UNIVERSITY OF CALIFORNIA
Santa Barbara

Bio-Inspired Synchronization of Pulse-Coupled Oscillators and its Application to Wireless Sensor Networks

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

Doctor of Philosophy

in

Electrical and Computer Engineering

by

Felipe Eduardo Núñez Retamal

Committee in Charge:

Professor Francis J. Doyle III, Chair

Professor Andrew R. Teel

Professor João P. Hespanha

Professor Jeff Moehlis

Dr. Yongqiang Wang, Project Scientist

December 2014

The dissertation of
Felipe Eduardo Núñez Retamal is approved:

Professor Andrew R. Teel

Professor João P. Hespanha

Professor Jeff Moehlis

Dr. Yongqiang Wang, Project Scientist

Professor Francis J. Doyle III, Chair

September 2014

Bio-Inspired Synchronization of Pulse-Coupled Oscillators and its Application to Wireless Sensor Networks

by

Felipe Eduardo Núñez Retamal

Por sustento y arte...

Acknowledgements

First and foremost, I have to thank the Government of Chile through the Comisión Nacional de Investigación, Ciencia, y Tecnología (CONICYT) and the U.S. Department of State through the Fulbright Commission for giving me the opportunity to pursue a doctoral degree in the U.S.

I would like to express my gratitude to my advisor Professor Frank Doyle III for taking me as his student and for his support throughout my staying at UCSB.

I am very grateful to my committee members Professors João Hespanha, Jeff Moehlis, and in particular to Professor Andy Teel for his outstanding lectures and help with several technical issues faced during the writing of this dissertation.

My lab colleagues also deserve a special mention for their company day after day. Among them, my long-term collaborator, and also member of the committee, Yongqiang Wang has a privileged place. I must thank him for his support, advice, and the multiple conversations about Chinese culture and history that showed me a whole new way to look at the existence. And if existentialism is a matter of discussion, I have to thank my great friend Martin for not letting me forget that the unexamined life is not worth living.

None of this would have been possible without the endless support from my family in Chile. And at the end of the day, a moment with my wife Maria and our daughter Mariana reduces everything to a smile.

Curriculum Vitæ

Felipe Eduardo Núñez Retamal

Education

2007–2008	M.Sc. Electrical Engineering. Pontificia Universidad Católica de Chile, Santiago, Chile.
2002–2007	B.Sc. Electrical Engineering. Pontificia Universidad Católica de Chile, Santiago, Chile.

Working Experience

2010–2014	Graduate Research Assistant, Institute for Collaborative Biotechnologies, University of California Santa Barbara, USA.
2008–2010	Scientist, Electrical Engineering Department, Pontificia Universidad Católica de Chile, Santiago, Chile.
07/2008–01/2009	Control engineer, DICTUC S.A.
2006–2008	Research Assistant, Electrical Engineering Department, Pontificia Universidad Católica de Chile, Santiago, Chile.
01/2007–02/2007	Internship, Centrales del Maule Endesa Chile.
2005–2007	Teaching Assistant, Electrical Engineering Department, Pontificia Universidad Católica de Chile, Santiago, Chile.

Journal Publications

1. Núñez, F., Wang, Y.Q., Teel, A.R., and Doyle III, F.J., Synchronization of pulse-coupled oscillators to a global pacemaker. Submitted to *Systems & Control Letters*.

2. Núñez, F., Wang, Y.Q., Grasing, D., Desai, S., Cakiades, G., and Doyle III, F.J., Pulse-coupled time synchronization for distributed acoustic event detection using mobile wireless sensor networks. Submitted to *IEEE Access*.

3. Núñez, F., Wang, Y.Q., and Doyle III, F.J., Synchronization of pulse-coupled oscillators on (strongly) connected graphs. Accepted in *IEEE Transactions on Automatic Control*.

4. Núñez, F., Wang, Y.Q., and Doyle III, F.J., Global synchronization of pulse-coupled oscillators interacting on cycle graphs. Submitted to *Automatica*.

5. Wang, Y.Q., Núñez, F., and Doyle III, F.J., Statistical analysis of the pulse-coupled synchronization strategy for wireless sensor networks. *IEEE Transactions on Signal Processing*, 61(21), 5193-5204, 2013.

6. Wang, Y.Q., Núñez, F., and Doyle III, F.J., Increasing the synchronization rate of weakly connected pulse-coupled oscillators through design of phase response function: theory and application to wireless networks. *IEEE Transactions on Control Systems Technology*, 21(4), 1455-1462, 2013.

7. Wang, Y.Q., Núñez, F., and Doyle III, F.J., Energy-efficient synchronization strategy design for wireless sensor networks through reduced idle listening. *IEEE Transactions on Signal Processing*, 60(10), 5293-5306, 2012.

8. Carrasco, R., Núñez, F., and Cipriano, A., Fault detection and isolation in co-operative mobile robots using multi layer architecture and dynamic observers. *Robotica*, 29(4), 555-562, 2011.

9. Berrios, R., Núñez, F., and Cipriano, A., Fault tolerant measurement system based on Takagi-Sugeno fuzzy models for a gas turbine in a combined cycle power plant. *Fuzzy Sets & Systems*, 174(1), 114-130, 2011.

10. Grube, P., Núñez, F., and Cipriano, A., An event-driven simulator for multi-line Metro systems and its application to Santiago de Chile Metropolitan Rail Network. *Simulation, Modelling, Practice and Theory*, 19(1), 393-405, 2011.

11. Núñez, F., Reyes, F., Grube, P., and Cipriano, A., Simulating railway and metropolitan rail networks: from planning to on-line control. *IEEE Intelligent Transportation Systems Magazine*, 2(4), 18-30, 2010.

12. Núñez, F., Tapia, L., and Cipriano, A., Hierarchical hybrid fuzzy strategy for column flotation control. *Minerals Engineering*, 23(2), 117-124, 2010.

13. Lira, F., Muñoz, C., Núñez, F., and Cipriano, A., Short term forecasting of electricity prices in the Colombian market. *Generation, Transmission & Distribution, IET*, 3(11), 980-986, 2009.

14. Núñez, F., and Cipriano, A., Visual information model based predictor for froth speed control in flotation process. *Minerals Engineering*, 22(4), 366-371, 2009.

Referred Conference Publications

1. Wang, Y.Q., Núñez, F., and Doyle III, F.J., Mobility induced network evolution speeds up synchronization of wireless sensor networks. *Proceedings of the 2014 American Control Conference*, 3553-3558, Portland, USA.

2. Núñez, F., Wang, Y.Q., Desai, S., Cakiades, G. and Doyle III, F.J., Bio-inspired synchronization of wireless sensor networks for acoustic event detection systems. *Proceedings of the 2012 International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication ISPCS 2012*, 85-90, San Francisco, USA.

3. Núñez, F., Wang, Y.Q., and Doyle III, F.J., Bio-inspired hybrid control of pulse-coupled oscillators and application to synchronization of a wireless network. *Proceedings of the 2012 American Control Conference*, 2818-2823, Montreal, Canada.

4. Núñez, F., Wang, Y.Q., Teel, A.R., and Doyle III, F.J., Bio-inspired synchronization of non-identical pulse-coupled oscillators subject to a global cue and local interactions. *Proceedings of the* 4^{th} *IFAC Conference on Analysis and Design of Hybrid Systems 2012*, 115-120, Eindhoven, The Netherlands.

5. Núñez, F., and Cipriano, A., Model predictive control of multi-line metro systems: a distributed approach. *Proceedings of the* 9^{th} *IEEE International Conference on Control & Automation (IEEE ICCA'11)*, 532-537, Santiago, Chile.

6. Núñez, F., Silva, D., and Cipriano, A., Characterization and modeling of semi-autogenous mill performance under ore size distribution disturbances. *Proceedings of the* 18^{th} *IFAC World Congress 2011*, 9941-9946, Milano, Italy.

7. Núñez, F., Tejeda, G., Silva, D., and Cipriano, A., Global characterization of froth speed behavior in a rougher flotation line. *Preprints IFAC Workshop on Automation in Mining, Mineral and Metal Industry - IFACMMM2009, L. Bergh Ed.*, Viña del Mar, Chile.

8. Núñez, F., Navarro, S., Aguado, A., and Cipriano, A., State estimation based model predictive control for LHD vehicles. *Proceedings of the* 17^{th} *IFAC World Congress 2008*, 1448-1453, Seoul, Korea.

9. Núñez, F., and Cipriano, A., Hybrid modeling of froth flotation superficial appearance applying dynamic textures analysis. *Proceedings of the* 27^{th} *Chinese Control Conference 2008*, 117-121, Kunming, China.

10. Berrios, R., Núñez, F., Cipriano, A., and Paredes, R., Expert fault detection and diagnosis for the refrigeration process of a hydraulic power plant. *Proceedings of the* 27^{th} *Chinese Control Conference 2008*, 122-126, Kunming, China.

11. Núñez, F., and Cipriano, A., Superficial froth characterization in a flotation cell applying dynamic textures analysis and on-line texture classification. *Proceedings of the* 1^{st} *Mining Automation Congress: Automining 2008*, 163-171, Santiago, Chile.

Reviewing Activities

IEEE Transactions on Automatic Control; IEEE Transactions on Control of Network Systems; IEEE Transactions on Fuzzy Systems; IEEE Transactions on Vehicular Technology; IEEE Transactions on Computational Biology and Bioinformatics; Asian Journal of Control; Chemical Engineering Communications; Journal of Intelligent and Fuzzy Systems; American Control Conference; IEEE Conference on Decision and Control; IFAC World Congress; IFAC Symposium on Automation in the Mining, Mineral and Metal Industries; IFAC Workshop on Automation in the Mining, Mineral and Metal Industries.

Awards

2009	Fulbright scholarship to pursue the Ph.D. degree in the United States of America.
2002	Santiago scholarship for undergraduate studies in Chile.
2001	Best score in the math section of the Chilean national selection test.

Abstract

Bio-Inspired Synchronization of Pulse-Coupled Oscillators and its Application to Wireless Sensor Networks

by

Felipe Eduardo Núñez Retamal

Precise synchronization among networked agents is responsible for phenomena as diverse as coral spawning and consistency in stock market transactions. The importance of synchronization in biological and engineering systems has triggered an avalanche of studies analyzing the emergence of a synchronized behavior within a network of, possibly heterogeneous, agents. In particular, synchronization of networks of coupled oscillators has received great attention since limit cycle oscillators are a natural abstraction for systems where periodicity is a distinctive property. Examples of such systems include circadian rhythms and alternate-current power generators. This work deals with synchronization of pulse-coupled limit cycle oscillators (PCOs). A reverse engineering approach is taken with the objective of obtaining an abstraction for PCO networks able to capture the key properties observed in the classical biological PCO model, to finally implement it in an engineering system. To this end, we first reformulate the PCO model as a hybrid system, able to integrate in a smooth manner the continuous-time dynamics of

the individual oscillators and the impulsive effect of the coupling. Using our new model, we analyze the existence and stability of synchronization in a variety of PCO network topologies, starting from the simplest all-to-all network where global synchronization is proven to exist, to end giving synchronization conditions in the general strongly connected network case. Inspired by the strong synchronization properties of PCO networks we design a PCO-inspired time synchronization protocol for wireless sensor networks that enjoys all the advantages of our optimized PCO setup. A pilot implementation is presented going from a simulation stage to a hardware implementation in Gumstix development boards and industrial acoustic sensors. To test the potential of the protocol in a real application, we implement the PCO-based time synchronization protocol in a distributed acoustic event detection system, where a sensor network combines local measurements over an infrastructure-free wireless network to find the source of an acoustic event. An evaluation by simulation is given to illustrate the advantages of using the pulse-coupled synchronization strategy.

The contributions of this thesis range from the theoretical synchronization conditions for a variety of PCO networks to the design and implementation of a synchronization strategy for wireless sensor networks that seems to be the natural choice when using an infrastructure-free wireless network due to its simple formulation and natural scalability.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Synchronization of networks of agents is receiving increased attention due to its broad applications in biological systems [40, 27, 90, 5], mobile autonomous agents [49, 21, 86, 6], distributed computing [75, 64], and communication networks [52, 104, 24, 76]. In particular, synchronization of limit cycle oscillators is a pervasive topic in various disciplines where periodicity is a distinctive property. A limit cycle oscillator is a natural abstraction for periodicity-driven systems so diverse as alternate-current power generators [25], firing neurons [41, 34, 48, 63] , and computer clocks [43, 76]. Therefore, establishing conditions for the appearance of a synchronized state in a network of limit cycle oscillators has broad implications. In the particular setting we will study in this dissertation, an oscillator network is characterized by the number of, possibly heterogeneous, oscillators N, the struc-

ture of an underlying communication network that is modeled as a graph $\mathcal{R}$, and the coupling strategy by which oscillators influence their neighbors, which in this dissertation is assumed to be impulsive.

