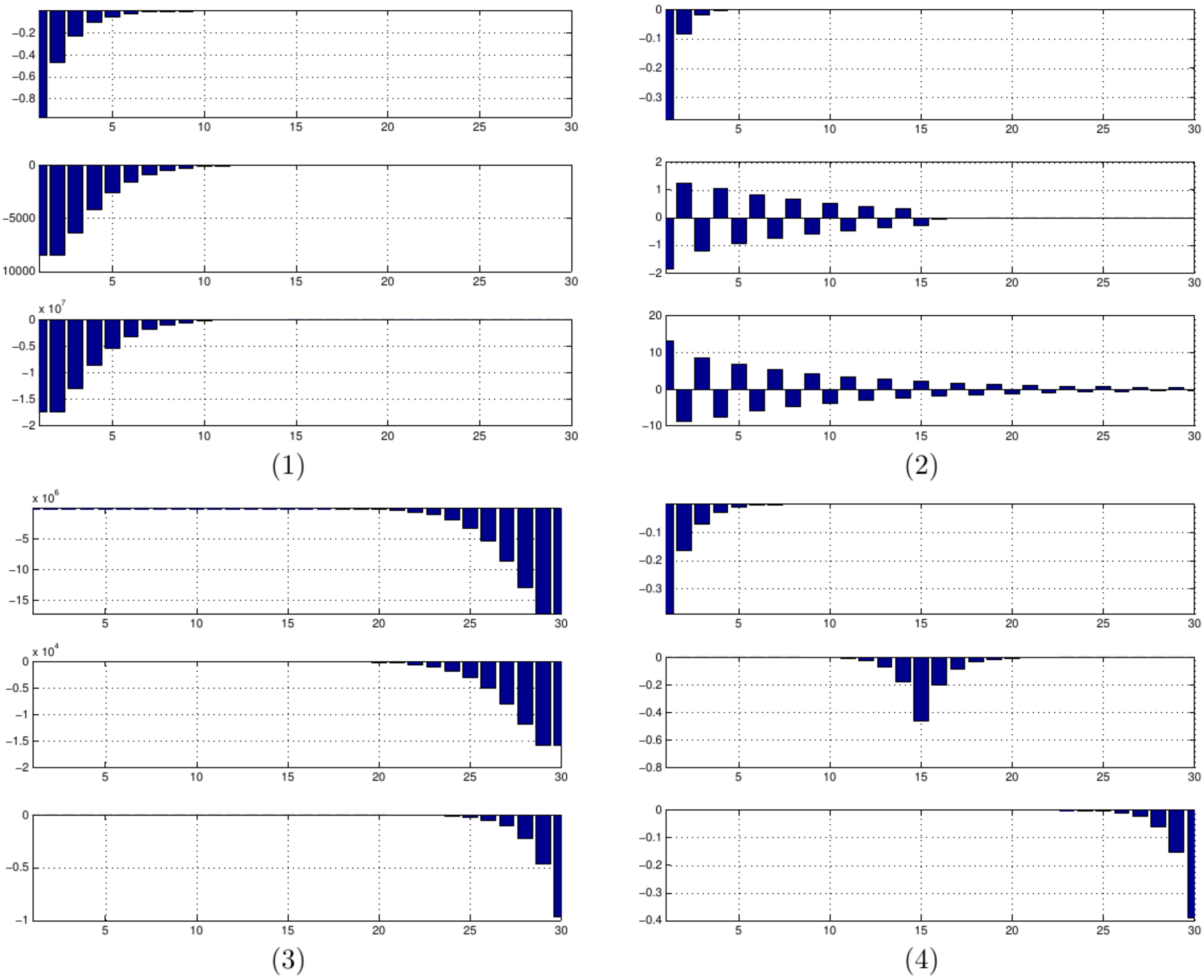
$$H_\theta = \begin{pmatrix} 0 & 1 \\ -b/c & -a/c \end{pmatrix},$$

and initial condition $x(1) = \beta(i)\,(1, a)^T$. Based on the expression of H_θ's eigenvalues,

$$\frac{\mu_+}{2\,c}, \quad \frac{\mu_-}{2\,c}.$$

one can conclude the following. 2-a) If $|\mu_+| < 2\,|c|$ and $|\mu_-| < 2\,|c|$, both eigenvalues lay inside the unit circle and the sequence $\{y^i(j)\}$, for $j = i, i+1, \ldots, n$, is decreasing. 2-b) If $|\mu_+| > 2\,|c|$ and $|\mu_-| > 2\,|c|$, at least one eigenvalue lays outside the unit circle and the sequence $\{y^i(j)\}$ is increasing. □

Theorem (5.3.1) characterises the behaviour of the sequences defined by the entries of the rows of A^{-1}. Fig. 5.1 contains graphical illustrations of the possible trends that may occur depending on the network parameters a, b, c. While Fig. 5.2 shows, for the same value of coefficients, a complete characterisation of the elements of A^{-1} through heat maps. Theorem (5.3.1) can be used to describe an optimal sensor location, and indeed to solve Problem (5.2.1), for tridiagonal Toeplitz networks. The following corollary can be established under the hypothesis that the designer has a priori knowledge of the location of the input node (see for example [78]).



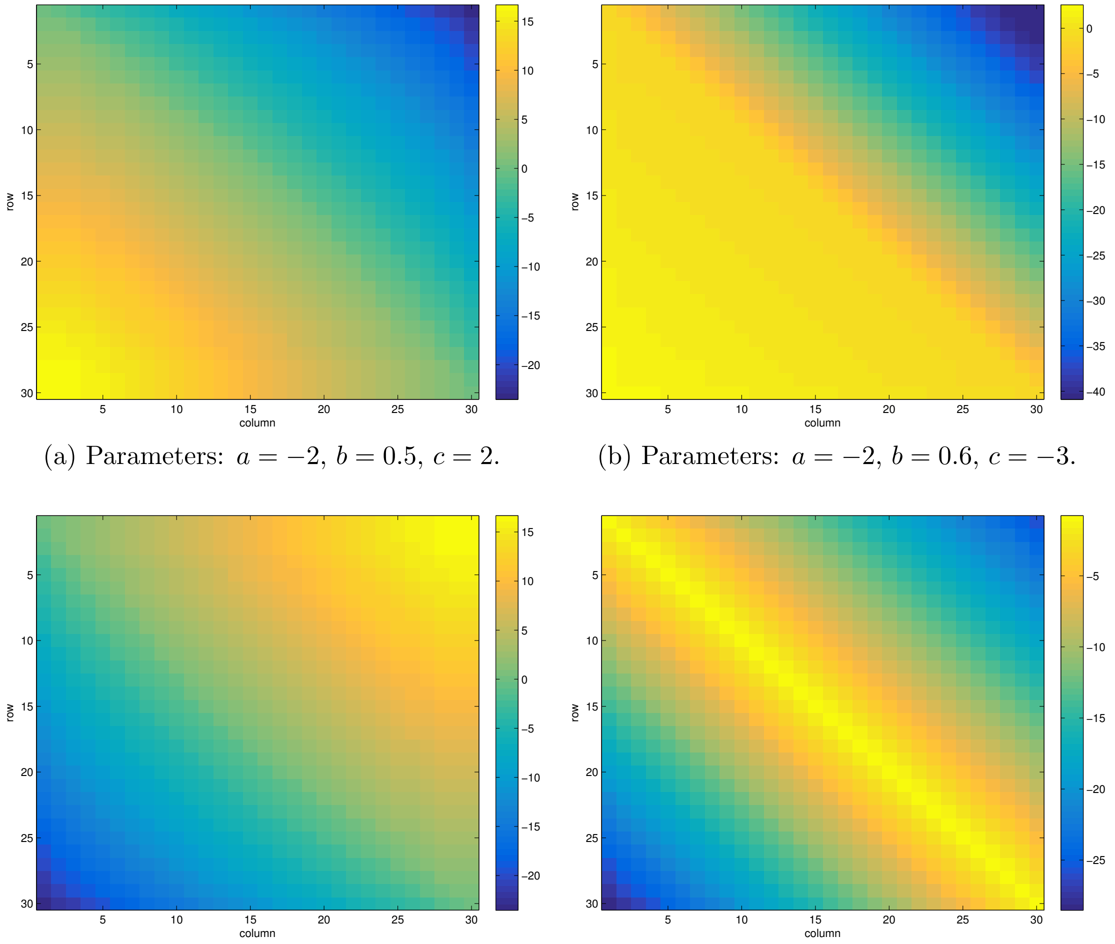
(1)

(2)

(3)

(4)

Figure 5.1: Trends of the first, middle, and last rows of the inverse of stable tridiagonal Toeplitz matrices A of dimension $n = 30$. The different behaviour are described in the following: (1) with the coefficients $a = -2$, $b = 0.5$, $c = 2$, the quantities $|\mu_-|$ and $|\mu_+|$ are less than $2|c|$ and greater than $2|b|$, thus ensuring the convergent behaviour of both the entries below the diagonal and of those above it; (2) with the coefficients $a = -2$, $b = 0.6$, $c = -3$, similar conditions of the previous case, but two eigenvalues are complex numbers; (3) with the coefficients $a = -2$, $b = 2$, $c = 0.5$, the quantities $|\mu_-|$ and $|\mu_+|$ are greater than $2|c|$ and less than $2|b|$, thus ensuring the divergent behaviour of both the entries below the diagonal and of those above it; (4) with the coefficients $a = -3$, $b = 1.1$, $c = 1$, the quantities $|\mu_-|$ and $|\mu_+|$ are greater than both $2|c|$ and $2|b|$, thus ensuring the divergence of the entries below the diagonal and the convergence of those above it.



(a) Parameters: $a = -2$, $b = 0.5$, $c = 2$.

(b) Parameters: $a = -2$, $b = 0.6$, $c = -3$.

(c) Parameters: $a = -2$, $b = 2$, $c = 0.5$.

(d) Parameters: $a = -3$, $b = 1.1$, $c = 1$.

Figure 5.2: This figure shows the heat map of the matrix $\log(|A^{-1}|)$ for different parameters of the tridiagonal Toeplitz matrix $A \in \mathbb{R}^{30\times 30}$. Notice that brighter colours corresponds to entries with larger absolute values. The figure shows different behaviours of the entries of A^{-1} as described in Theorem 5.3.1. In particular, the parameters in Fig. 5.2a and Fig. 5.2b satisfy conditions 1-b and 2-a in Theorem 5.3.1 so that, for each row, the sequence of entries is decreasing. The parameters in Fig. 4.7 satisfy conditions 1-a and 2-b in Theorem 5.3.1 so that, for each row, the sequence of entries is increasing. Finally, the parameters in Fig. 5.2d satisfy conditions 1-b and 2-b in Theorem 5.3.1 so that, for $j \geq i$, the sequence of entries is decreasing and, for $j < i$, the sequence of entries is increasing. Notice that, in all four cases, the network matrices are Hurwitz stable.

Corollary 5.3.1. ***(Optimal sensor placement for Toeplitz networks)*** *Consider a Toeplitz line network as in (5.1). If the input is located at node i, then the sensor node j such that A^{-1}_{ij} is largest is determined as follows:*

1. *If 1-a and 2-a hold, then $j = i$, that is, the sensor is co–located with the input node;*
2. *if 1-b and 2-a hold, then $j = 1$, that is, the sensor is located at the first node;*
3. *if 1-a and 2-b hold, then $j = n$, that is, the sensor is located at the last node of the line;*
4. *if 1-b and 2-b hold, then $j = h$, where h is the nearest between 1 and n to vertex i, i.e., if n is even*

$$h = \begin{cases} 1 & \text{if } i \leq \frac{n}{2}\,, \\ n & \text{if } i > \frac{n}{2}\,, \end{cases}$$

and if n is odd,

$$h = \begin{cases} 1 & \text{if } i \leq \left\lfloor \frac{n}{2} \right\rfloor\,, \\ n & \text{if } i > \left\lceil \frac{n}{2} \right\rceil\,, \\ 1 \text{ or } n & \text{if } i = \left\lceil \frac{n}{2} \right\rceil\,. \end{cases}$$

Proof. The proof trivially follows from Theorem 5.3.1. □

Remark 5.3.1. ***(Stability of the network matrix)*** *It should be observed that the stability of a tridiagonal Toeplitz matrix A does not imply the convergence or divergence of the entries of its inverse A^{-1}. In fact, Theorem 5.3.1 shows that, even for stable matrices, the rows of the inverse of the network matrix may contain convergent or divergent sequences of entries. See Section 5.4 for an example.*

5.3.2 General line networks

Let us now focus on the case of general line networks described by tridiagonal adjacency matrices of the form:

$$A = \begin{pmatrix} a_1 & b_1 & 0 & \cdots & 0 & 0 \\ c_1 & a_2 & b_2 & \cdots & 0 & 0 \\ 0 & c_2 & a_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-1} & b_{n-1} \\ 0 & 0 & 0 & \cdots & c_{n-1} & a_n \end{pmatrix}, \tag{5.10}$$

with coefficients $a_k, b_k, c_k \in \mathbb{R}_{\neq 0}$ for all k. The formula for the entries of the inverse $A^{-1} = [A^{-1}_{i,j}]$ is generalized as follows [53]:

$$A^{-1}_{ij} = \frac{1}{\theta_n} \begin{cases} (-1)^{i+j}\, b_i \cdots b_{j-1}\, \theta_{i-1}\, \phi_{j+1}, & \text{for } i \leq j\,, \\ (-1)^{i+j}\, c_j \cdots c_{i-1}\, \phi_{i+1}\, \theta_{j-1}, & \text{for } i > j\,, \end{cases} \tag{5.11}$$

where the coefficients θ_k and ϕ_k are obtained respectively through the forward and backward iterations:

$$\theta_k = a_k\, \theta_{k-1} - b_{k-1}c_{k-1}\, \theta_{k-2}\,, \text{ for } k = 2, \ldots, n\,,$$
$$\phi_k = a_k\, \phi_{k+1} - b_k c_k\, \phi_{k+2}\,, \text{ for } k = n-1, \ldots, 1\,,$$

with initial conditions respectively given by $\theta_0 = 1, \theta_1 = a_n$, and $\phi_{n+1} = 1, \phi_n = a_n$.

A complete characterization of the behaviour of the inverse A^{-1} of a general tridiagonal matrix A is not immediate, and in fact involves conditions that cannot be easily verified. However, a conservative yet useful analysis test can be derived, based on lower and upper approximations of A by suitable tridiagonal Toeplitz matrices. This fact is shown in the following result:

Theorem 5.3.2. ***(Inverse of tridiagonal matrix)*** *Let $A \in \mathbb{R}^{n\times n}$ be a tridiagonal matrix with parameters $a_i, b_i, c_i \in \mathbb{R}_{\neq 0}$. Let $a, b, c, \bar{a}, \bar{b}, \bar{c} \in \mathbb{R}_{>0}$. Assume that*

$$\begin{aligned}(\bar{a}^2 - 4\,b\,c)\,(\bar{a}^2 - 4\,b\,\bar{c}) > 0\,,\\ (a^2 - 4\,\bar{b}\,\bar{c})\,(a^2 - 4\,\bar{b}\,c) > 0\,,\end{aligned} \tag{5.12}$$

and, for all indices i,

$$a \leq |a_i| \leq \bar{a},\; b \leq |b_i| \leq \bar{b},\; c \leq |c_i| \leq \bar{c}.$$

Let $\nu_1 = \bar{a} + \sqrt{\bar{a}^2 - 4\,b\,c}$, and $\nu_2 = a - \sqrt{a^2 - 4\,\bar{b}\,\bar{c}}$.

As the dimension n grows, the entries A^{-1}_{ij} satisfy the following conditions:

1. *For all rows i,*

   (a) *if $|\nu_2| > 2\,\bar{b}$, then the sequence A^{-1}_{ij}, with $j \geq i$, is decreasing;*

   (b) *if $|\nu_1| < 2\,b$, then the sequence A^{-1}_{ij}, with $j \geq i$, is increasing.*

2. *For all rows i,*

   (a) *if $|\nu_1| < 2\,c$, then the sequence A^{-1}_{ij}, with $j < i$, is decreasing;*

   (b) *if $|\nu_2| > 2\,\bar{c}$, then the sequence A^{-1}_{ij}, with $j < i$, is increasing.*

Proof. As in Theorem 5.3.1 one can conveniently fix a row index i and study the behaviour of all column entries j for $i \leq j$ and $i > j$.