In an oscillator network, the role of each agent, i.e., leader or slave, can also determine the resulting dynamics. Therefore, analyzing the influence of a possible leader on the resulting synchronized state is of great importance. In fact, in the achievement of synchronization the interplay between a global cue and local interactions between agents is an important feature [99]. For example, in the mammalian olfactory bulb, ensembles of neurons synchronize to discriminate odors by utilizing intercellular interplays among individual neurons while at the same time receiving a global driving odorant stimulus via the odorant receptors [91]. In engineering, the coordination of a network of unmanned ground vehicles is achieved by means of the interplay between individual vehicles and external coordination from the central resources [86], while in clock synchronization of wireless networks a standard approach is to synchronize different time references in a real-time network through internal interactions between different nodes and external coordination from the central server [52, 31]. In this dissertation, we will analyze two different scenarios: purely decentralized synchronization (leaderless) and synchronization to a unique leader agent.

In the following, we will denote a non-linear limit cycle oscillator, as the dynamical system given by:

$$\dot{z} = H(z, t, u) \tag{1.1}$$

where H is a non-linear mapping that satisfies the classical smoothness properties required for existence and uniqueness of solutions from an initial condition z_0, and u is a measurable function describing the input applied to the oscillator. We write as $\vartheta(t, z_0, u)$ the solution of (1.1) from the initial condition z_0 and input u. The characteristic property of a limit cycle oscillator is that its zero-input steady-state behavior is periodic. Hence, we assume that the system $\dot{z} = H(z, t, 0)$ admits a stable periodic orbit ζ, sufficiently attractive, with period $T = \frac{2\pi}{w}$, with w being the natural frequency of the oscillator. To analyze limit cycle oscillator networks, a usual approach is to map the dynamics of the multidimensional limit cycle oscillator into the one dimensional phase space $[0, 2\pi]$ (where 0 and 2π are mapped one into the other) via phase reduction techniques [47, 14]. After the phase reduction stage, the dynamics of the network are entirely described by the relationship between the oscillators' phases; such interconnected systems are known as phase-coupled oscillators. This dissertation focuses on synchronization of networks of interconnected oscillators described in the phase space, i.e., on phase-coupled oscillators.

Regarding the effect of the input function u, this has to be addressed carefully. In general, the attractiveness of the periodic orbit ζ will limit the strength of the

input u since a strong perturbation can cause the oscillator to damp. In this dissertation, we will assume that the periodic orbit is sufficiently attractive, i.e., the basin of attraction is sufficiently big, such that the oscillator (1.1) converges to the periodic orbit after an impulsive perturbation is applied. In the corresponding phase space, the effect of the input function u is characterized by a phase response curve (PRC) [18, 22, 105, 106, 88], which tabulates the phase shift of an oscillator resulting from the application of a stimulus. In the remainder of this dissertation, we will focus on limit cycle oscillators coupled in an impulsive rather than smooth manner, i.e., the input function u is a train of impulses. From a terminology point of view, the impulsive coupling has two connotations: 1) oscillators that interact only at discrete time instants are known as firing oscillators; and 2) an ensemble of firing oscillators is known as a pulse-coupled network, or as pulse-coupled oscillators.

Pulse-coupled oscillators (PCOs) are limit cycle oscillators that are coupled together to form a network by exchanging pulses at discrete time instants. A pulse has two effects on the network state: 1) it resets the phase at the originating oscillator, and 2) it induces a jump on the phase of the receiving oscillators. The magnitude of the impulsive jump induced is, in general, phase dependent and is given in the form of a phase response curve (PRC) Q. Moreover, a usual approach is to include a coupling strength l to scale the effect of the PRC. In this setting, the value of l can be interpreted as the extra energy needed to synchronize the

system, as is indeed the case when PCOs are realized using passive circuits, or as an extra gain present at the receiver side.

The dynamics of a network of PCOs, and thus its synchronization properties, are fully determined by the interaction topology (communication network) $\mathcal{R}$, the number of oscillators in the network N, the initial phases x_0, and the feedback strategy given by Q and l, i.e., the PRC and the coupling strength. Despite the simple formulation and behavior of an isolated firing oscillator, networks of PCOs are able to exhibit intricate collective dynamics. For this reason, PCOs have emerged as a powerful modeling and design tool in complex networked biological and engineering systems. Examples of biological systems that have been modeled using PCOs include cardiac pacemakers [78], crickets that chirp in unison [98], and rhythmic flashing of fireflies [15, 16]. While one of the most important applications of PCOs in engineering is time synchronization in sensor networks [76, 42, 43, 45, 51, 7].

Synchronization of PCOs has been analyzed since the first appearance of the integrate and fire oscillator model by Peskin [78]. In his work, Peskin made the following conjectures: 1) for arbitrary initial conditions, the system approaches a state in which all of the oscillators fire synchronously, and 2) this remains true, even when the oscillators are not quite identical. Numerous studies addressing these conjectures have been conducted, with variable success. In one of the most

remarkable studies [62], using a slightly different PRC than the one used in [78], the authors proved that synchronization of identical PCOs in an all-to-all setting is possible from every initial condition except from a set of zero Lebesgue measure, yet the non-identical and general coupling topology cases are not examined. Under the assumption of weak coupling, several authors have continued studying networks of PCOs using the phase model in [47] for different communication topologies and coupling functions. By exploiting weak coupling, the impulsive updates can be replaced by their average, hence resorting to an equivalent ordinary differential equation to do the analysis. However, the weak coupling assumption needed to apply the techniques in [47] reduces the applicability, making it harder to claim a general result. Synchronization in networks of PCOs has proven difficult to establish and it is still not clear whether it is feasible or not to achieve synchronization in networks more general than all-to-all coupled. Recently, [57] showed that all-to-all connected PCOs exhibit a dichotomic behavior, i.e., the network can either synchronize, or the oscillators form clusters distributed in the unit circle, depending on the characteristics of the PRC. Considering the PRC a variable element rather than given is an appealing strategy for synthetic PCOs, and the work in [57] illustrates how important the PRC is for the existence and stability of a synchronized state. A further step in this direction is the one taken in [102, 100] where the PRC is considered to be a design parameter and shaped accordingly

to improve the synchronization rate of the network, giving as a result an optimal PRC in the sense of synchronization rate that should be used to gain understanding of networks with arbitrary coupling structure. The lack of results regarding synchronization of arbitrary coupled PCOs is mainly due to the difficulties arising in the analysis when the coupling is not all-to-all. Typical methods for conducting the analysis rely on an invariant firing sequence, which is not present when the network is not all-to-all coupled. However, arbitrary coupled networks are of great importance and thus a strong theoretical understanding is needed. For example, interconnected oscillators interacting on cycles, or rings, have been used to model a variety of physiological phenomena such as segmental undulations in the leech, and hexapodal gait generation in insects [26]. In engineering, ring buses are pervasive in industrial computer networks. Furthermore, in an engineering system such as a large-scale sensor network, all-to-all coupling is extremely unlikely to exist. This gap motivates the study of synchronization conditions for networks of PCOs with a general coupling structure. However, not only synchronization conditions are important; also the attractiveness of the synchronized state is a preponderant matter, especially in engineering applications of PCOs where global synchronization of general networks is a desirable property. Nonetheless, a precise characterization of the basin of attraction of the synchronization manifold for networks of PCOs is not available.

In this dissertation, we aim to fill the gap regarding synchronization properties of arbitrary coupled PCOs and the corresponding structure of the basin of attraction. Of particular interest is to find conditions for global synchronization. The previous work on PCOs relied on the direct use of the biological model (after a phase reduction stage), which leads to a fixed feedback strategy since the PRC is a property of the firing oscillator. In this dissertation we propose to re-design the PCO model to combine successful synchronization strategies taken from biology with modern control techniques to improve performance. Specifically, in this dissertation PCOs are first modeled as a hybrid dynamical system, as suggested in [58], to handle the impulsive behavior naturally. We then analyze the resulting model to establish synchronization conditions for a variety of networks based on both, the PRC Q, and the coupling strength l. Our primary focus is to study the synchronization properties of networks coupled using the rate optimal PRC in the sense of [102, 100]. To show the implications of our theoretical findings in a real-world setting, we design a time synchronization protocol based on and supported by the PCO paradigm and apply it to a practical synchronization problem in wireless sensor networks.

1.2 Objectives, Contributions, and Organization

The main objective of this work is to gain a deep understanding of the principles leading to the appearance of a stable synchronized state in a network of biological PCOs, and to combine these principles into a rigorous abstraction to be applied in engineering applications. We focus on the application of pulse-coupled synchronization principles to the problem of time synchronization in infrastructure-free (ad-hoc) networks, particularly wireless sensor networks (WSNs). We refer to this process as reverse engineering biological pulse-coupled synchronization into time synchronization in WSNs. This dissertation presents the reverse engineering process in four main chapters, which are briefly outlined in the following.

In Chapter 2 we review the classical formulation of the biological model of PCOs starting with the integrate and fire oscillator model by Peskin [78] and its formulation as a network. Early synchronization results derived from the biological model of PCOs are reviewed, of particular relevance is the work [62], which can be regarded as the first rigorous formulation of a network of PCOs and its synchronization properties. We use the abstraction in [62] as starting point for our developments. The first contribution of this work is presented at the end of Chapter 2: a new hybrid model of PCOs. Motivated by the fact that the existing analysis relies heavily on the use of weak-coupling approximations [47] or the assumption that the firing order is invariant [62, 57], we opted to reformulate the PCO model

in a framework that allows analyzing the PCO network in a general case. Our new model integrates in an harmonious way the continuous-time nature of a limit cycle oscillator and the impulsive nature of the coupling by using a hybrid dynamical systems formulation. Moreover, the appearance of advanced tools for the analysis of hybrid systems allows us to draw strong conclusions regarding synchronization without the need of using weak-coupling approximations or restricting the network to maintain a fixed firing order, which, in turn, allows us to study synchronization in networks with general coupling topology and coupling strength.

Chapter 3 deals with synchronization of PCOs under different coupling (or communication) topologies and system structures, i.e., with and without a leader. We start by analyzing the case where a network of PCOs synchronizes to an omnipresent leader, i.e., an agent that can reach every other agent in the network and does not react to any incoming pulse. Synchronization conditions are given for the identical natural frequencies case and conditions for synchronization in frequency (a weaker synchronization notion) are given for the non-identical natural frequencies case. For the leaderless case, the main contributions of this chapter are conditions for global synchronization in all-to-all, strongly rooted, and connected bi-directional networks. We finish the leaderless section studying in detail the cycle network case, for which we give the exact value of the critical coupling strength, as a function of the number of oscillators in the cycle, that leads to global synchro-

nization. Numerical examples for all the theoretical results presented are given at the end of the Chapter to support our analytical findings.

In Chapter 4 a PCO-based synchronization protocol for time synchronization in wireless sensor networks is presented. We start the chapter by reviewing existing packet-based synchronization protocols, as well as pulse-based synchronization protocols. The rest of the Chapter focuses on the translation of the PCOs abstraction into a protocol suitable for operation in a wireless network, which is the main contribution of the Chapter. First, we present the operating philosophy of the algorithm, which is based on the CSMA and IEEE 802.11 protocols, combining carrier sense for collision avoidance and control packets for coupling information. Next, we present the event-driven algorithmic formulation of the protocol, its main configuration parameters, and operation modes, i.e., its data sending capabilities, synchronization mode, and sleep mode to save energy by reducing idle listening. The synchronization properties of the protocol and the implementation strategy as a MAC layer protocol are given next. To test the protocol in a realistic environment, we use simulations in Qualnet$^{\text{TM}}$ [85], a network simulation tool that can be used to simulate wireless and wired communication networks. The two final sections of the chapter present pilot implementations of the pulse-coupled synchronization protocol in Gumstix development boards and a network of acoustic sensors provided by BioMimetic Systems, Inc.

In Chapter 5, the application of our pulse-coupled time synchronization algorithm to a distributed acoustic event detection system is presented. Acoustic event detection systems are highly sensitive to synchronization errors, hence, a precise time synchronization protocol is needed to ensure correct localization. We start the chapter by describing the sensor network-based distributed acoustic event detection system. A rigorous formulation of the centralized and distributed localization problem is presented. We contribute a novel strategy to solve the distributed localization problem using a protocol based on double linear iterations and linear consensus algorithms. A performance example is then given to show the necessity of having a precise time synchronization protocol for both, centralized and distributed localization. The main contribution of this chapter is the application of our time synchronization protocol to the distributed localization of the acoustic source. We present a strategy that integrates the distributed localization protocol and the PCO-based time synchronization in an harmonious manner over an infrastructure-free wireless network. We finish the chapter by presenting an evaluation by simulation of the distributed acoustic event detection system operating with pulse-coupled time synchronization.

The dissertation concludes in Chapter 6 where general conclusions and future directions of research are given.

The main findings of this dissertation resulted in a series of publications currently available or under evaluation. Results from Chapter 3 can be found in [69], [70], [71], and [74]. Results from Chapter 4 can be found in [101], [102], and [103]. Results from Chapter 5 can be found in [68] and [73].