If $i \leq j$, the entry A^{-1}_{ij} can be factorized as

$$y^i(j) := A^{-1}_{ij} = \alpha'(i)\,(-1)^j\,b_i \cdots b_{j-1}\,\phi(j+1),$$

where $\alpha'(i) = (-1)^i \, \theta_n^{-1} \, \theta_{i-1}$. The entry $A^{-1}_{i,j-1}$ can be rewritten as follows:

$$\begin{aligned} y^i(j-1) &= -\alpha'(i)\,(-1)^j\, b_i \cdots b_{j-2}\, \phi(j) \\ &= -\frac{a_j}{b_{j-1}} \underbrace{\alpha'(i)\,(-1)^j\, b_i \cdots b_{j-2}\, b_{j-1}\, \phi(j+1)}_{y^i(j)} \\ &- \frac{c_j}{b_{j-1}} \underbrace{\alpha'(i)\,(-1)^{j+1}\, b_i \cdots b_{j-2}\, b_{j-1}\, b_j\, \phi(j+2)}_{y^i(j+1)}\,, \end{aligned}$$

and thus, after translating of a step backward, the following difference equation is obtained:

$$y^i(j-2) = -\frac{c_j}{b_{j-1}} y^i(j) - \frac{a_j}{b_{j-1}} y^i(j-1)\,,$$

for $j = n+1, \ldots, i+2$. One can obtain the following column-dependent, backward state form that generalizes (5.8):

$$\begin{pmatrix} y^i(j-1) \\ y^i(j-2) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\dfrac{c_j}{b_{j-1}} & -\dfrac{a_j}{b_{j-1}} \end{pmatrix} \begin{pmatrix} y^i(j) \\ y^i(j-1) \end{pmatrix}, \tag{5.13}$$

for $j = n-1, \ldots, 2$, and whose dynamic matrix has the following column–dependent eigenvalues

$$\lambda^{(j)}_{1,2} = \frac{-\,a_j \pm \sqrt{a_j^2 - 4\, b_{j-1}\, c_j}}{2\, b_{j-1}}\,.$$

If $i > j$, the entry A^{-1}_{ij} can be factorized as

$$y^i(j) := A^{-1}_{ij} = \beta'(i)\,(-1)^j\, c_j \cdots c_{i-1}\, \theta(j-1)\,,$$

where $\beta'(i) = (-1)^i \, \theta_n^{-1} \, \phi_{i+1}$, and one can obtain the following column-dependent state form that generalizes (5.9):

$$\begin{pmatrix} y^i(j+1) \\ y^i(j+2) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\dfrac{b_{j-1}}{c_j} & -\dfrac{a_j}{c_j} \end{pmatrix} \begin{pmatrix} y^i(j) \\ y^i(j+1) \end{pmatrix}, \tag{5.14}$$

for $j = 1, \dots, i-1$, and whose dynamic matrix has the following column-dependent eigenvalues

$$\mu_{1,2}^{(j)} = \frac{-\,a_j \pm \sqrt{a_j^2 - 4\,b_{j-1}\,c_j}}{2\,c_j}.$$

Under the conditions in (5.12), it is possible to find the following upper and lower bounds for the modules of the eigenvalues $\lambda_{1,2}^{(j)}$ and $\mu_{1,2}^{(j)}$:

$$\frac{|\nu_2|}{2\,\bar{b}} \le |\lambda_{1,2}^{(j)}| \le \frac{|\nu_1|}{2\,b}, \quad \frac{|\nu_2|}{2\,\bar{c}} \le |\mu_{1,2}^{(j)}| \le \frac{|\nu_1|}{2\,c}.$$

As in Theorem 5.3.1, when both eigenvalues $\lambda_{1,2}^{(j)}$ lay inside the unit circle, for all j, the backward dynamics in (5.13) is decreasing and the forward sequence $\{y^i(j)\}$ is increasing, while, when both eigenvalues $\mu_{1,2}^{(j)}$ lay inside the unit circle, for all j, the dynamics in (5.14) is decreasing and the same holds for the sequence $\{y^i(j)\}$. This reasoning explains the conditions in the statement of the theorem. □

As for the case of Toeplitz line networks discussed in Theorem 5.3.1, Theorem 5.3.2 provides guidelines for the selection of sensor nodes with respect to the input location and the network parameters.

5.4 Applications

In this section we validate our findings with three different areas of application. In particular, in Section 5.4.1 through numerical examples we show that by fixing the diagonal entries of network matrix and plotting the regions of the parameters space, one can yield decreasing and increasing behaviours of the entries of the network inverse matrix. Instead, in Section 5.4.2 we present an electronic network whose dynamics are described by a tridiagonal matrix, and we analyse its steady state behaviour as a function of the network elements and the locations of the input and sensor nodes. Finally, in Section 5.4.3 we exploit the theoretical results obtained before in Section 5.3 to solve the problem of detect the disturbance propagation in a vehicular platoon.

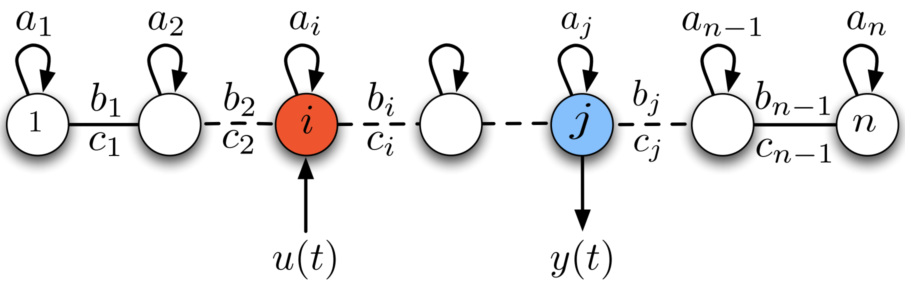


Figure 5.3: Line network with n nodes. The i–th node is affected by an attack, and the j–th node represents the optimal sensor placement.

5.4.1 Map of decreasing and increasing parameters

In this subsection we present numerical examples of line networks with n vertex nodes (Fig. 5.3), where a node i is affected by a disturbing signal $u(t)$, and it is required to find, by means of Theorems 5.3.1 and 5.3.2, the sensor location j maximizing the ability of an observer to detect such disturbance. Consider the case of a stable Toeplitz line network with $n = 30$, $a_k = a = -0.3$, for all k, while the other two coefficients $b_k = b$ and $c_k = c$, for all k, are still to be chosen. Suppose that a white noise signal $u(t)$, with zero mean and unit variance, is injected through the first node, i.e. $i = 1$. Based on Theorem 5.3.1,

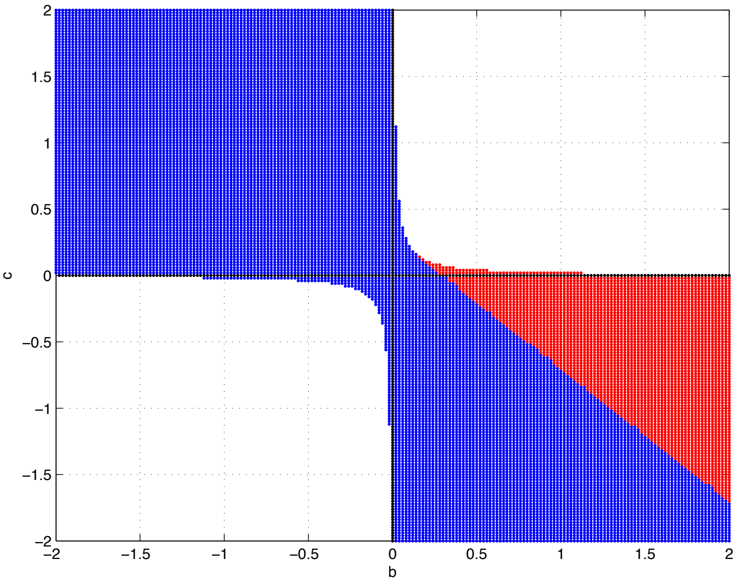


Figure 5.4: Map of convergence and divergence for fixed value $a = -0.3$ and for $-2 \leq b, c \leq 2$. Blue and red indicate regions where the entries A^{-1}_{ij} of the inverse A^{-1} have convergent and divergent behaviours, respectively. White indicate regions where the network is unstable. The axes $b = 0$ and $c = 0$ are excluded since they make A not invertible.

one can build a map telling for different values of b and c whether the behaviour of the first row of the inverse A^{-1} is convergent or divergent (Fig. 5.4).

Assume first the numerical values $b = 1$ and $c = -0.25$. The map reveals that the behaviour of the entries of the first row of A^{-1} is divergent, and thus the optimal sensor location is at the furthest vertex from the attack, i.e. $j = 30$. This result is counterintuitive as one can think that the optimal choice to detect an attack is to place the sensor at the same node where the attack is performed.

In fact, the adopted metric, the trace of the cross–Gramian W_{co}, reaches its maximum at the last node, which implies that the corresponding output signal is more detectable when measured at that location. As a second example, assume the numerical values $b = 0.57$ and $c = -1.37$; based on the map the optimal placement is obtained by co-locating the sensor at the vertex where the disturbance is originated, that is $j = i = 1$. Fig. 5.5 shows the amplification of the signal $u(t)$ for the two cases above, and it validates

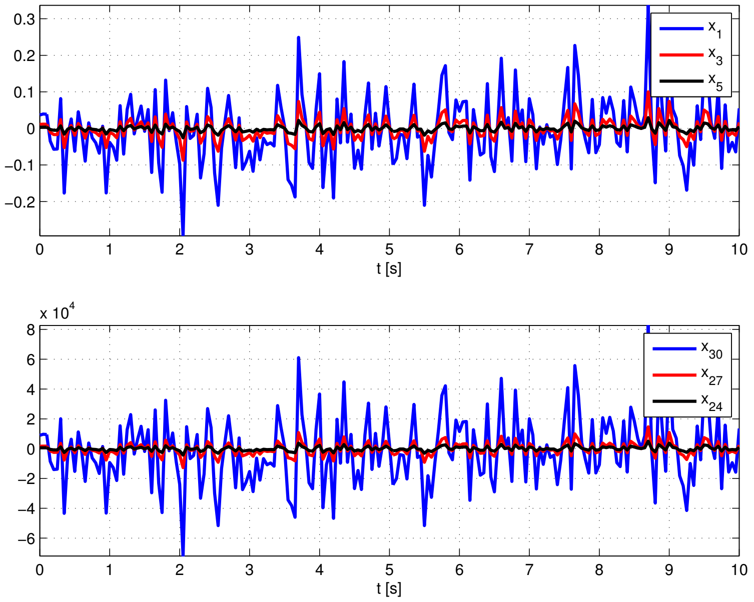


Figure 5.5: Simulation runs with two Toeplitz line networks belonging to case (3) and (1) of Fig. 5.2, respectively. In the top figure the maximum amplification of the input $u(t)$ is achieved at furthest nodes, while in the bottom figure it is obtained at same node where $u(t)$ is applied.

the results expected from the theorems.

5.4.2 An electronic network yielding tridiagonal dynamics

Consider the electric circuit in Fig. 5.6, which represents a multistage amplifier [82], and consists of a chain of RC elements connected by voltage–feedback transconductance amplifiers. The i-th part of the system, for $i = 1, \ldots, n$, includes a resistor R_i, a capacitor

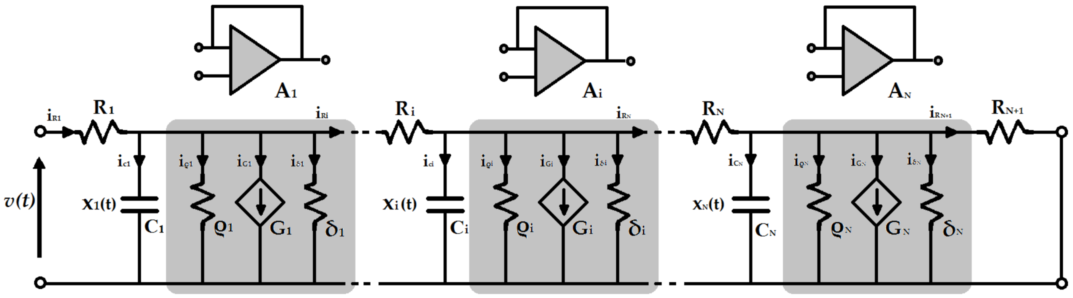


Figure 5.6: RC–chain network with n voltage–feedback transconductance amplifiers A_i, which are depicted in the gray boxes.

C_i, and the elements of the local amplifier's equivalent circuit. Each amplifier A_i comprises an input resistance, ρ_i, an output resistance, δ_i, and a controlled generator G_i injecting a current linearly depending on the capacitor C_i's voltage through a gain coefficient k_i. Let $x_i(t)$ be the voltage at time t of the capacitor C_i. Then, the current injected by G_i is $k_i\, x_i(t)$, where $k_i > 0$ if the amplifier works in non-inverting configuration, and $k_i < 0$ if the amplifier is in inverting configuration. The electric circuit is controlled through an input signal $v(t)$ that is applied at the first RC-branch, while a short circuit is imposed at the last branch. We aim to characterise the propagative properties of a disturbance signal $u(t)$ that is applied at the i-th node. To this aim, we first determine a dynamical model of the system. By applying Kirchhoff's current law, we obtain the following balance equations:

$$i_{R_i}(t) - i_{C_i}(t) - i_{\rho_i} - i_{G_i}(t) - i_{\delta_i} - i_{R_{i+1}}(t) = 0\,,$$

for $i = 1, 2, \ldots, n$, which can be written in terms of the system's state by exploiting Kirchhoff's voltage law:

$$\frac{x_{i-1}(t) - x_i(t)}{R_i} - C_i\, \dot{x}_i(t) - \frac{x_i(t)}{\rho_i} - k_i\, x_i(t) - \frac{x_i(t)}{\delta_i} - \frac{x_i(t) - x_{i-1}(t)}{R_{i+1}} = 0\,.$$

The network dynamical model is thus given by

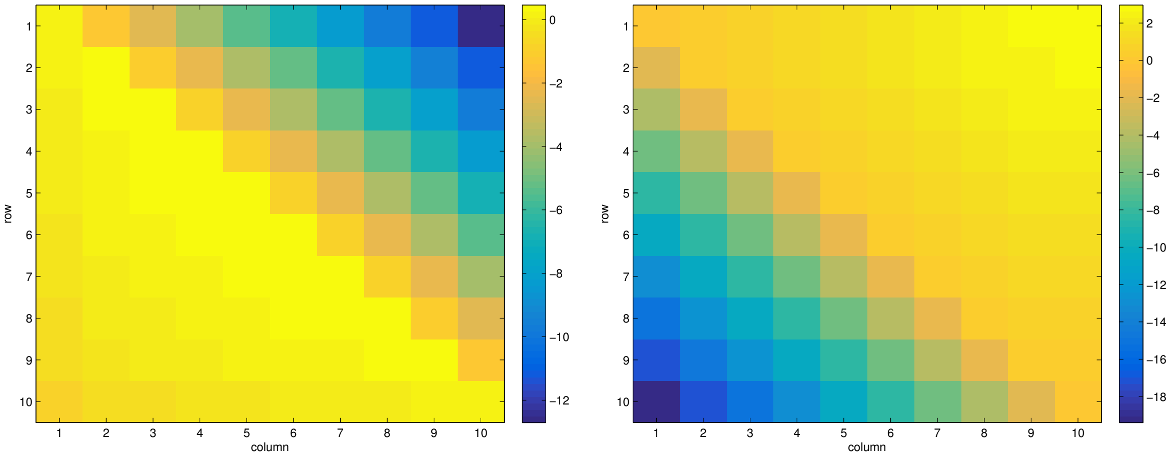
$$\dot{x}_i(t) = c_i\, x_{i-1}(t) + a_i\, x_i(t) \; + \; b_i\, x_{i+1}(t)\,, \tag{5.15}$$

where

$$\begin{aligned} c_i &= \frac{1}{R_i C_i} > 0 \quad b_i = \frac{1}{R_{i+1} C_i} > 0\,, \\ a_i &= -\frac{1}{C_i}\left(\frac{1}{\rho_i} + \frac{1}{\delta_i} + \frac{1}{R_i} + \frac{1}{R_{i+1}} - k_i\right) = -b_i - c_i - \frac{1}{\rho_i\, C_i} - \frac{1}{\delta_i\, C_i} + \frac{k_i}{C_i}\,, \end{aligned} \tag{5.16}$$

for $i = 1, \dots, n$, $x_0(t) = u(t)$ and $x_{n+1}(t) = 0$. Thus, the electric circuit in Fig. 5.6 can be modelled as the line network system in Fig. 5.3 with dynamics of the form (5.1), where the parameters a_i, b_i, and c_i are given in (5.16). We now study two numerical instances of the above electric circuit with $n = 10$.

For simplicity, we let all input and output resistances of the amplifiers have equal value, that is, $\rho_i = \rho = 1 \cdot 10^6\,\Omega$ and $\delta_i = \delta = 3 \cdot 10^6\,\Omega$. By choosing the other circuit parameters as in Table 5.1-a), the network parameters are such that the network matrix A is Toeplitz and satisfies the conditions 1-b and 2-b in Theorem 5.3.1. Thus, the sequences A^{-1}_{ij}, for $j \geq i$ (respectively $j < i$), show a decreasing (respectively increasing) behaviour. Instead, if the circuit parameters are chosen as in Table 5.1-b), the network matrix A satisfies the conditions 1-a and 2-b in Theorem 5.3.1, so that the sequences A^{-1}_{ij} are increasing for all rows i. A graphical illustration of the behaviours of the entries of the inverse of the network matrix is reported in Fig. 5.7. As a consequence of this analysis, for the first set of parameters, a disturbance is mostly visible close to the signal source. Instead, for the latter set of parameters, the effect of a disturbance is greatest at the right extreme of the circuit.



(a) Parameters: $a = -1$, $b = 0.2$, $c = 0.75$.