1.3 Preliminaries

1.3.1 Basic Notation and Definitions

In this work, $\mathbb{R}$ denotes the real numbers, $\mathbb{R}_{\geq 0}$ denotes the set of nonnegative real numbers, $\mathbb{Z}_{\geq 0}$ denotes the set of nonnegative integers, $\mathbb{R}^n$ denotes the Euclidean space of dimension n, and $\mathbb{R}^{n\times n}$ denotes the set of $n \times n$ square matrices with real coefficients. I_n denotes the n by n identity matrix. For a vector $v \in \mathbb{R}^n$, $\text{diag}(v) \in \mathbb{R}^{n\times n}$ denotes the diagonal matrix with the elements of v in the diagonal. We denote as $\mathbf{1}$ the column vector of all ones of appropriate dimension. Throughout the dissertation, $\epsilon \in \mathbb{R}$ denotes a real number sufficiently small. For a real number a, $[a]$ denotes its integer part. For a countable set χ, we denote its cardinality as $|\chi|$; for two sets Λ_1 and Λ_2, we denote their difference as $\Lambda_1 \setminus \Lambda_2$.A set-valued mapping $\Phi : A \rightrightarrows B$ associates to the element $\alpha \in A$ the set $\Phi(\alpha) \subseteq B$; the graph of a set-valued mapping is the set: $\text{graph}(\Phi) := \{(\alpha, \beta) \in A \times B : \beta \in \Phi(\alpha)\}$. A

set-valued mapping Φ is outer semi-continuous if and only if its graph is a closed set [79, Theorem 5.7(a)]. The acronym w.l.o.g. stands for without loss of generality.

1.3.2 Hybrid Systems Preliminaries

In this work we follow the framework given in [36, 37]. A hybrid system $\tilde{\mathcal{H}}$ consists of continuous-time dynamics (flows), discrete-time dynamics (jumps), and sets on which these dynamics apply:

$$\tilde{\mathcal{H}} : \begin{cases} \dot{x} \in F(x), & x \in \mathcal{C} \\ x^+ \in G(x), & x \in \mathcal{D} \end{cases} \tag{1.2}$$

where the flow map F and the jump map G are set-valued mappings, $\mathcal{C} \subseteq \mathbb{R}^n$ is the flow set, and $\mathcal{D} \subseteq \mathbb{R}^n$ is the jump set, $(F, \mathcal{C}, G, \mathcal{D})$ is the data of $\tilde{\mathcal{H}}$. A subset $E \subset \mathbb{R}_{\geq 0} \times \mathbb{Z}_{\geq 0}$ is a hybrid time domain if it is the union of infinitely many intervals of the form $[t_j, t_{j+1}] \times j$, or of finitely many such intervals, with the last one possibly of the form $[t_j, t_{j+1}] \times j$, $[t_j, t_{j+1}) \times j$, or $[t_j, \infty) \times j$. A solution to $\mathcal{H}$ is a function $\phi : \operatorname{dom} \phi \to \mathbb{R}^n$ where $\operatorname{dom} \phi$ is a hybrid time domain and for each fixed j, $t \mapsto \phi(t, j)$ is a locally absolutely continuous function on the interval $I_j = \{t : (t, j) \in \operatorname{dom} \phi\}$. ϕ is called a hybrid arc, and is such that: for each $j \in \mathbb{N}$ for which I_j has nonempty interior $\dot{\phi}(t, j) \in F(\phi(t, j))$ for almost all

$t \in I_j$, $\phi(t,j) \in \mathcal{C}$ for all $t \in [\min I_j, \sup I_j)$; for each $(t,j) \in \operatorname{dom}\phi$ for which $(t,j+1) \in \operatorname{dom}\phi$, $\phi(t,j+1) \in G(\phi(t,j))$, $\phi(t,j) \in \mathcal{D}$. A solution ϕ is nontrivial if its domain contains at least one point different from $(0,0)$, is maximal if it can not be extended, and is complete if its domain is unbounded.

1.3.3 Graph Theory

Throughout this dissertation we use several concepts from algebraic graph theory. An extensive treatment of the topic can be found in [35, 61]. Consider a network with $N \in \mathbb{Z}_{\geq 0}$ agents. The communication between agents is modeled by a weighted directed graph $\mathcal{R} = \{\mathcal{V}, \mathcal{E}_\mathcal{R}, \mathcal{A}_\mathcal{R}\}$, where $\mathcal{V} = \{1, \ldots, N\}$ is the node set of the graph. $\mathcal{E}_\mathcal{R} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set of the graph, whose elements are such that $(i,k) \in \mathcal{E}_\mathcal{R}$ if and only if node k receives the pulse emitted by node i, we assume that the self edge $(i,i) \notin \mathcal{E}_\mathcal{R}$ unless it is explicitly required. $\mathcal{A}_\mathcal{R} = [a_{ik}] \in \mathbb{R}^{N \times N}$ is the weighted adjacency matrix of $\mathcal{R}$ with $a_{ik} \in \{0, l\}$, where $a_{ik} = l$ if and only if $(i,k) \in \mathcal{E}_\mathcal{R}$. We use the notation $\bar{\mathcal{R}} = \{\mathcal{V}, \bar{\mathcal{E}}_\mathcal{R}, \bar{\mathcal{A}}_\mathcal{R}\}$ when we explicitly restrict the graph to be bidirectional, or undirected, i.e., $(i,k) \in \bar{\mathcal{E}}_\mathcal{R}$ if and only if $(k,i) \in \bar{\mathcal{E}}_\mathcal{R}$. A directed graph $\mathcal{R}$ is said to be strongly connected if there is a directed path between any pair of nodes; the equivalent concept for undirected graphs is the graph being connected. For node i, $\mathcal{N}^{i+} = \{k \in \mathcal{V} : (k,i) \in \mathcal{E}_\mathcal{G}\}$ denotes its out-neighbors, i.e., the set of nodes that receive the pulse emitted by

i. Similarly, $\mathcal{N}^{i-} = \{k \in \mathcal{V} : (i,k) \in \mathcal{E}_\mathcal{G}\}$ denotes the in-neighbors of node i, i.e., the set of nodes whose pulses are received by i. If the graph is restricted to be bidirectional, we denote the set of neighbors of node i as $\mathcal{N}^i$ (note that in this case $\mathcal{N}^{i+} = \mathcal{N}^{i-}$). For a network, we define its indegree as $\min_{i\in\mathcal{V}} |\mathcal{N}^{i-}|$. A graph $\mathcal{R}$ is strongly rooted [19] if a node can reach all other nodes, i.e., there exists $i \in \mathcal{V}$ such that $|\mathcal{N}^{i+}| = N - 1$; such node i is called the root of $\mathcal{R}$. For a given square matrix $A_* \in \mathbb{R}^{N\times N}$, we will refer to its underlying graph as the graph formed by N nodes with an edge from i to k if and only if the ikth entry of A_* is nonzero.

Chapter 2

Pulse-Coupled Oscillators

As an introduction to the results developed in the rest of this dissertation, this Chapter presents an overview of mathematical models of firing oscillators and their interconnection to form a network of pulse-coupled oscillators.

2.1 Historical Perspective

Firing oscillators were introduced as early as the work by Lapicque in 1907 [1] where they were used as a one dimensional abstraction for a firing neuron. The success of the firing oscillator model to emulate firing neurons has placed it as the top choice to study the collective behavior of neural networks, although the simple RC circuit used by Lapicque has evolved into higher order state-space models as the one by Hodgkin and Huxley [41], which is able to describe the neuron's dynamics more accurately than the original RC circuit. However, a multi-dimensional model presents several drawbacks to study the collective behavior in a large network of

firing oscillators, posing the necessity of obtaining a low dimensional abstraction able to mimic the high order dynamics while being mathematically tractable when a large number of oscillators are coupled. The integrate and fire oscillator appears as a one-dimensional reduction of multidimensional firing dynamics that focuses only in the "slow" part of the cycle and models the "fast" part as an impulsive reset.

In his study about mathematical aspects of heart physiology, Peskin [78] used the integrate and fire model to construct an abstraction for the cardiac pacemaker. Starting from a version of the Hodgkin and Huxley neuron model specially adapted to the heart by Noble and Tsien [67], Peskin developed a reduced one dimensional model able to capture the firing dynamics of the cardiac pacemaker. Peskin's integrate and fire model is given by:

$$\dot{z} = -\gamma z + S_0 + u(t) > 0 \tag{2.1}$$

where $z \in [0, 1]$ is the state variable, $\gamma > 0$ is the inhibition coefficient, and $S_0 > \gamma$ is the internal driving signal of the oscillator. When the state reaches the upper limit $z = 1$, the oscillator fires and resets its state to $z = 0$. The model (2.1) is known as the "leaky integrate and fire model", which is widely used in neuroscience to study ensemble of firing neurons since it is a direct simplification of the Hodgkin and Huxley model.

While complex high order models (e.g. the Hodgkin and Huxley model) focus on the entire cycle of the oscillator, including the firings, the simplified dynamics of the integrate and fire model focuses only on the sub-cycle between the firings, and the firings are considered purely as impulsive resets. The approximation is relevant yet not restrictive, especially when considering a network of coupled firing oscillators. In this case, the behavior of the whole network does not depend on the fast firings but on the evolution between two firings. Nonetheless, the use of an integrate and fire model reduces the cost of numerical simulations and dramatically simplifies analytical studies.

An important contribution of Peskin's work is the analysis of a network of leaky integrate and fire oscillators and the formulation of the synchronization question. In the networked system proposed by Peskin, each oscillator follows the dynamics given by (2.1) with $u(t) = 0$. Every time an oscillator reaches the upper limit $z = 1$, it resets its state to $z = 0$ and induces an increment of ϵ/N in the state of every other oscillator [78]. It is important to note that Peskin's model considers all-to-all communication and requires a weak coupling. Regarding feasibility of synchronization, Peskin made the following conjectures 1) for arbitrary initial conditions, the system approaches a state in which all of the oscillators fire synchronously, and 2) this remains true, even when the oscillators are not quite identical. Peskin himself addressed the first conjecture for the case of 2 oscillators and left the question

open for the other cases, starting an avalanche of works focusing on synchronization properties of pulse-coupled oscillators.

2.2 Existing Models and Synchronization Results

Peskin's model of the leaky integrate and fire oscillator (2.1) can be regarded as an special case of the more general class of monotone integrate and fire oscillators. Mirollo and Strogatz [62] formulated the general monotone integrate and fire oscillator model as a one dimensional system with a state variable $z \in [0, 1]$ that when coupled gives rise to the classical formulation of a network of PCOs.

The network is formed by N integrate and fire oscillators, where each oscillator $i \in \{1, 2, \ldots, N\}$ follows

$$z_i = f(x_i) \tag{2.2}$$

where $f : [0, 1] \to [0, 1]$ is smooth, monotonic increasing, and concave down, i.e., $f'(x_i) > 0$, $f''(x_i) < 0$; and $x_i \in [0, 1]$ is a phase-like variable such that:

$$\frac{\partial x_i}{\partial t} = \frac{1}{T} = w \tag{2.3}$$

and $x_i = 1$ ($x_i = 0$) when the oscillator is at the end (start) of the cycle, i.e., when $z_i = 1$ ($z_i = 0$). Therefore, $f(0) = 0$ and $f(1) = 1$ holds. The oscillators are

assumed to interact by a simple form of pulse coupling: when an oscillator fires it increases the state of all the other oscillators by an amount ϵ, or force them to fire, whichever is less. That is,

$$z_i(t) = 1 \Rightarrow z_i(t^+) = 0, \quad z_k(t^+) = \min(1, z_k(t) + \epsilon), \quad \forall k \neq i \tag{2.4}$$

In [62], Mirollo and Strogatz proved that a network of all-to-all connected monotone integrate and fire oscillators synchronizes from every initial condition except from a set of zero measure. And, as a corollary, that Peskin's first conjecture holds for all N and for all ϵ, $\gamma > 0$.

Exploiting the fact that there is a trivial change of variables from the one-dimensional state z of the integrate and fire model to the phase variable x, several studies have been conducted focusing on the phase space formulation of the integrate and fire model:

$$\dot{x_i} = w_i + Q(x_i)u(t) \tag{2.5}$$

where w_i corresponds to the natural frequency of the oscillator, and Q is the PRC [14].

Remark 2.1 *In the existing PCOs literature, most of the studies use a phase variable defined between 0 and 1. For consistency with the traditional concept of*

phase in an oscillating system, throughout this dissertation the phase x is defined in the interval $[0, 2\pi]$ with the firing occurring whenever $x = 2\pi$.

Remark 2.2 *Note that for an integrate and fire oscillator, if the function f is constant and equal to w, then the phase and state dynamics match. Although not considered in the formulation by Mirollo and Strogatz, linear integrate and fire oscillators have caught the attention lately due to its simple implementation in a digital processor [3]. Figure 2.1 shows the state dynamics of a linear integrate and fire oscillator as well as the effect of the impulsive coupling of the form (2.4).*

The explicit appearance of the PRC in (2.5), and the fact that any integrate and fire model can be transformed to the form (2.5) by a change of variables and, moreover, any higher order state-space model can be reduced to the form (2.5) via phase reductions methods [44, 47, 106, 14], gives researchers the opportunity to analyze networks of phase oscillators of the form (2.5) by focusing on the properties of the PRC Q.

The work by Mauroy [57] looks to elucidate the influence of the PRC on the synchronization properties of a network of PCOs. In [57], the author studies a class of integrate and fire oscillators more general than the purely concave-down monotonic oscillators analyzed in [62], including concave-up monotonic oscillators, quadratic integrate and fire, and quadratic-like integrate and fire oscillators. The most important finding in [57] is that networks of all-to-all coupled PCOs present

a dichotomic behavior, i.e., they synchronize or converge to a phase-locked state, depending on the characteristics of the vector field f, i.e., depending on the characteristics of the PRC.

The importance of the PRC as a synchronization enabler, motivates to consider the PRC a design element and shape it accordingly to achieve a desired behavior in a network of PCOs (e.g., global synchronization, fast convergence, phase-locked-configuration, etc).

Remark 2.3 *Note that the PRC Q is an intrinsic characteristic of the multidimensional oscillator. By considering the PRC a design variable, the biological links are broken and the system becomes purely a mathematical object.*

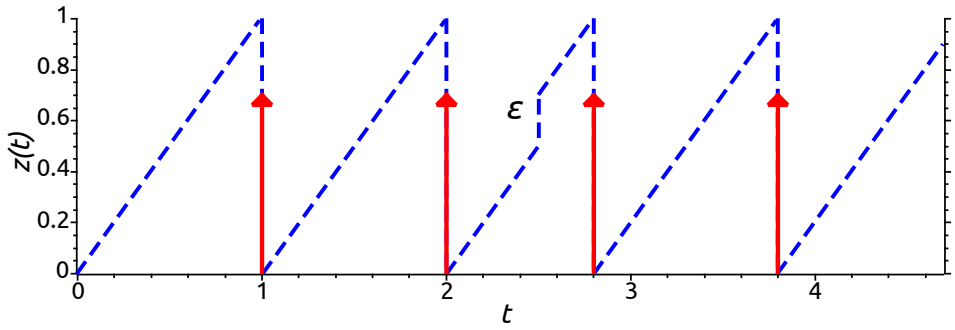


Figure 2.1: Linear integrate and fire oscillator and effect of the impulsive coupling.

2.3 A New Hybrid Model of PCOs

In the following, we will develop a phase space model for networks of PCOs. The aim is to obtain a model able to incorporate the PRC as a general mapping, and to handle the, possibly strong, impulsive coupling in a natural way. Moreover, we introduce the underlying communication graph as part of the formulation since we will focus on networks with general coupling structures.

The network of PCOs consists of N oscillators interacting on a given graph $\mathcal{R} = \{\mathcal{V}, \mathcal{E}_{\mathcal{R}}, \mathcal{A}_{\mathcal{R}}\}$. The phase of each oscillator evolves continuously following its natural frequency, and jumps impulsively upon receiving a pulse. Pulses are generated following an integrate-and-fire process, i.e., when its phase reaches the limit (2π in this case), the oscillator fires, i.e., emits a pulse, and resets its phase to 0. When an oscillator receives a pulse, it updates its phase according to the coupling strength $l \in (0, 1]$ and the PRC, which is formally defined in the framework of hybrid systems as follows:

Definition 2.1 (Phase Response Curve) *A phase response curve (PRC), or phase resetting curve [18, 22, 105, 106, 88], describes the change in the phase of an oscillator due to a pulse stimulus, as a function of the phase at which the pulse is received. A phase response curve* $Q : [0, 2\pi] \rightrightarrows \mathcal{Q} \subseteq \mathbb{R}_{\geq 0}$ *is called an advance-only PRC. Similarly, a phase response curve* $Q : [0, 2\pi] \rightrightarrows \mathcal{Q} \subseteq \mathbb{R}_{<0}$ *is*

called a delay-only PRC. A phase response curve $Q : [0, 2\pi] \rightrightarrows \mathcal{Q} \subseteq \mathbb{R}$ *is called an advance-delay PRC if there exists* $\bar{q}_1 \in Q(q_1)$ *and* $\bar{q}_2 \in Q(q_2)$ *satisfying* $\bar{q}_1 > 0$ *and* $\bar{q}_2 < 0$.

Remark 2.4 *In the mathematical neuroscience literature, advance-only PRCs are referrer to as Type I PRCs. Similarly, advance-delays PRCs are referred to as Type II PRCs. To deal with a delay-only PRC the system is modeled as an inhibitory system, i.e., a system where the coupling strength is negative, coupled using a Type I PRC [44, 14].*

2.3.1 Data

In this dissertation we consider a constant and identical coupling strength l, and then the weighted adjacency matrix $\mathcal{A}_{\mathcal{R}}$ is such that $a_{ij} \in \{0, l\}$. The network of N oscillators is modeled as the hybrid system $\mathcal{H}$ with state x given by:

$$x := [x_1, \ldots, x_N]^T \tag{2.6}$$

where $x_i \in [0, 2\pi]$ denotes the phase of the ith oscillator. The data of $\mathcal{H}$ is given by:

If $x \in \mathcal{C} := \{x \in \mathbb{R}^N : x_i \in [0, 2\pi],\ \forall i \in \mathcal{V}\} := [0, 2\pi]^N$ then:

$$\dot{x}_i = w_i \tag{2.7}$$

similarly, if $x \in \mathcal{D}_i := \{x \in \mathcal{C} : x_i = 2\pi\}$ then:

$$\left.\begin{aligned} x_i^+ &= 0 \\ x_k^+ &\in \{\tilde{x}_k : \tilde{x}_k = \mathrm{sat}_0^{2\pi}(x_k + a_{ik}q),\ q \in Q(x_k)\} \end{aligned}\right\} =: G_i(x) \tag{2.8}$$

where $w_i \in \mathbb{R}_{>0}$ denotes the natural frequency, $a_{ik} \in \{0, l\}$ is the corresponding entry from $\mathcal{A}_{\mathcal{R}}$, $Q : [0, 2\pi] \rightrightarrows \mathbb{R}$ is the PRC, and $\mathrm{sat}_0^{2\pi}$ is the linear function with slope one that saturates at 2π from above and 0 from below. It should be noted that the $\in$ in (2.8) implies that the PRC might be a set-valued mapping. The jump map when $x \in \mathcal{D}_i$ can be rewritten as:

$$\left.\begin{aligned} x_i^+ &= 0 \\ x_k^+ &\in \mathrm{sat}_0^{2\pi}(x_k + a_{ik}Q(x_k)) \end{aligned}\right\} =: G_i(x), \quad x \in \mathcal{D}_i \tag{2.9}$$

Moreover, the effect of the saturation function can be eliminated by imposing a range condition on the PRC as $\mathrm{graph}(lQ) \subseteq \Omega := \{(x, y) : x \in [0, 2\pi], -x \leq y \leq 2\pi - x\}$ (see Figure 2.2). This condition is not restrictive since if part of the graph lies outside Ω, we can replace the PRC with a saturated version of it, without

affecting the resulting dynamics. To continue the analysis we utilize the following assumption.

Assumption 1 *The PRC Q is such that: $Q(0) = Q(2\pi) = \{0\}$. Moreover, Q is an outer semi-continuous set-valued mapping and bounded on $[0, 2\pi]$.*

Finally, the jump set is defined as the union over the node set of the individual jump sets previously defined:

$$\mathcal{D} := \bigcup_{i \in \mathcal{V}} \mathcal{D}_i \tag{2.10}$$

and, similarly, the jump map is defined as:

$$G(x) := \bigcup_{i \in \mathcal{V}: x \in \mathcal{D}_i} G_i(x) \tag{2.11}$$

Assumption 1 guarantees that the hybrid system $\mathcal{H}$ as defined above is well-posed [36, 37]. Moreover, the condition $Q(0) = Q(2\pi) = \{0\}$ restricts any undesired avalanche-type behavior. It should be noted that, although not treated in detail in this dissertation, the proposed well-posed hybrid system enjoys several robustness properties [36, 37] that make the stability (synchronization) results proven in the following sections still hold in a practical sense in the presence of small perturbations.

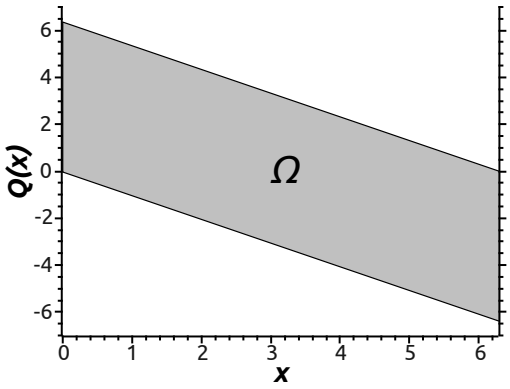


Figure 2.2: Allowable region for the graph of the PRCs.

Remark 2.5 *An important concept used in the analysis of PCOs is absorption [59, 62], which leads to synchronization in finite time. It should be noted that in our model (2.7)-(2.11) absorption is modeled by the saturation function and, if* $graph(Q) \subseteq \Omega$*, it can take place only when* $l = 1$ *and otherwise synchronization will be asymptotic.*

Remark 2.6 *Note that the use of the saturation at* 2π *from above is consistent with the use of the* min *function in the Mirollo and Strogatz's model (2.4). In the same line, the use of the saturation at* 0 *from below is a natural extension when an advance-delay PRC is used.*

Remark 2.7 *It should be noted that the Mirollo and Strogatz's model considers a constant* ϵ*, in contrast with the* $\frac{\epsilon}{N}$ *used by Peskin. Although* ϵ *is required to be small, for a large network the Mirollo and Strogatz's model is susceptible to firing*

avalanches since multiple incoming pulses add up [62]. Our model handles this undesired behavior by requiring $Q(0) = Q(2\pi) = \{0\}$.

2.3.2 Solutions to the Hybrid Model

The behavior of the solutions to the hybrid system $\mathcal{H}$ on a graph $\mathcal{R}$, is characterized as follows.

Proposition 2.1 *For every initial condition* $\phi_0 \in \mathcal{C} \cup \mathcal{D}$*, there exists a nontrivial solution starting at* ϕ_0*. Furthermore, let* ϕ *be a maximal solution to the hybrid system* $\mathcal{H}$ *on* $\mathcal{R}$ *with initial condition* $\phi(0,0) = \phi_0 \in \mathcal{C} \cup \mathcal{D}$*. Then the following statements are true:*

(a) ϕ *is complete.*

(b) ϕ *has at most* N *consecutive jumps with no flow in between.*

(c) The amount of ordinary time between jumps is at most $\frac{2\pi}{\max(w_i)}$.

To prove Proposition 2.1 we need the following result.

Lemma 2.1 (Theorem S3 *in [36]*) *Suppose* $\tilde{\mathcal{H}}$ *is a well posed hybrid system and, for every* $\xi \in \mathcal{C} \cup \mathcal{D}$*, there exists a nontrivial solution to* $\tilde{\mathcal{H}}$ *starting from* ξ*. Let* x *be a maximal solution to* $\tilde{\mathcal{H}}$*. Then exactly one of the following three cases holds:*

(a) x is complete

(b) x blows up in finite hybrid time

(c) x eventually jumps out of $\mathcal{C} \cup \mathcal{D}$.

The proof of Proposition 2.1 follows.

Proof. To analyze existence, note that for every $\xi \in \mathcal{C} \backslash \mathcal{D}$ there exists $\sigma > 0$ and an absolutely continuous function $z : [0, \sigma] \to \mathbb{R}^n$ such that $z(0) = \xi$, $\dot{z}(t) = F(z(t))$ for almost all $t \in [0, \sigma]$ and $z(t) \in \mathcal{C}$ for all $t \in (0, \sigma]$. Note also that $G(\mathcal{D}) \subset \mathcal{C} \cup \mathcal{D}$. Then there exists a nontrivial solution from every initial condition in $[0, 2\pi]^N$.

Since $G(\mathcal{D}) \subset \mathcal{C} \cup \mathcal{D}$, condition (c) of Lemma 2.1 is not satisfied. Now it is convenient to point out that since $F(x) = [w, \ldots, w]^T$ is constant, it is globally Lipschitz, and there are no finite escape times. So, no maximal solution can satisfy condition (b) and therefore, all maximal solutions satisfy condition (a) of Lemma 2.1, i.e., are complete.

Note that $\cap \mathcal{D}_k \neq \emptyset$, i.e., a point $\tilde{\phi}(t, j) \in [0, 2\pi]^N$ might belong to more than one $\mathcal{D}_k$. By construction, the jump map is such that if $\tilde{\phi}(t, j)$ belongs to exactly m sets from the collection $\mathcal{D}_k$, with $m \leq N$, then there will be at least m consecutive jumps with no flow in between; moreover, after m jumps it is possible that $\tilde{\phi}(t, j + m)$ belongs to others $\mathcal{D}_k$ due to the coupling effect, and more jumps are required. In

any case, there will be at most N consecutive jumps with no flow in between since Assumption 1 gives $Q(0) = \{0\}$.