(b) Parameters: $a = -1$, $b = 1.2$, $c = 0.1$.

Figure 5.7: This figure shows the heat map of the matrix $\log(|A^{-1}|)$ for different parameters of the tridiagonal Toeplitz matrix $A \in \mathbb{R}^{10\times 10}$ as obtained in Section 5.4.2 from the network parameters in Table 5.1. The parameters in Fig. 5.7a satisfy conditions 1-b and 2-b in Theorem 5.3.1 so that, for $j \geq i$, the sequence of entries is decreasing and, for $j < i$, the sequence of entries is increasing. The parameters in Fig. 5.7b satisfy conditions 1-a and 2-b in Theorem 5.3.1 so that, for all rows, the sequence of entries is increasing. Notice that, in both cases, the network matrices are Hurwitz stable.

Table 5.1: Network parameters yielding decreasing (left) and increasing (right) behaviours of A^{-1}.

i	R_i [Ω]	C_i [F]	k_i
1	$1 \cdot 10^0$	$1.33 \cdot 10^0$	$-6.67 \cdot 10^{-2}$
2	$3.75 \cdot 10^0$	$3.56 \cdot 10^{-1}$	$-1.78 \cdot 10^{-2}$
3	$1.41 \cdot 10^1$	$9.48 \cdot 10^{-2}$	$-4.74 \cdot 10^{-3}$
4	$5.27 \cdot 10^1$	$2.53 \cdot 10^{-2}$	$-1.26 \cdot 10^{-3}$
5	$1.98 \cdot 10^2$	$6.74 \cdot 10^{-3}$	$-3.36 \cdot 10^{-4}$
6	$7.42 \cdot 10^2$	$1.8 \cdot 10^{-3}$	$-8.86 \cdot 10^{-5}$
7	$2.78 \cdot 10^3$	$4.79 \cdot 10^{-4}$	$-2.26 \cdot 10^{-5}$
8	$1.04 \cdot 10^4$	$1.28 \cdot 10^{-4}$	$-5.06 \cdot 10^{-6}$
9	$3.91 \cdot 10^4$	$3.41 \cdot 10^{-5}$	$-3.71 \cdot 10^{-7}$
10	$1.47 \cdot 10^5$	$9.09 \cdot 10^{-6}$	$8.79 \cdot 10^{-7}$
10	$5.5 \cdot 10^5$		
	$a = -1$	$b = 0.2$	$c = 0.75$

(a)

i	R_i [Ω]	C_i [F]	k_i
1	$1 \cdot 10^{10}$	$1 \cdot 10^{-9}$	$1.33 \cdot 10^{-6}$
2	$8.33 \cdot 10^8$	$1.2 \cdot 10^{-8}$	$1.34 \cdot 10^{-6}$
3	$6.94 \cdot 10^7$	$1.44 \cdot 10^{-7}$	$1.38 \cdot 10^{-6}$
4	$5.79 \cdot 10^6$	$1.73 \cdot 10^{-6}$	$1.85 \cdot 10^{-6}$
5	$4.82 \cdot 10^5$	$2.07 \cdot 10^{-5}$	$7.55 \cdot 10^{-6}$
6	$4.02 \cdot 10^4$	$2.49 \cdot 10^{-4}$	$7.6 \cdot 10^{-5}$
7	$3.35 \cdot 10^3$	$2.99 \cdot 10^{-3}$	$8.97 \cdot 10^{-4}$
8	$2.79 \cdot 10^2$	$3.58 \cdot 10^{-2}$	$1.08 \cdot 10^{-2}$
9	$2.33 \cdot 10^1$	$4.3 \cdot 10^{-1}$	$1.29 \cdot 10^{-1}$
10	$1.94 \cdot 10^0$	$5.16 \cdot 10^0$	$1.55 \cdot 10^0$
10	$1.62 \cdot 10^{-1}$		
	$a = -1$	$b = 1.2$	$c = 0.1$

(b)

5.4.3 Asymmetric bidirectional control of vehicle platoons

Distributed policies for vehicle platooning can potentially increase the safety and capacity of highways and, at the same time, can allow drivers to relax during a travel [45] or the daily commute. In the context of automated vehicular control, various motion control mechanisms and stability properties of the formation have been studied and described in the literature [91]. The problem in its most basic form is to steer a collection of vehicles from an initial point to another one with certain desired speed and such that each vehicle should keep some preset distance respect to the predecessor vehicle and the following [83]. The simplest approach to be implemented, among the ones based on fixed-distance, is the *predecessor-following* approach, according to which every vehicle is equipped with two sets of sensors, measuring the relative distances from the preceding vehicle and the following vehicle. Therefore, bidirectional symmetric-control or asymmetric-control can be implemented. In all such strategies, scalability in terms of disturbance propagation is a phenomenon known as *string instability* could be avoided [67]. String instability occurs when a disturbance or measurement error acting at a given vehicle is amplified as it is propagated along the platoon of vehicles.

While symmetric control has been well studied and symmetric systems can be made string stable, though at the expense of tolerating long transients, asymmetric control is only recently receiving a wide attention from many authors [44, 91]), for its interesting properties in terms of convergence speed and controllability [39, 72]. However, the asymmetric control leads to an undesired behaviour of the entire system, which is referred to as *harmonic instability*, namely the phenomenon occurring when the peak of the magnitude frequency response exponentially grows with the number of vehicles. In [91] authors has shown how the asymmetric systems are even harmonically unstable.

The theoretical results obtained in Section 5.3 find also application in the aforementioned context of harmonic instability for homogeneous vehicles platoon. To this re-

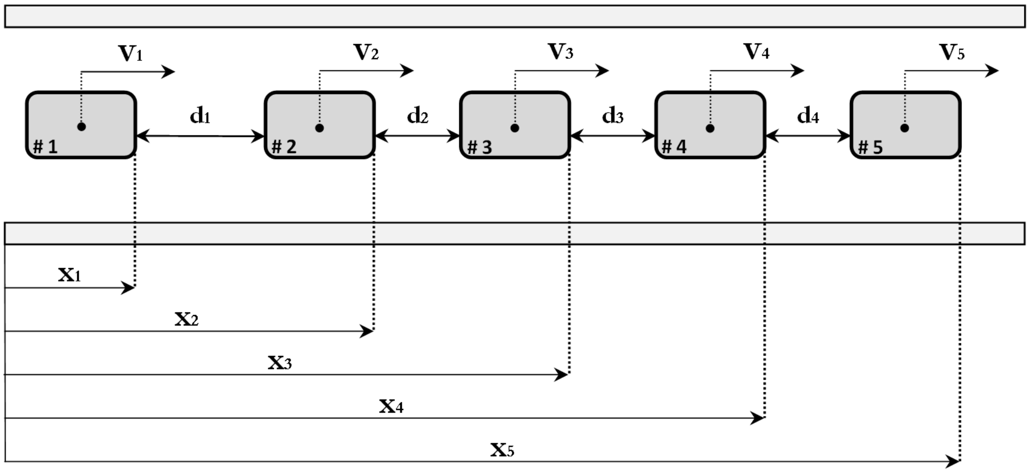


Figure 5.8: Example of platoon composed by 5 vehicles. Each car $\#_i$ with position x_i is moving in one dimension with velocity v_i. Every vehicle in the platoon should automatically keep the distances d_i respect to the predecessor $\#_{i-1}$ and the following vehicle $\#_{i+1}$.

gard, [56] in this framework authors want to show an interesting scenario where a disturbance propagation along finite string vehicles can be detect. Indeed, given as metric the cross-Gramian, it will be shown how the disturbance injected i.e by the tail string could be detected by an optimal design of network parameters. In this way the leading vehicle informations will not be affected by the error.

Consider a generic system of N vehicles such as represented in Fig. 5.8 (for the particular case of 5 vehicles) described by the second-order dynamic model

$$\begin{aligned} \dot{x}_i &= v_i\,, \\ \dot{v}_i &= u_i\,, \end{aligned}$$

where u_i, for $i = 1, \ldots, N$, are the normalised inputs of each vehicle, x_i and v_i are their position and velocity. In general, centralised control is impractical for medium to large sized platoons; thus a decentralised controller approach should be used. Therefore, we want to find a decentralised law $u = (u_1, \ldots, u_N)^T$ s.t. all vehicles asymptotically move at a desired speed $\bar{v}$, i.e.

$$v_i \to \bar{v}\,,$$

and according to the *asymmetric* bidirectional control policy [67], it maintain a relative distance $\bar{d}$, i.e .

$$d_i = x_{i-1} - x_i \to \bar{d}\,, \quad \text{for } i = 2, \ldots, N\,.$$

Decentralized means that each vehicle's input u_i must depend only on the information of its own position and velocity and that of its preceding and following vehicle. Let $\tilde{d}_i = \bar{d} - d_i$ and $\tilde{v}_i = \bar{v} - v_i$. A possible solution is the following

$$u_i = -(k_1 + k_2)\,\tilde{v}_i + k_1\,\tilde{v}_{i-1} + k_2\,\tilde{v}_{i+1} + \lambda\,\tilde{d}_i\,,$$

for $i = 2, \ldots, N-1$, and

$$u_1 = -k_2\,\tilde{v}_1 + k_2\,\tilde{v}_2\,,$$

$$u_N = -k_1\,\tilde{v}_N + k_1\,\tilde{v}_{N-1} + \lambda\,\tilde{d}_N\,.$$

The system model for $N = 3$ is the following:

$$\begin{aligned}
\dot{\tilde{v}}_1 &= -k_2\,\tilde{v}_1 + k_2\,\tilde{v}_2\,,\\
\dot{\tilde{v}}_2 &= -(k_1 + k_2)\,\tilde{v}_2 + k_1\,\tilde{v}_1 + k_2\,\tilde{v}_3 + \lambda\,\tilde{d}_2\,,\\
\dot{\tilde{v}}_3 &= -k_1\,\tilde{v}_3 + k_1\,\tilde{v}_2 + \lambda\,\tilde{d}_3\,,\\
\dot{\tilde{d}}_2 &= v_1 - v_2\,,\\
\dot{\tilde{d}}_3 &= v_2 - v_3\,,
\end{aligned}$$

It is linear and it thus can be represented as

$$\begin{pmatrix} \dot{\tilde{v}}_1 \\ \dot{\tilde{v}}_2 \\ \dot{\tilde{v}}_3 \\ \dot{\tilde{d}}_2 \\ \dot{\tilde{d}}_3 \end{pmatrix} = A \begin{pmatrix} \tilde{v}_1 \\ \tilde{v}_2 \\ \tilde{v}_3 \\ \tilde{d}_2 \\ \tilde{d}_3 \end{pmatrix} + B \left(\begin{array}{c} 0 \\ 0 \\ 0 \\ \hline \bar{d} \\ \bar{d} \end{array}\right)$$

where

$$A = \left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right) = \left(\begin{array}{ccc|cc} -k_2 & k_2 & 0 & 0 & 0 \\ k_1 & -(k_1+k_2) & k_2 & \lambda & 0 \\ 0 & k_1 & -k_1 & 0 & \lambda \\ \hline 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \end{array}\right)$$

and B equals the identity matrix I.

Without considering the contribute of the leading and the tail vehicle (namely, if we neglect the first and last row of A), the sub–network matrix A_{11} can be considered as a Tridiagonal Toeplitz with coefficients: $a = -(k_1 + k_2)$, $b = k_2$ and $c = k_1$. That makes sense when the number of vehicles grows-up enough; thus the behaviour of the platoon can be described by a line network (as shown before in Fig. 5.3), where each node corresponds to a single vehicle. Consider a platoon composed of 20 vehicles and supposing an external disturbance (i.e. deceleration signal caused by internal vehicle fault or non-uniform road surface) acts into vehicle 2; we want to find the optimal placement to detect the attack. Given the results obtained in Theorem 3.1, by appropriately selecting $k_1 = 0.1$ and $k_2 = 0.2$, we have that the first eigenvalue $\sigma_1 > 1$, while the other eigenvalue σ_2 lays inside on the unit circle. Therefore, the entry $A^{-1}_{i,j}$ have decreasing behaviour as illustrated in Fig. 5.9, than according to the case 1-a) of Corollary 5.3.1, the optimal

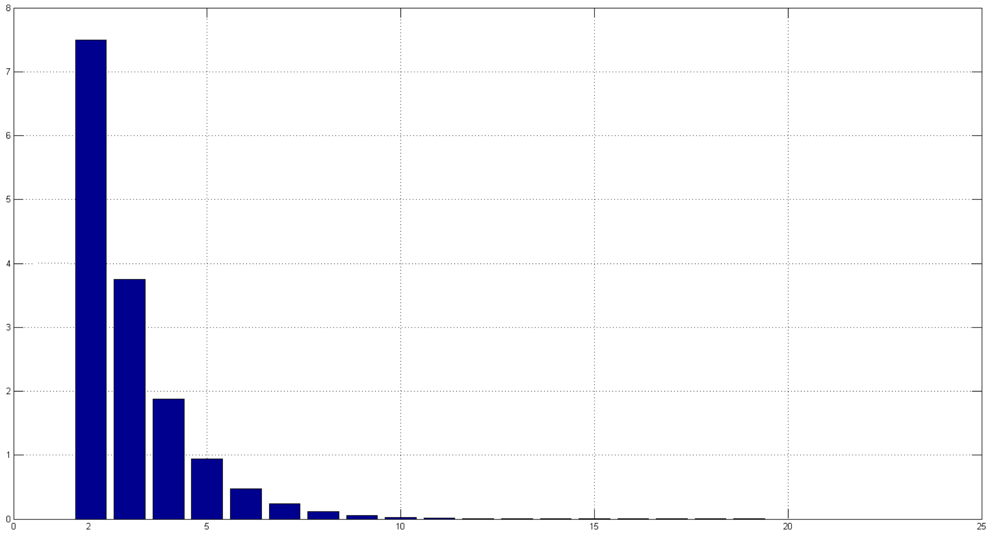


Figure 5.9: Decreasing behaviour of the second row of $A_{i,j}^{-1}$. According to Corollary 5.3.1, the optimal sensor placement will be for $j = 2$, where the metric value is the biggest.

sensor placement will be for $j = i$ (sensor co-located with the attacker). In this way, one can establish exactly which vehicle of the platoon has decelerated, in order to decide which control law must be chosen to achieve the desired speed $\bar{v}$.

Conclusion and Future Work

This thesis work has mainly focused on (i) analysing the convergence of multi-agent systems by considering their complex dynamic behaviour through the consensus model and the HK model, (ii) addressing the well-known data clustering problem with additional distance centroid constraints for a wireless network application, (iii) and dealing the problem of attack detection through optimal sensor placement in line network systems.

In Chapter 3 we have given centralised and distributed conditions to determine, at the generic instant t, if a system described by the HK opinion dynamic model can eventually reach an agreement state or, more in general, a steady state (based on the doubly stochasticity of the dynamic matrix $A(t)$). Furthermore, we have also given conditions about the connectedness of the graph at time instant $t+1$ and have investigated on the feasibility of a local convergence criterion, which is valid except under the cases when the graph contains m-paths. Through an alternative formulation of the HK model insights on its complexity are found, which suggests a way to study the detection of the agreement condition. The proposed criteria represent a method to determine whether the a set of agents, interacting via the HK model, can stop. To show the effectiveness of the proposed criteria, simulation results have been shown which successfully confirm their validity. Future works in this topic will be along two main directions. First, we will inspect a generalisation of the HK model to a n-dimensional vector opinion problems and study the finite-time convergence to a steady state, in order to characterise an upper bound for the time instant when a steady state is reached. Secondly, we will use the alternative formalisation of the model provided in Section 3.1 to inspect the possibility of finding a closed-form solution to the HK model and determine conditions for reaching a steady state and agreement, depending only on the initial state $x(0)$ and the confidence interval length ε.

Chapter 4 described a distributed algorithm to solve a data clustering for a sensors network, with the constraint that cluster centroids must be minor than ε. The proposed approach is two steps: first, an admissible, although sub-optimal, solution of the problem is found by using an HK model "filter", which requires no a-priori specification by the user of the number of clusters, and then the found solution is optimised by exploiting the k–means algorithm. In spite of the HK model in its original application is a centralised algorithm, we proposed a new distributed implementation based on a combination of consensus algorithms. Future research will focus on proving the convergence of HK model in finite time by considering vectorial opinions and testing the algorithm in a real world scenario, including noise and packet loss.

In Chapter 5 we studied the problem of selecting sensor nodes for optimal signal detection in network systems. We adopted the trace of the cross–Gramian or, equivalently, for single-input single-output systems, the static gain of the network system, to evaluate different sensor positions with respect to the origin of a signal to be detected. For the class of line networks characterised by Toeplitz dynamic matrices or, more in general, by tridiagonal dynamic matrices, we have shown that the entries of the inverse of the network matrix can exhibit drastically different behaviours. Consequently, to maximise the detection performance, the sensor should either be co-located with the origin of the signal, or as far as possible from it, depending on the network parameters. We illustrated our findings through a synthetic example and a class of electrical circuits. Several problems are left as the subject of future research, including the extension to multi-input multi-sensor scenarios and the study of different network topologies.

Bibliography

[1] Chaouki T Abdallah and Herbert G Tanner. Complex networked control systems. 2007.