It follows that the amount of ordinary time between jumps is upper bounded by the fastest natural period of the network, $\frac{2\pi}{\max(w_i)}$. To see this, suppose that an oscillator has just fired and the fastest oscillator has phase equal to x_f. The next firing of the fastest oscillator will occur after an amount of time less or equal to $\frac{2\pi}{\max(w_i)}$, with equality if $x_f = 0$, unless it receives a pulse in the delay part of the PRC; however, this imply that a pulse was fired before an amount of time equal to $\frac{2\pi}{\max(w_i)}$ has elapsed from the previous jump. Then, the upper bound between jumps is $\frac{2\pi}{\max(w_i)}$. □

Remark 2.8 *Proposition 2.1 tells us that solutions behave as observed in biological systems and fulfill reasonable engineering expectations: they are complete and jump periodically. Statement (b) rules out the existence of Zeno or avalanche-firing-type solutions and statement (c) guarantees that jumps are persistent, i.e., it rules out the existence of solutions that only flow. It should also be noted that, in general, solutions to $\mathcal{H}$ are not unique, even if the PRC is a single-valued mapping. For example, consider a point $\tilde{\phi} \in [0, 2\pi]^N$ that belongs to more than one $\mathcal{D}_k$. The jumping rule is the union of the rules for each $\mathcal{D}_k$ and hence is set-valued and trajectories are not unique.*

2.3.3 Examples

In the following, we provide several examples of networks of PCOs that can be regarded as particular instances of our model. These examples include the SNIPER PRC, the Hodgkin-Huxley's model, as well as a system coupled through a phase response distribution (PRD), a relatively new concept that allows including uncertainty in the value of the PRC [2].

PCOs coupled via SNIPER PRC

The SNIPER PRC arises for neurons near a SNIPER bifurcation, i.e., a saddle-node bifurcation on a periodic orbit [29, 14] and is defined as $Z_d(1-\cos(x))$, where Z_d is a positive constant. Note that Assumption 1 holds for the SNIPER PRC. The phase space model of a neuron with natural firing frequency w_i and SNIPER PRC is given by (*cf.* Equation (2.5))

$$\dot{x_i} = w_i + Z_d(1 - \cos(x_i))u(t) \tag{2.12}$$

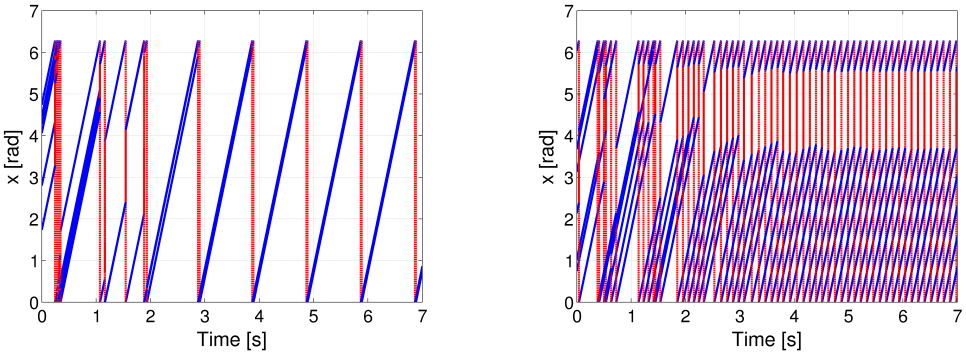
We can write a network of firing neurons coupled using the SNIPER PRC as a hybrid system $\mathcal{H}$ of the form (2.7)-(2.11) as follows:

$$\dot{x_i} = w_i, \quad x \in \mathcal{C} := [0, 2\pi]^N \tag{2.13}$$

$$\left.\begin{aligned} x_i^+ &= 0 \\ x_k^+ &= x_k + a_{ik}(1-\cos(x_k)) \end{aligned}\right\} =: G_i(x), \quad x \in \mathcal{D}_i \tag{2.14}$$

where $a_{ik} \in \{0, Z_d\}$, i.e., we consider the constant Z_d as the coupling strength and $Q(x) = 1 - \cos(x)$ as the PRC.

To show the behavior of a SNIPER-coupled network we conducted simulations, using a general purpose hybrid systems simulator [82, 83], of a 6-node all-to-all coupled network and a 6-node bidirectional cycle network from random initial conditions. We consider identical natural frequencies $w_i = w = 2\pi$ and coupling strength $Z_d = 1$. Figure 2.3(a) shows the phases of the oscillators for the 6-node all-to-all network. It can be seen that the network synchronizes asymptotically as time goes to infinity. Figure 2.3(b) shows the phases of the oscillators for the 6-node bidirectional cycle. It can be seen that the network converges to a phase locked state, i.e., the time between firings is constant. This behavior is well-known to exist in networks of PCOs. In fact, pulse-coupled time-division-multiple-access (TDMA) has been shown to emerge even in all-to-all coupled networks when the coupling is repulsive [23]. Note that the results presented are from a particular initial condition and no global behavior should be inferred.



(a) (b)

Figure 2.3: Simulations of the SNIPER PRC-coupled network. (a): Results for the all-to-all 6-node network; (b) Results for the 6-node bidirectional cycle.

PCOs coupled via Hodgkin-Huxley's PRC

The Hodgkin-Huxley's model is a four-dimensional conductance-based model for the membrane voltage dynamics of the squid giant axon [41]. The dynamics are given by:

$$C\dot{V} = -g_{Na}(V - V_{Na})m^3h - g_K(V - V_K)n^4 - g_L(V - V_L) + I_b \tag{2.15a}$$

$$\dot{m} = \frac{0.1V + 4}{1 - \exp(-4 - 0.1V)}(1 - m) - 4\exp(-\frac{V - 65}{18})m \tag{2.15b}$$

$$\dot{h} = 0.07\exp(-\frac{V - 65}{20})(1 - h) - \frac{1}{1 + \exp(-0.1V - 3.5)}h \tag{2.15c}$$

$$\dot{n} = \frac{0.01(V + 55)}{1 - \exp(-5.5 - 0.1V)}(1 - n) - 0.125\exp(-\frac{V + 65}{80}) \tag{2.15d}$$

where $g_{Na} = 120$, $V_{NA} = 50$, $g_K = 36$, $V_K = -77$, $g_L = 0.3$, $V_L = -54.4$, and $I_b = 10$. With the given parameter set, the system presents a periodic firing

behavior with a natural frequency of $w = 0.43\frac{rad}{ms}$, where the voltage variable V shows an impulsive-like reset from $V \approx 30mV$ to $V \approx -70mV$.

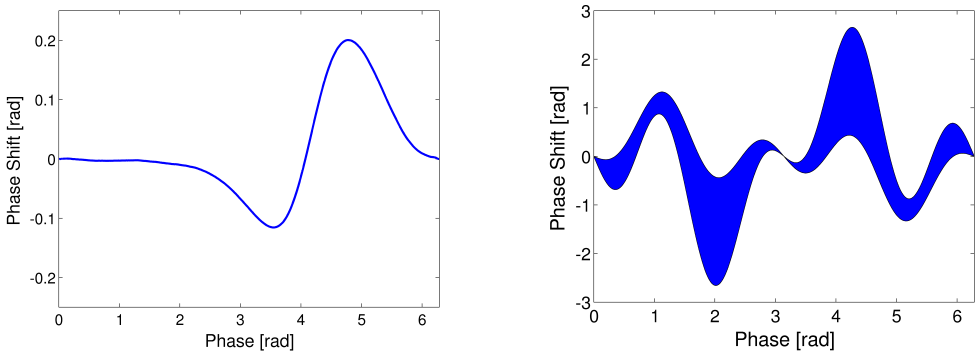
The four-dimensional state-space model of the form (1.1) can be reduced to a phase model of the form (2.5) via phase reduction; however, the PRC must be obtained numerically. Figure 2.4(a) shows the Hodgkin-Huxley's PRC for the parameter set given above, which was obtained numerically in Matlab. Using the PRC, we can write a network of firing neurons coupled using the Hodgkin-Huxley's PRC as a hybrid system $\mathcal{H}$ of the form (2.7)-(2.11) as follows:

$$\dot{x}_i = w_i, \quad x \in \mathcal{C} := [0, 2\pi]^N \tag{2.16}$$

$$\left.\begin{aligned} x_i^+ &= 0 \\ x_k^+ &= x_j + a_{ik}Q(x_k) \end{aligned}\right\} =: G_i(x), \quad x \in \mathcal{D}_i \tag{2.17}$$

where $a_{ik} \in \{0, l\}$, and $Q(x)$ is given by the PRC in Figure 2.4(a). Note that Assumption 1 holds and that the graph of the Hodgkin-Huxley's PRC is contained in the set Ω, hence the saturation function can be removed from the formulation of the model.

To show the behavior of a network of pulse-coupled oscillators coupled using the Hodgkin-Huxley's PRC, we conducted simulations of a 6-node all-to-all coupled network and a 6-node bidirectional cycle network from random initial conditions.



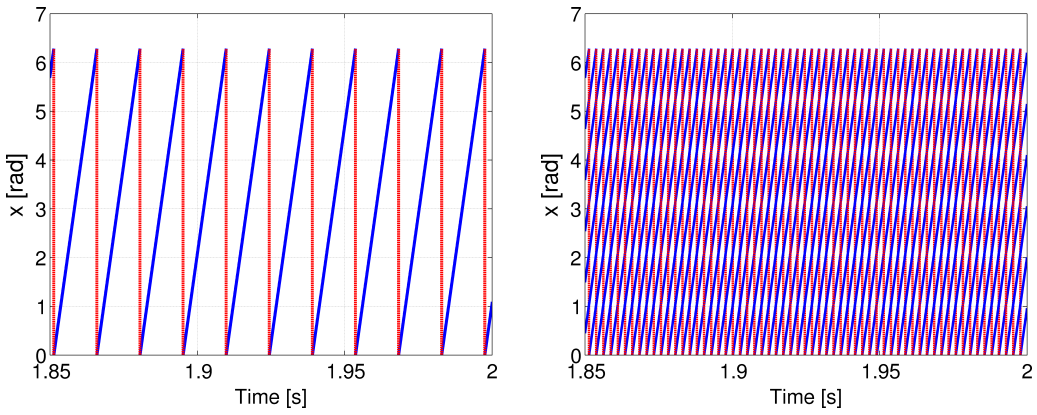
(a) (b)

Figure 2.4: (a): Hodgkin-Huxley's PRC obtained numerically using Matlab; (b) Phase response distribution (PRD) used in the simulations.

We consider identical natural frequencies $w_i = w = 0.43\frac{rad}{ms}$ and coupling strength $l = 1$. Figure 2.5(a) shows the phases of the oscillators for the 6-node all-to-all network. It can be seen that the network synchronizes. Figure 2.5(b) shows the phases of the oscillators for the 6-node bidirectional cycle. It can be seen that the network converges to a phase locked state, i.e., the time between firings is constant. As in the previous example, the results presented are from a particular initial condition and no global behavior should be inferred.

PCOs coupled via a phase response distribution

In some systems, it is impossible to obtain the PRC with absolute certainty due to the existence of process or measurement noise. For these systems, it is useful to allow the PRC to be multi-valued in order to capture the complete possible dynamics. Such multi-valued PRC is known as the phase response distribution



(a) (b)

Figure 2.5: Simulations of the Hodgkin-Huxley's PRC-coupled network. (a): Results for the all-to-all 6-node network; (b) Results for the 6-node bidirectional cycle.

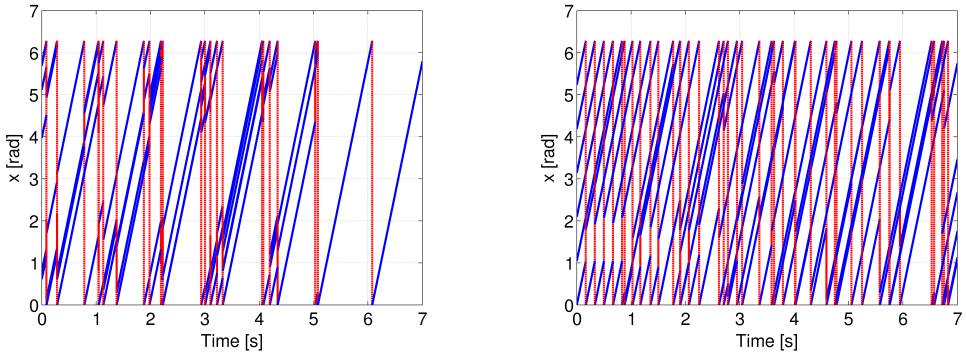
(PRD) of the system [2]. We illustrate the use of a PRD and how our modeling framework can handle the PRD with an example. In the following, we consider that the PRC is given by the set-valued mapping shown in Figure 2.4(b). For simulation purposes, at each phase point x_i the phase shift is chosen randomly from the corresponding set $Q(x_i)$ following a uniform distribution. Hence, a simulation corresponds to a realization of the system. However, our modeling framework allows writing the whole dynamics and the analysis can be done for all solutions instead of for a particular realization. We can write a network of PCOs coupled using the PRD as a hybrid system $\mathcal{H}$ of the form (2.7)-(2.11) as follows:

$$\dot{x}_i = w_i, \quad x \in \mathcal{C} := [0, 2\pi]^N \tag{2.18}$$

$$\left.\begin{aligned} x_i^+ &= 0 \\ x_k^+ &\in \mathrm{sat}_0^{2\pi}(x_k + a_{ik}Q(x_k)) \end{aligned}\right\} =: G_i(x), \quad x \in \mathcal{D}_i \tag{2.19}$$

where $a_{ik} \in \{0, l\}$, and $Q(x_i)$ is given by the PRD in Figure 2.4(b). Note that the graph of the PRD is not contained entirely inside the set Ω, hence the saturation function must be kept in the model formulation and it will play an active role for some realizations of the system. As in the previous examples, Assumption 1 holds.