[2] R. Anguluri, R. Dhal, S. Roy, and F. Pasqualetti. Network invariants for optimal input detection. Boston, MA, USA, 2015. Submitted.

[3] Jon L Bentley, Donald F Stanat, and E Hollins Williams Jr. The complexity of finding fixed-radius near neighbors. *Information processing letters*, 6(6):209–212, 1977.

[4] V.D. Blondel, J.M. Hendrickx, and J.N. Tsitsiklis. On Krause's multi-agent consensus model with state-dependent connectivity. *IEEE Transactions on Automatic Control*, 54(11):2586–2597, 2009.

[5] Vincent Blondel, Julien M Hendrickx, Alex Olshevsky, J Tsitsiklis, et al. Convergence in multiagent coordination, consensus, and flocking. *IEEE Conference on Decision and Control (CDC)*, volume 44, page 2996, 2005.

[6] Vincent D Blondel, Julien M Hendrickx, and John N Tsitsiklis. Opinion dynamics for agents with opinion-dependent connections. *49th IEEE Conference on Decision and Control (CDC)*, pages 6626–6632, 2010.

[7] Stefano Boccaletti, Vito Latora, Yamir Moreno, Martin Chavez, and D-U Hwang. Complex networks: Structure and dynamics. *Physics reports*, 424(4):175–308, 2006.

[8] Katy Börner, Soma Sanyal, and Alessandro Vespignani. Network science. *Annual review of information science and technology*, 41(1):537–607, 2007.

[9] PS Bradley, KP Bennett, and Ayhan Demiriz. Constrained k-means clustering. *Microsoft Research, Redmond*, pages 1–8, 2000.

[10] James M Brase and David L Brown. Modeling, simulation and analysis of complex networked systems. *Lawrence Livermore National Laboratory, Tech. Rep*, 2009.

[11] Peter Brucker. On the complexity of clustering problems. *Optimization and operations research*, pages 45–54. Springer, 1978.

[12] Sergey V Buldyrev, Roni Parshani, Gerald Paul, H Eugene Stanley, and Shlomo Havlin. Catastrophic cascade of failures in interdependent networks. *Nature*, 464(7291):1025–1028, 2010.

[13] Duncan S Callaway, Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Network robustness and fragility: Percolation on random graphs. *Physical review letters*, 85(25):5468, 2000.

[14] Claudio Canuto, Fabio Fagnani, and Paolo Tilli. An eulerian approach to the analysis of krause's consensus models. *SIAM Journal on Control and Optimization*, 50(1):243–265, 2012.

[15] Alvaro A Cárdenas, Saurabh Amin, and Shankar Sastry. Research challenges for the security of control systems. *HotSec*, 2008.

[16] Francesca Ceragioli and Paolo Frasca. Continuous and discontinuous opinion dynamics with bounded confidence. *Nonlinear Analysis: Real World Applications*, 13(3):1239–1251, 2012.

[17] Dah-Ming Chiu and Raj Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN systems*, 17(1):1–14, 1989.

[18] I. Constantin Morarescu and A. Girard. Opinion Dynamics with Decaying Confidence: Application to Community Detection in Graphs. *IEEE Transactions on Automatic Control*, 2010.

[19] Paolo Crucitti, Vito Latora, Massimo Marchiori, and Andrea Rapisarda. Efficiency of scale-free networks: error and attack tolerance. *Physica A: Statistical Mechanics and its Applications*, 320:622–642, 2003.

[20] György Dán and Henrik Sandberg. Stealth attacks and protection schemes for state estimators in power systems. *IEEE First International Conference on Smart Grid Communications*, pages 214–219, 2010.

[21] Ian Davidson and SS Ravi. Clustering with constraints: Feasibility issues and the k-means algorithm. *SDM*, volume 5, pages 201–211. SIAM, 2005.

[22] Pietro DeLellis, Franco Garofalo, Davide Liuzza, et al. Analysis and stability of consensus in networked control systems. *Applied Mathematics and Computation*, 217(3):988–1000, 2010.

[23] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[24] Rahul Dhal and Sandip Roy. Vulnerability of continuous-time network synchronization processes: A minimum energy perspective. pages 823–828, 2013.

[25] L Di Paola, M De Ruvo, P Paci, D Santoni, and A Giuliani. Protein contact networks: an emerging paradigm in chemistry. *Chemical reviews*, 113(3):1598–1613, 2012.

[26] Jan Christian Dittmer. Consensus formation under bounded confidence. *Nonlinear Analysis-Theory Methods and Applications*, 47(7):4615–4622, 2001.

[27] Daniel M Dunlavy, Bruce Hendrickson, and Tamara G Kolda. Mathematical challenges in cybersecurity. *Sandia Report SAND*, 805, 2009.

[28] Joseph C Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters, 1973.

[29] Adriano Fagiolini and Antonio Bicchi. On the robust synthesis of logical consensus algorithms for distributed intrusion detection. *Automatica*, 49(8):2339–2350, 2013.

[30] Hamza Fawzi, Paulo Tabuada, and Suhas Diggavi. Secure state-estimation for dynamical systems under active adversaries. *49th IEEE Annual Allerton Conference on Communication, Control, and Computing*, pages 337–344, 2011.

[31] K.V. Fernando and H. Nicholson. On the structure of balanced and other principal representations of siso systems. *IEEE Transactions on Automatic Control*, 28(2):228–231, 1983.

[32] J. R. P. French. A formal theory of social power. *Psychological Review*, 63:181–194, 1956.

[33] Federica Garin and Luca Schenato. A survey on distributed estimation and control applications using linear consensus algorithms. *Networked Control Systems*, pages 75–107. Springer, 2010.

[34] Audrey P Gasch and Michael B Eisen. Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. *Genome Biol*, 3(11):1–22, 2002.

[35] Andrea Gasparri and Gabriele Oliva. Fuzzy opinion dynamics. *IEEE American Control Conference (ACC)*, pages 5640–5645, 2012.

[36] D. Georges. The use of observability and controllability Gramians or functions for optimal sensor and actuator location in finite-dimensional systems. 34th *IEEE Conference on Decision and Control (CDC)*. New Orleans, LA, USA, 1995.

[37] Semyon Aranovich Gershgorin. Uber die abgrenzung der eigenwerte einer matrix. *Izvestija Rossijskoj akademii nauk. Serija matematiceskaja*, (6):749–754, 1931.

[38] M. H. De Groot. *Reaching a consensus*, volume 69. Wiley, 1974.

[39] He Hao, Huibing Yin, and Zhen Kan. On the robustness of large 1-d network of double integrator agents. *IEEE American Control Conference (ACC)*, pages 6059–6064, 2012.

[40] Peter Hegarty and Edvin Wedin. The Hegselmann-Krause dynamics for the continuous-agent model and a regular opinion function do not always lead to consensus. 2014.

[41] Rainer Hegselmann and Ulrich Krause. Opinion dynamics and bounded confidence models, analysis, and simulation. *Journal of Artifical Societies and Social Simulation (JASSS) vol*, 5(3), 2002.

[42] Rainer Hegselmann and Ulrich Krause. Opinion dynamics driven by various ways of averaging. *Computational Economics*, 25(4):381–405, 2005.

[43] Rainer Hegselmann and Ulrich Krause. Opinion dynamics under the influence of radical groups, charismatic leaders, and other constant signals: A simple unifying model. *Networks and Heterogeneous Media*, 10(3):477–509, 2015.

[44] Ivo Herman, Dan Martinec, Zdenek Hurák, and Michael Sebek. Harmonic instability of asymmetric bidirectional control of a vehicular platoon. *IEEE American Control Conference (ACC)*, pages 5396–5401, 2014.

[45] Ivo Herman, Dan Martinec, Zdeněk Hurák, and Michael Sebek. Scaling of transfer functions in vehicular platoons: the role of asymmetry disputed. *arXiv preprint arXiv:1410.3943*, 2014.

[46] Christian Himpe and Mario Ohlberger. Cross–gramian–based combined state and parameter reduction for large-scale control systems. *Mathematical Problems in Engineering*, 2014.

[47] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[48] Anil K Jain and Richard C Dubes. *Algorithms for clustering data.* Prentice-Hall, Inc., 1988.

[49] T. Kailath. *Linear Systems.* Prentice-Hall, 1980.

[50] John G Klincewicz. Hub location in backbone/tributary network design: a review. *Location Science*, 6(1):307–335, 1998.

[51] Sascha Kurza and J org Rambaua. On the Hegselmann-Krause conjecture in opinion dynamics. *Journal of Difference Equations and Applications*, 17(6):859–876, 2010.

[52] A.J. Laub, Leonard M. Silverman, and M. Verma. A note on cross–grammians for symmetric realizations. *Proceedings of the IEEE*, 71(7):904–905, 1983.

[53] Jerry W Lewis. Inversion of tridiagonal matrices. *Numerische Mathematik*, 38(3):333–345, 1982.

[54] Shihua Li, Haibo Du, and Xiangze Lin. Finite-time consensus algorithm for multi-agent systems with double-integrator dynamics. *Automatica*, 47(8):1706–1712, 2011.

[55] Zhongkui Li, Zhisheng Duan, Guanrong Chen, and Lin Huang. Consensus of multi-agent systems and synchronization of complex networks: a unified viewpoint. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 57(1):213–224, 2010.

[56] Xiangheng Liu, Andra Goldsmith, Sunider Sonia Mahal, and J Karl Hedrick. Effects of communication delay on string stability in vehicle platoons. *IEEE Intelligent Transportation Systems. Proceedings*, pages 625–630, 2001.

[57] Yang-Yu Liu, Jean-Jacques Slotine, and Albert-László Barabási. Controllability of complex networks. *Nature*, 473(7346):167–173, 2011.

[58] Yang-Yu Liu, Jean-Jacques Slotine, and Albert-László Barabási. Observability of complex systems. *Proceedings of the National Academy of Sciences*, 110(7):2460–2465, 2013.

[59] J. Lorenz. A stabilization theorem for continuous opinion dynamics. *Physica A*, 355(1):217–223, 2005.

[60] J. Lorenz. Consensus strikes back in the Hegselmann-Krause model of continuous opinion dynamics under bounded confidence. *Journal of Artificial Societies and Social Simulation*, 9(1):8, 2006.

[61] Jan Lorenz. Continuous opinion dynamics under bounded confidence: A survey. *International Journal of Modern Physics C*, 18(12):1819–1838, 2007.

[62] James MacQueen et al. Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[63] B. Marx, D. Koenig, and D. Georges. Optimal sensor and actuator location for descriptor systems using generalized Gramians and balanced realizations. *American Control Conference (ACC)*, pages 2729–2734, Boston, MA, USA, 2004.

[64] Mohammad-Ali Massoumnia, George C Verghese, and Alan S Willsky. Failure detection and identification. *IEEE Transactions on Automatic Control*, 34(3):316–321, 1989.

[65] Anthony R Metke and Randy L Ekl. Security technology for smart grid networks. *IEEE Transactions on Smart Grid*, 1(1):99–107, 2010.

[66] Estefanía Etchevés Miciolino, Gabriele Oliva, and Roberto Setola. Distributed opinion dynamics with heterogeneous reputation. *International Journal of System of Systems Engineering*, 4(3):277–290, 2013.

[67] Richard H Middleton and Julio H Braslavsky. String instability in classes of linear time invariant formation control with limited communication range. *IEEE Transactions on Automatic Control*, 55(7):1519–1530, 2010.

[68] Anahita Mirtabatabaei and Francesco Bullo. Opinion dynamics in heterogeneous networks: convergence conjectures and theorems. *SIAM Journal on Control and Optimization*, 50(5):2763–2785, 2012.

[69] B. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26(1):17–32, 1981.

[70] HP Ng, SH Ong, KWC Foong, PS Goh, and WL Nowinski. Medical image segmentation using k-means clustering and improved watershed algorithm. *IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 61–65, 2006.

[71] 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.

[72] Reza Olfati-Saber, Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[73] Gabriele Oliva, Damiano La Manna, Adriano Fagiolini, and Roberto Setola. Distance-constrained data clustering by combined k-means algorithms and opinion dynamics filters. *22nd IEEE Mediterranean Conference on Control and Automation (MED)*, pages 612–619, 2014.

[74] Gabriele Oliva, Roberto Setola, and Christoforos N Hadjicostis. Distributed k-means algorithm. *arXiv preprint arXiv:1312.4176*, 2013.

[75] F. Pasqualetti, S. Zampieri, and F. Bullo. Controllability metrics, limitations and algorithms for complex networks. 1(1):40–52, 2014.

[76] Fabio Pasqualetti. *Secure control systems: A control-theoretic approach to cyber-physical security.* PhD thesis, University of California, Santa Barbara, 2012.

[77] Fabio Pasqualetti, Florian Dorfler, and Francesco Bullo. Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, 58(11):2715–2729, 2013.

[78] Charles P Pfleeger and Shari Lawrence Pfleeger. *Security in computing.* Prentice Hall Professional Technical Reference, 2002.

[79] Marieke Quant, Peter Borm, and Hans Reijnierse. Congestion network problems and related games. *European Journal of Operational Research*, 172(3):919–930, 2006.

[80] Wei Ren, Randal W Beard, and Ella M Atkins. A survey of consensus problems in multi-agent coordination. *IEEE American Control Conference (ACC)*, pages 1859–1864, 2005.

[81] Jerome H Saltzer and Kenneth T Pogran. A star-shaped ring network with high maintainability. *Computer Networks (1976)*, 4(5):239–244, 1980.

[82] Adel S Sedra and Kenneth Carless Smith. *Microelectronic circuits*, volume 1. Oxford university press, 1998.

[83] Pete Seiler, Aniruddha Pant, and Karl Hedrick. Disturbance propagation in vehicle strings. *IEEE Transactions on Automatic Control*, 49(10):1835–1842, 2004.

[84] Iman Shames, Themistoklis Charalambous, Christoforos N Hadjicostis, and Mikael Johansson. Distributed network size estimation and average degree estimation and control in networks isomorphic to directed graphs. *50th IEEE Annual Allerton Conference on Communication, Control, and Computing*, pages 1885–1892, 2012.

[85] Frantisek Slanina. Dynamical phase transitions in Hegselmann-Krause model of opinion dynamics and consensus. *The European Physical Journal B*, 79(1):99–106, 2011.

[86] Siddharth Sridhar, Adam Hahn, and Manimaran Govindarasu. Cyber-physical system security for the electric power grid. *Proceedings of the IEEE*, 100(1):210–224, 2012.

[87] Stefan Streif, Rolf Findeisen, and Eric Bullinger. Relating cross gramians and sensitivity analysis in systems biology. *International Symposium on Mathematical Theory of Networks and Systems*, 2006.

[88] T. H. Summers and J. Lygeros. Optimal sensor and actuator placement in complex dynamical networks, arXiv preprint arXiv:1306.2491, 2013.

[89] Shreyas Sundaram and Christoforos N Hadjicostis. Finite-time distributed consensus in graphs with time-invariant topologies. *IEEE American Control Conference (ACC)*, pages 711–716, 2007.

[90] Yang Tang, Feng Qian, Huijun Gao, and Jürgen Kurths. Synchronization in complex networks and its application-a survey of recent advances and challenges. *Annual Reviews in Control*, 38(2):184–198, 2014.

[91] Folkert M Tangerman, JJP Veerman, and Borko D Stošic. Asymmetric decentralized flocks. *IEEE Transactions on Automatic Control*, 57(11):2844–2853, 2012.

[92] Arild Thowsen. The Routh-Hurwitz method for stability determination of linear differential-difference systems. *International Journal of Control*, 33(5):991-995, 1981.

[93] V Tzoumas, MA Rahimian, GJ Pappas, and A Jadbabaie. Minimal actuator placement with bounds on control effort. *arXiv preprint arXiv:1409.3289*, 2014.

[94] Alfonso Valdes and Keith Skinner. Adaptive, model-based monitoring for cyber attack detection. *Recent Advances in Intrusion Detection*, pages 80-93. Springer, 2000.

[95] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. *ICML*, volume 1, pages 577–584, 2001.

[96] Xiao Fan Wang and Guanrong Chen. Synchronization in scale-free dynamical networks: robustness and fragility. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 49(1):54–62, 2002.

[97] Chai Wah Wu. *Synchronization in complex networks of nonlinear dynamical systems*, volume 76. World Scientific, 2007.

[98] Feng Xiao, Long Wang, and Aiping Wang. Consensus problems in discrete-time multi-agent systems with fixed topology. *Journal of Mathematical Analysis and Applications*, 322(2):587–598, 2006.

[99] Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004.

[100] Gang Yan, Georgios Tsekenis, Baruch Barzel, Jean-Jacques Slotine, Yang-Yu Liu, and Albert-Laszlo Barabasi. Spectrum of controlling and observing complex networks. *arXiv preprint arXiv:1503.01160*, 2015.

[101] Yuecheng Yang, Dimos V Dimarogonas, and Xiaoming Hu. Opinion consensus of modified Hegselmann–Krause models. *Automatica*, 50(2):622–627, 2014.

[102] Wenwu Yu, Guanrong Chen, and Ming Cao. Some necessary and sufficient conditions for second-order consensus in multi-agent dynamical systems. *Automatica*, 46(6):1089–1095, 2010.