To show the behavior of a network of pulse-coupled oscillators coupled using the PRD in Figure 2.4(b), we conducted simulations of a 6-node all-to-all coupled network and a 6-node bidirectional cycle network from random initial conditions. We consider identical natural frequencies $w_i = w = 2\pi$ and coupling strength $l = 1$. As previously mentioned, at each phase point x_i the phase shift is chosen randomly from the corresponding set $Q(x_i)$ following a uniform distribution. Figure 2.6(a) shows the phases of the oscillators for the 6-node all-to-all network. It can be seen that the network synchronizes. Figure 2.6(b) shows the phases of the oscillators for the 6-node bidirectional cycle. It can be seen that the network does not synchronize, in fact it shows a chaotic behavior. Note that the results presented are from a particular initial condition and for a particular realization, hence no global behavior should be inferred.



(a) (b)

Figure 2.6: Simulations of the PRD-coupled network. (a): Results for the all-to-all 6-node network; (b) Results for the 6-node bidirectional cycle.

The previous examples illustrate the flexibility of our model (2.7)-(2.11) to handle a variety of pulse-coupled networks. In the next chapter, we use the model (2.7)-(2.11) to derive conditions that ensure synchronization in networks of PCOs.

Chapter 3

Synchronization of Pulse-Coupled Oscillators

In this chapter we present synchronization conditions for networks of PCOs interacting on a variety of coupling topologies and network structures. The following definitions of phase and frequency synchronization will be used throughout the chapter.

Definition 3.1 (Phase Synchronization) *Consider a PCO network. We say that the network synchronizes in phase if, for j sufficiently large, at every I_j with nonempty interior and for all $t \in I_j$ we have $x_i(t,j) = x_k(t,j)$ for every pair of oscillators i, $k \in \mathcal{V}$.*

Definition 3.2 (Frequency Synchronization) *Consider a PCO network, let (t_{i_k}, j_{i_k}) and $\left(t_{i_{k+1}}, j_{i_{k+1}}\right)$ denote time instants of consecutive visits of the oscillator i to the set $\mathcal{D}_i$. We say that the network synchronizes in frequency if there*

Portions of this Chapter have been published in [71] and submitted for publication to Automatica [70], and Systems and Control Letters [72].

exists a positive T, the collective period, and $J \geq 0$ such that $t_{i_{k+1}} - t_{i_k} = T$ holds for every oscillator when $j_{i_k} > J$.

It is clear from the definitions that phase synchronization implies frequency synchronization. In the sequel, we will use the terms synchronization and phase synchronization interchangeably.

3.1 Synchronization of PCOs to a Global Cue

A particular network structure that is of great interest is the one when an omnipresent leader is part of the network, which we will denote as the global cue or master node. In this setup, the network under analysis consists of a global cue and N slave oscillators aiming to synchronize their phases to the phase of the global cue. We assume that the slave oscillators interact following a given graph $\mathcal{R}$, not necessarily connected. Each oscillator modifies its phase following its natural frequency and using entrainment information received in the form of pulses from the global cue and neighbor oscillators. The global cue is not affected by pulses, thus, its phase evolution is determined only by its natural frequency. We can rewrite the model (2.7)-(2.11) after the addition of a global cue as follows:

$$\left.\begin{array}{l} \dot{x}_g = w_g \\ \dot{x}_i = w_i \end{array}\right\} =: F(x), \quad x \in \mathcal{C} \tag{3.1}$$

$$\left.\begin{aligned} x_g^+ &= 0 \\ x_i^+ &\in \mathrm{sat}_0^{2\pi}(x_i + g_i Q_g(x_i)) \end{aligned}\right\} =: G_g(x),\ x \in \mathcal{D}_g \tag{3.2}$$

$$\left.\begin{aligned} x_g^+ &= x_g \\ x_i^+ &= 0 \\ x_k^+ &\in \mathrm{sat}_0^{2\pi}(x_k + a_{ik} Q_l(x_k)) \end{aligned}\right\} =: G_i(x),\ x \in \mathcal{D}_i \tag{3.3}$$

where $\mathcal{C} := \{x \in [0, 2\pi]^{N+1}\}$, $\mathcal{D}_g := \{x \in \mathcal{C} : x_g = 2\pi\}$, $\mathcal{D}_i := \{x \in \mathcal{C} : x_i = 2\pi\}$, $w_g \in \mathbb{R}_{>0}$ is the natural frequency of the global cue, $g_i \in [0, 1)$ is the global coupling strength, and $Q_g : [0, 2\pi] \rightrightarrows R$, $Q_l : [0, 2\pi] \rightrightarrows R$ are the global and local PRC. Then, we define the jump set as:

$$\mathcal{D} := \mathcal{D}_g \cup \bigcup_{i \in \mathcal{V}} \mathcal{D}_i \tag{3.4}$$

and the corresponding jump map as:

$$G(x) := \bigcup_{i \in \mathcal{V} \cup \{g\} : x \in \mathcal{D}_i} G_i(x) \tag{3.5}$$

In the following we will refer to the model (3.1)-(3.5) as the hybrid system $\mathcal{H}_g$.

3.1.1 The Identical Natural Frequencies Case

In this section we analyze the synchronization properties of the hybrid system $\mathcal{H}_g$ in the ideal case when there is no frequency drift between the oscillators and the global cue. In the following, we refer to an *arc* as a connected subset of $[0, 2\pi]$ where 0 and 2π are associated with each other. To conduct the analysis, we use the following:

Assumption 2 *The global cue and the slave oscillators have identical natural frequencies, i.e.,* $w_g = w_i = w,\ \forall i \in \mathcal{V}$.

Assumption 3 *We assume identical and strictly positive global coupling, i.e.,* $g_i = g > 0,\ \forall i \in \mathcal{V}$.

Remark 3.1 *Note that in [102] it is stated that the condition* $g_i = g > 0,\ \forall i \in \mathcal{V}$ *is necessary to ensure synchronization when the oscillators are distributed in the whole interval* $[0, 2\pi]$. *Moreover, the condition* $w_g = w_i = w$ *is required to ensure perfect synchronization in phase.*

To analyze synchronization, define the difference between the global cue and the ith slave oscillator as $\xi_i(t, j) := x_g(t, j) - x_i(t, j)$ and the vector of differences as $\xi = [\xi_1, \ldots, \xi_N]$. We consider synchronization achieved whenever $|\xi_i| = 0$ or

$|\xi_i| = 2\pi \ \forall i$. Hence the synchronization set can be written as:

$$\mathcal{A} := \{x \in \mathcal{C} : \xi_i = 0 \text{ or } \xi_i = \pm 2\pi \, \forall i \in \mathcal{V}\} \tag{3.6}$$

The synchronization condition is as follows.

Theorem 3.1 *Consider the network of PCOs given by $\mathcal{H}_g$. If:*

1. *Assumptions 1, 2, and 3 hold*
2. *Q_g and Q_l are such that if $x_i \in (\pi, 2\pi)$, then $\forall \bar{q} \in Q_q(x_i),\ q \in \{g, l\},\ \bar{q} > 0$; and if $x_i \in (0, \pi)$, then $\forall \underline{q} \in Q_q(x_i),\ q \in \{g, l\},\ \underline{q} < 0$,*
3. *the influence of the global cue is strong enough compared with the local coupling*

and moreover, the PRCs satisfy $0 \notin Q_q(\pi),\ q \in \{g, l\}$, then the network synchronizes from every initial condition $x(0, 0) \in \mathcal{C}$.

Proof. Consider the following family of functions representing the distance to the synchronization set $\mathcal{A}$:

$$V_i(\xi(t, j)) = \min\left(|\xi_i(t, j)|, 2\pi - |\xi_i(t, j)|\right). \tag{3.7}$$

Note that the V_i are continuous functions and positive definite with respect to $\mathcal{A}$. It is clear due to Assumption 2 that the V_i functions are unchanged during flows.

Hence, we will focus on the underlying discrete-time system to study synchronization.

We will prove synchronization in two steps. First, we will show that there exists a forward invariant neighborhood of $\mathcal{A}$, denoted as $\mathcal{B}$, such that if the state belongs to $\mathcal{B}$, then the network synchronizes irrespective of the strength of the couplings. Secondly, we will show that the network eventually reaches $\mathcal{B}$ from every initial condition, if the global coupling is strong enough.

In the following, we make use of the concept of containing arc. Given an arc α, i.e., a connected subset of $[0, 2\pi]$ with 0 and 2π mapped to each other, with associated length d, the oscillators are contained in α if and only if $x_i \in \alpha,\ \forall i \in \mathcal{V}$.

To prove synchronization, we will show that if the smallest containing arc α has length $d < \pi$ the network synchronizes to the global cue for all gQ_g and lQ_l. To this end, whenever $x(t, j) \in \mathcal{D}$, define $i_* := \arg\min_{i\in\mathcal{V}} x_i$, $x_i \in (\pi, 2\pi]$ and $i^* := \arg\max_{i\in\mathcal{V}} x_i$, $x_i \in [0, \pi)$ or $i^* := \arg\max_{i\in\mathcal{V}} x_i$, $x_i \in (\pi, 2\pi]$ if $\{i : x_i \in [0, \pi)\}$ is empty (note that they can be a set, or the global cue, yet the following arguments still hold in that case). A key observation is that when the oscillators are contained in an arc of length $d < \pi$, at any time instant (t, j) such that $x(t, j) \in \mathcal{D}$, the length of the arc is given by $d(\xi) = V_{i_*}(\xi)+V_{i^*}(\xi)$. In the following, consider the Lyapunov candidate $W(\xi) = d(\xi) = V_{i_*}(\xi) + V_{i^*}(\xi)$, and for every $\mu \in \mathbb{R}_{\geq 0}$ define the set $L_v(\mu) := \{x \in \mathcal{C} : d(\xi) = \mu\}$. We focus on the initial conditions contained in

$\mathcal{B} := \{x \in \mathcal{C} : d(\xi) < \pi\}$. Suppose $x \in (\mathcal{D} \cap \mathcal{B}) \setminus \mathcal{A}$, then if $x \in \mathcal{D}_g$ we have $W^+ - W < 0$ since both i_* and i^* will be attracted by the global cue. When $x \in \mathcal{D}_i$, since $x_{i_*} \in (\pi, 2\pi]$ it will be advanced, if affected, and similarly, since $x_{i^*} \in [0, \pi)$ it will be advanced, if affected. Therefore, we have that $W^+ - W < 0$ when either $i \in \mathcal{N}^{i_*-}$ or $i \in \mathcal{N}^{i^*-}$, or we have $W^+ - W = 0$ when both $i \notin \mathcal{N}^{i_*-}$ and $i \notin \mathcal{N}^{i^*-}$ hold. Then, $W^+ - W \leq 0$ for all $x \in (\mathcal{D} \cap \mathcal{B}) \setminus \mathcal{A}$. However, since every solution is complete and the global cue jumps periodically, for every $\mu > 0$ no complete solution to $\mathcal{H}_g$ remains in $\mathcal{B} \cap L_v(\mu)$. Since the hybrid system $\mathcal{H}_g$ is well-posed, we can rely on the invariance principle to establish synchronization. In particular, directly applying Theorem 23 in [36] gives asymptotic stability of the set $\mathcal{A}$ with basin of attraction $\mathcal{B}$.

Now consider $x(0,0) \in \mathcal{C} \setminus \mathcal{B}$ and define $W(t,j) := \sum_{i \in \mathcal{V}} V_i(t,j)$ as the total distance of the system to the global cue. A sufficient condition for $d < \pi$ is $W < \pi$. Hence, if there exist a time instant (T, J) such that $W(T, J) < \pi$, then the network synchronizes.

Suppose $x \in \mathcal{D}$. We analyze the change in V_i when x_g jumps. We have that $x_g = 2\pi$ and $x_i \in [0, 2\pi]$, then:

$$\begin{aligned} V_i &= \min\{2\pi - x_i, x_i\} \\ V_i^+ &\in \min\{x_i + gQ_g(x_i), 2\pi - (x_i + gQ_g(x_i))\} \end{aligned} \tag{3.8}$$

Since the phase is advanced if $x_i \in (\pi, 2\pi)$, and the phase is delayed if $x_i \in (0, \pi)$, then, $V_i > V_i^+$ holds for all $x \notin \mathcal{A}$ independent of the value of x_i before x_g jumps. Hence the distance to the set $\mathcal{A}$ is reduced. It is easy to see that when x_i jumps we have $V_i = V_i^+$ since no update on x_g occurs. Now let us analyze the change in V_i when an oscillator m, $m \neq i$ with $m \in \mathcal{N}^{i-}$, jumps. In this case we have:

$$\begin{aligned} V_i &= \min\{|x_g - x_i|, 2\pi - |x_g - x_i|\} \\ V_i^+ &\in \min\{|x_g - (x_i + lQ_l(x_i))|, 2\pi - |x_g - (x_i + lQ_l(x_i))|\} \end{aligned} \tag{3.9}$$

Whether V_i increases or decreases after jumps is unknown. In fact, it depends on the phase difference between the oscillator i and the global cue. However, we can bound the possible increase of V_i. To this end first note that each oscillator fires at most 2 times per cycle of the global cue, i.e., each oscillator fires at most 2 times in an interval of ordinary time of length $\frac{2\pi}{w}$. This is true since from Assumption 2 the natural frequencies are equal and for an oscillator that has just fired to fire again as quickly as possible, it must receive a pulse after its phase reaches π (since the phase is delayed if $x_i \in (0, \pi)$), then it cannot fire more than 2 times per cycle of the global cue.

Now we can bound the possible increase of V_i, between the jumps k and $k+1$ of the global cue, by using information regarding the neighbors of the ith oscillator as

$2l|\mathcal{N}^{i-}| \max_{q\in Q_l(y), y\in[0,2\pi]} |q|$. To prove convergence to $\mathcal{B}$, let (t_k, j_k), $k = 1, 2, \ldots$, be the time instants of the kth jump of the global cue and note that if ϕ is a solution to the system, then (t_k, j_k) and $(t_k, j_k + 1) \in \text{dom } \phi$ and $\phi(t_k, j_k) \in \mathcal{D}_g$. If the influence of the global cue gQ_g is strong enough compared with the local coupling lQ_l so that

$$W(t_{k+1}, j_{k+1}) \leq \sum_{i\in\mathcal{V}} \left\{V_i(t_k, j_k + 1) + 2l|\mathcal{N}^{i-}| \max_{q\in Q_l(y), y\in[0,2\pi]} |q|\right\} < W(t_k, j_k) \quad (3.10)$$

holds, then, the total distance to the global cue is reduced in one global cue cycle. Then, we can restrict gQ_g and lQ_l to ensure that the influence of the global cue is strong enough compared with the local coupling, so that the previous inequality holds whenever $W \geq \pi$, hence ensuring convergence to the set $\mathcal{B}$. Therefore, the network eventually enters $\mathcal{B}$ and hence it synchronizes from every initial condition $x(0, 0) \in \mathcal{C}$. □

Remark 3.2 *Note that Theorem 3.1 does not impose any connectivity requirement on the communication graph $\mathcal{R}$ or uses any assumption on the monotonicity of the PRCs Q_g and Q_l. This suggests that Theorem 3.1 is valid for a class of oscillators larger than monotone oscillators, for which the PRC is monotone [57].*

Theorem 3.1 emphasizes the role of the global coupling, which should be strong enough compared with the local coupling. A practical bound can be stated as follows.

Corollary 3.1 *Consider the network of PCOs given by $\mathcal{H}_g$ and suppose conditions 1 and 2 in Theorem 3.1 hold. If the global coupling gQ_g is such that:*

1. *for all $x \in [\frac{\pi}{2}, \pi)$, if $\underline{q} \in Q_g(x)$ then $g\underline{q} < \frac{\pi}{2} - x$*

2. *for all $x \in (\pi, \frac{3\pi}{2}]$, if $\bar{q} \in Q_g(x)$ then $g\bar{q} > \frac{3\pi}{2} - x$*

3. *if $q \in Q_g(\pi)$ then $gq \in [-\pi, -\frac{\pi}{2}) \cup (\frac{3\pi}{2}, 2\pi]$*

then the network synchronizes for all $x(0,0) \in \mathcal{C}$ irrespective of the strength of the local coupling lQ_l.

Proof. Conditions 1, 2, and 3 ensure that after the first jump of the global cue the oscillators will be contained in an arc α of length $d < \pi$. Hence the network synchronizes from every initial condition. □

The previous Corollary gives a practical bound since the global coupling does not need to be stronger than the coupling characterized in Corollary 3.1.

In the rest of this chapter, we will consider that the feedback strategy is given by the optimal advance-delay PRC:

$$Q(x) = \begin{cases} 2\pi - x, & \text{if } x > \pi \\ \{\pi, -\pi\}, & \text{if } x = \pi \\ -x, & \text{if } x < \pi \end{cases} \tag{3.11}$$

which corresponds to the set-valued regularization of the discontinuous function $Q(x) = 2\pi - x$, $x \in [\pi, 2\pi]$; $Q(x) = -x$, $x \in [0, \pi)$. Note that (3.11) is an outer semi-continuous set-valued mapping and bounded; hence Assumption 1 holds. Moreover, the graph of (3.11) lies entirely inside the set Ω (*cf.* Figure 2.2). The PRC (3.11) has been proven to be optimal in terms of synchronization rate in [100] and thus, it will be used in the rest of this dissertation. Figure 3.1 shows the graph of the PRC (3.11).

3.1.2 The Non-identical Natural Frequencies Case

In this section, we analyze synchronization of PCO networks when the natural frequencies are non-identical. To conduct the analysis, we use the following assumption:

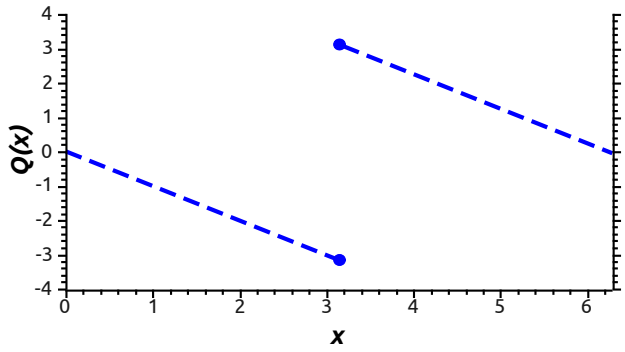


Figure 3.1: Graph of PRC used in this work, which is optimal in terms of synchronization rate.

Assumption 4 *The frequency drift between a pair of oscillators is small, i.e.,* $\Delta w_{ik} := w_i - w_k \ll 1$ *for every* $i, k \in \{g\} \cup \mathcal{V}$.

The synchronization condition is as follows

Theorem 3.2 *Consider the network of non-identical PCOs given by* $\mathcal{H}_g$ *and suppose that Assumptions 3 and 4 hold. Moreover, consider* $w_i = (1 + \epsilon_i)w_g$ *with* $\epsilon_i \in [-\bar{\epsilon}, \bar{\epsilon}]$, *define the frequency drift with respect to the global cue as:* $\Delta w_{gi} := w_g - w_i = -\epsilon_i w_g$, *and assume that the coupling is such that* $g > \frac{1+\bar{\epsilon}}{2}$, $2|\epsilon_i| + l|\mathcal{N}^{i-}| \leq g$, $2|\epsilon_i| \geq l|\mathcal{N}^{i-}|$, *and* $(1-g)l(1+|\mathcal{N}^{i+}|) < g$ *holds for every oscillator. Then, when the PRC is given by (3.11) the network synchronizes in frequency, with collective period equal to the period of the global cue,* $\frac{2\pi}{w_g}$, *from every initial condition* $x(0,0) \in \mathcal{C}$.

Proof. First, note that if an oscillator fires with a given period T, it satisfies the following necessary equilibrium condition $\delta x_i = 2\pi - w_i T$, where δx_i is the total phase change due to pulses in an interval of length T. In particular, for an oscillator to fire with the period of the global cue this reduces to: $\delta x_i = 2\pi\left(1 - \frac{w_i}{w_g}\right) = -2\pi\epsilon_i$. Since $g > 2\bar{\epsilon}$, we have that $\max(\delta x_i) \geq g\pi > 2\pi\bar{\epsilon}$, and $\min(\delta x_i) \leq -g\pi < -2\pi\bar{\epsilon}$, and hence the period of the global cue is inside the bounds. Therefore, synchronization in frequency is feasible, i.e., there exists an equilibrium point $x^* \in [0, 2\pi]^N$ such that if $x = x^*$ then the network is synchronized in frequency with collective period equal to the period of the global cue. In the following, we will prove that the network converges to x^*.

To this end, we need to prove that the network reaches a state in which every oscillator fires one time per cycle of the global cue. Consider that the global cue has just fired, and then, the phase of each oscillator x_i belongs to $[0, (1-g)\pi] \cup [(1+g)\pi, 2\pi]$. Consider the case $w_i \geq w_g$ and $x_i \in [(1+g)\pi, 2\pi]$. We focus on the distance from i to the global cue, defined as $\min\left(|x_g - x_i|, 2\pi - |x_g - x_i|\right)$, which at this instant is given by $2\pi - x_i \leq (1-g)\pi$. In this case, the distance will decrease during flows (and also if a pulse is received since $x_i \in [(1+g)\pi, 2\pi]$) until x_i jumps and then, while $x_i < x_g$, it will keep decreasing during flows. The worst situation is when $x_i = 2\pi$ (the distance is 0) since x_i will jump immediately and next, during flows, the distance will always increase. However, in an interval of length $\frac{2\pi}{w_g}$, the

distance will increase at most $2|\epsilon_i|\pi$ plus the increase due to neighbor pulses, which can be at most $(1-(1-l)^{|\mathcal{N}^{i-}|})\pi$, since $2|\epsilon_i|+(1-(1-l)^{|\mathcal{N}^{i-}|}) \leq 2|\epsilon_i|+l|\mathcal{N}^{i-}| \leq g < 1$ (by applying Bernoulli's inequality to $(1-l)^{|\mathcal{N}^{i-}|}$), then just after a future jump of the global cue, the phase of the oscillator i will be in the interval $[0, (1-g)\pi]$. Consider now the case when $x_i \in [0, (1-g)\pi]$. During flows, in an interval of length $\frac{2\pi}{w_g}$, the distance will increase at most $2|\epsilon_i|\pi$. Regarding changes due to neighbor pulses, the worst case is when the oscillator receives $|\mathcal{N}^{i-}|$ pulses at phase π, and then the maximum increase is bounded by $(1-(1-l)^{|\mathcal{N}^{i-}|})\pi$. Since $2|\epsilon_i|\pi + (1-(1-l)^{|\mathcal{N}^{i-}|})\pi \leq g\pi$, the increase is not enough to have $x_i = \pi$ just before the jump of the global cue. Similarly, if we have $x_i = 0$, during flows in an interval of length $\frac{2\pi}{w_g}$, the distance will increase at most $2|\epsilon_i|\pi$ and the phase can be delayed at most by $l|\mathcal{N}^{i-}|\pi$. Since $2|\epsilon_i| - l|\mathcal{N}^{i-}| \geq 0$, after the next global cue jump, and after all the following, we will have $x_i \in [0, (1-g)\pi]$. Using a similar argument, under these conditions the case $[(1+g)\pi, 2\pi]$ has the same properties for every oscillator slower than the global cue. Hence, there exists J such that for $j > J$ every oscillator fires one time per cycle of the global cue.

Now, for $j > J$, consider time instants when $x_g = 0$, i.e., the global cue has just jumped. We drop the time index t of the hybrid time domain and with abuse of notation, we denote as $x(j)$ the state after the j jump of the global cue. We will show that $x(j)$ converges to the equilibrium x^* as j goes to infinity. We have that

at time instants when the global cue jumps the system can be seen as the following discrete-time non-linear system:

$$x(j+1) \in x(j) + \mathcal{G}(x(j), \epsilon, \Delta(x(j))) \tag{3.12}$$

where $x(j) \in [0, 2\pi]^N$ denote the entire state of the system after the jump j of the global cue, ϵ is a vector whose components are the individual ϵ_i, and $\Delta(x(j))$ is a vector whose component i corresponds to the phase update for the oscillator i due to pulses received between jumps j and $j+1$ of the global cue, denote this as $\Delta_i(x(j))$.

It follows that at the equilibrium, every oscillator i updates its phase by 2π during a time interval of length $\frac{2\pi}{w_g}$. Therefore, the network is synchronized in frequency. We need to prove that the point $x^* \in [0, 2\pi]^N$ is globally attractive. To do so, we analyze the system component-wise. Component-wise the system is:

$$x_i(j+1) \in x_i(j) + \mathcal{G}_i(x_i(j), \epsilon_i, \Delta_i(x(j))) \tag{3.13}$$

where $\mathcal{G}_i(x_i, \epsilon_i, \Delta_i(x)) = 2\pi\epsilon_i + \Delta_i(x) + gQ(x_i + 2\pi\epsilon_i + \Delta_i(x_i))$ with $\mathcal{G}_i(x_i, \epsilon_i, \Delta_i(x)) = 0$ if $x = x^*$.

From the previous analysis, we know that for $\epsilon_i > 0$, and $j > J$, we have $x_i(j) \in [0, (1-g)\pi]$, $x_i(j+1) \in [0, (1-g)\pi]$ and therefore $x_i^* \in [0, (1-g)\pi]$. The system reduces to:

$$x_i(j+1) \in (1-g)x_i(j) + (1-g)2\pi\epsilon_i + (1-g)\Delta_i(x(j)) \tag{3.14}$$

At the equilibrium we have:

$$(1-g)2\pi\epsilon_i + (1-g)\Delta_i(x^*) - gx_i^* = 0 \tag{3.15}$$

Define $Dx_i(j) := x_i(j) - x_i^*$ and $D\Delta_i(x(j)) := \Delta_i(x) - \Delta_i(x^*)$. Combining the two previous equations yields:

$$x_i(j+1) \in x_i(j) - gDx_i(j) + (1-g)D\Delta_i(x(j)) \tag{3.16}$$

Subtracting x_i^* at both sides gives:

$$Dx_i(j+1) \in (1-g)Dx_i(j) + (1-g)D\Delta_i(x(j)) \tag{3.17}$$

Similarly, for $\epsilon_i < 0$ we know that $x_i(j) \in [(1+g)\pi, 2\pi]$, $x_i(j+1) \in [(1+g)\pi, 2\pi]$ and therefore $x_i^* \in [(1+g)\pi, 2\pi]$. The system reduces to:

$$x_i(j+1) \in (1-g)x_i(j) + (1-g)2\pi\epsilon_i + (1-g)\Delta_i(x(j)) + g2\pi \tag{3.18}$$

At the equilibrium we have:

$$(1-g)2\pi\epsilon_i + (1-g)\Delta_i(x^*) - gx_i^* + g2\pi = 0 \tag{3.19}$$

Combining the two previous equations yields:

$$x_i(j+1) \in x_i(j) - gDx_i(j) + (1-g)D\Delta_i(x(j)) \tag{3.20}$$

Subtracting x_i^* at both sides gives:

$$Dx_i(j+1) \in (1-g)Dx_i(j) + (1-g)D\Delta_i(x(j)) \tag{3.21}$$

Then, the dynamics of the deviation with respect to the equilibrium reduces to:

$$Dx_i(j+1) \in (1-g)Dx_i(j) + (1-g)D\Delta_i(x(j)) \tag{3.22}$$

We need to characterize the term $D\Delta_i(x(j))$. To this end, recall that if $\epsilon_i > 0$, then for all $j > J$, x_i belongs to $[0, (1-g)\pi]$ and if $\epsilon_i < 0$, x_i belongs to $[(1+g)\pi, 2\pi]$. Consider $x_1 = (1-g)\pi$ (recall x_1 is the faster oscillator) and $x_N = (1+g)\pi$ (recall x_N is the slowest). Since $g > \frac{1+\bar{\epsilon}}{2}$, x_N must have fired before $x_1 = \pi$. Then, when $j > J$ every oscillator for which $\epsilon_i > 0$ receives pulses from neighbors only in the interval $[0, \pi)$, and every oscillator for which $\epsilon_i < 0$ receives pulses from neighbors only in the interval $(\pi, 2\pi]$. Now, consider that an oscillator for which $\epsilon_i > 0$ should have received a pulse at phase $x_i = x_{i,1}$ but instead, it received the pulse at phase $x_i = x_{i,2}$. The nominal phase shift is given by $-lx_{i,1}$ and the actual phase shift is given by $-lx_{i,2}$. The difference between the nominal phase shift and the actual phase shift is given by: $-l(x_{i,2} - x_{i,1})$. Similarly, for an oscillator with $\epsilon_i < 0$ the nominal phase shift is given by $l(2\pi - x_{i,1})$ and the actual phase shift is given by $l(2\pi - x_{i,2})$. The difference between the nominal phase shift and the actual phase shift is also given by $-l(x_{i,2} - x_{i,1})$. With this in mind, we can write:

$$|D\Delta_i(x(j))| \leq \sum_{k \in \mathcal{N}^{i-} \cup \{i\}} l|Dx_k(j)| \tag{3.23}$$

Now, consider the following Lyapunov function:

$$V(Dx) = \sum_{i=1}^{N} |Dx_i| \tag{3.24}$$

We have that:

$$V(Dx(j+1)) = \sum_{i=1}^{N} |Dx_i(j+1)| \leq (1-g) \sum_{i=1}^{N} |Dx_i(j)| + (1-g)l \sum_{i=1}^{N} \sum_{k \in \mathcal{N}^{i-} \cup \{i\}} |Dx_k(j)| \tag{3.25}$$

Which gives:

$$V(Dx(j+1)) - V(Dx(j)) \leq \sum_{i=1}^{N} (-g + (1-g)l(1 + |\mathcal{N}^{i+}|))|Dx_i(j)| \tag{3.26}$$

which is negative since $(1-g)l(1+|\mathcal{N}^{i+}|) < g$. Then $Dx \to 0$ as $j \to \infty$, hence the network synchronizes in frequency from every initial condition. □

3.2 Decentralized Synchronization of PCOs

In this section we analyze the synchronization properties of decentralized PCO networks with identical natural frequencies. To this end, define the synchronization set as $\mathcal{S} := \{x \in \mathcal{C} : |x_i - x_{i+1}| = 0 \text{ or } |x_i - x_{i+1}| = 2\pi, \forall i \in \mathcal{V}\}$, with the understanding that node $N+1$ is mapped to node 1 (and node 0 to node N in the following). We say that the network synchronizes if the state x converges to the set $\mathcal{S}$ (note that this is consistent with the definition of phase synchronization given at the beginning of the chapter).

Consider the following family of functions:

$$v_{i,k}(x) = \min\left(|x_i - x_k|, 2\pi - |x_i - x_k|\right) \tag{3.27}$$

Note that $v_{i,k}(x)$ represents the length of the shortest segment joining oscillators i and k, i.e., the smallest arc containing i and k.

To appreciate the usefulness of the $v_{i,k}$ functions, consider the following example.

Example 1: Consider the 2-oscillator network depicted in Figure 3.2(a). Let $x_1(0,0)$, $x_2(0,0) \in [0, 2\pi]$, and note that the length of the containing arc is $v_{1,2}$; since the oscillators have identical natural frequencies, $v_{1,2}$ is not changed during flows. We analyze the change in $v_{1,2}$ after firing events starting w.l.o.g. with oscillator 1. We have that $x_1 = 2\pi$ and $x_2 \in [0, 2\pi]$, then:

$$v_{1,2}(x) = \min\left\{2\pi - x_2, x_2\right\} \tag{3.28}$$

$$v_{1,2}^{+}(x) = \begin{cases} x_2(1-l), & \text{if } x_2 \in [0, \pi] \\ (2\pi - x_2)(1-l), & \text{if } x_2 \in [\pi, 2\pi] \end{cases} \tag{3.29}$$

therefore, $v_{1,2}(x) > v_{1,2}^{+}(x) = (1-l)v_{1,2}(x)$ always holds. Similarly, when oscillator 2 fires $v_{1,2}(x) > v_{1,2}^{+}(x) = (1-l)v_{1,2}(x)$ holds. Then, after every firing event we

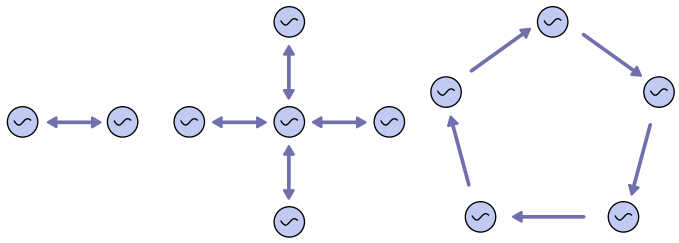


Figure 3.2: Networks used in the examples. (a): Network of N=2 PCOs; (b): Star network, N=5; (c): Unidirectional cycle with N=5.

have that $v_{1,2}(x) - v_{1,2}^{+}(x) = -lv_{1,2}(x)$. Hence, the 2-oscillator network asymptotically synchronizes from every initial condition $x(0,0) \in \mathcal{C}$, for every $l \in (0,1)$; or synchronizes after 1 jump if $l = 1$ □

3.2.1 The All-to-All Case

In this section, we focus on complete (all-to-all) graphs, i.e., $a_{ij} = l,\ \forall i,j \in \mathcal{V}$.

Theorem 3.3 *Consider a network of PCOs interacting on a complete (all-to-all) graph. If at a time instant (T,J), the phases are contained in an arc of length $d < \pi$, $l \in (0,1]$, and the PRC is given by (3.11), then the network (asymptotically) converges to the set $\mathcal{S}$.*

Proof. Label the oscillators based on their phases at (T,J) in increasing order, i.e., x_1 being the oscillator with the smallest phase. Define a vector of distance functions as:

$$V := [v_{1,2}, v_{2,3}, \ldots, v_{N-1,N}, v_{N,1}]^T \in [0,\pi]^N \tag{3.30}$$

and note that $V = 0$ if and only if $x \in \mathcal{S}$. We will refer to the component $q \in \{1, 2, \ldots, N\}$ of V as V_q. These components are continuous functions and positive definite with respect to $\mathcal{S}$. It is clear, since the oscillators have identical natural frequencies, that V_q remains unchanged during flows. Hence, the discrete-time dynamics entirely determine the synchronization properties. Since the oscillators are contained in an arc of length $d < \pi$, we have that $d = \max_q V_q$ (*cf.* Figure 3.3), and $x \in \mathcal{S} \Leftrightarrow \max_q V_q = 0$. Define $I := \arg\max_q V_q(x(T, J))$ and note that if I is not unique, the oscillators must be clustered at 2 points and I is a set containing 2 elements. In the following we assume that I is unique; however, the arguments still hold when I is not unique.

Consider $x \in \mathcal{D}$. We analyze the change in $v_{i,k}$ when oscillator i jumps. Suppose that i is going to fire, denote the time as (t, j) and the state as $x(t, j)$. We drop the time indices t and j to facilitate the notation; however, the reader should be aware that the time domain is a hybrid one and that $V^+(x)$ means $V(x(t, j+1))$. We have that $x_i = 2\pi$ and $x_k \in [0, 2\pi]$, then:

$$v_{i,k}(x) = \min\{2\pi - x_k, x_k\} \tag{3.31}$$

$$v_{i,k}^+(x) = \begin{cases} x_k(1-l), & \text{if } x_k \in [0, \pi] \\ (2\pi - x_k)(1-l), & \text{if } x_k \in [\pi, 2\pi] \end{cases} \tag{3.32}$$

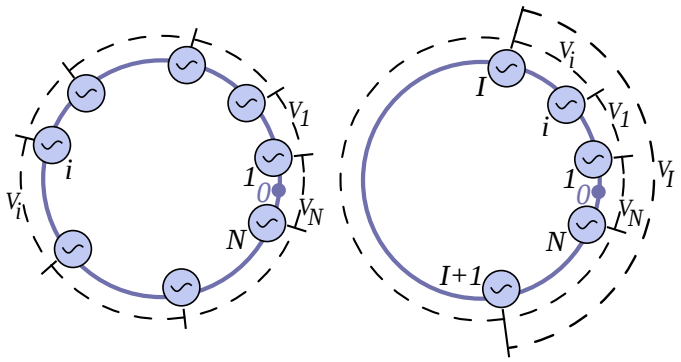


Figure 3.3: Labeling in increasing order and distance functions. When the oscillators are contained in an arc of length $d < \pi$ (on the right), $d = \max V_q$. When the oscillators are contained in an arc of length $d \geq \pi$ (on the left), $d = 2\pi - \max V_q$.

therefore, $v_{i,k}(x) > v^+_{i,k}(x) = (1-l)v_{i,k}(x)$ holds no matter the value of x_k before x_i jumps. Since the oscillators are contained in an arc of length $d < \pi$, we have that for every oscillator i at any time instant, $V_I = v_{i,I} + v_{i,I+1}$ (*cf.* Figure 3.3). Then, after i jumps we have $V_I^+ = v^+_{i,I} + v^+_{i,I+1} = (1-l)(v_{i,I} + v_{i,I+1}) < V_I$, then the length of the arc is reduced after every jump. Since, from Proposition 2.1, jumps are persistent, the network asymptotically synchronizes. □

Remark 3.3 *Lyapunov functions in the spirit of (3.27) have been used in consensus theory [19] and Kuramoto oscillators [25].*

Theorem 3.3 guarantees that the network synchronizes whenever the oscillators are contained in an arc of length less than π. When the oscillators are distributed in the whole set $\mathcal{C}$, the network can still synchronize if the coupling strength is strong enough.

Corollary 3.2 *If $l > 0.5$, the network synchronizes from every initial condition $x(0,0) \in \mathcal{C}$.*

Proof. Consider $x_i(0,0) \in [0, 2\pi]$ and suppose an oscillator i has just fired the first pulse. Since $l > 0.5$, the phase of every oscillator k is such that $x_k \in [0, 0.5\pi) \cup (1.5\pi, 2\pi]$. Hence, the oscillators are contained in an arc of length $d < \pi$. Invoking Theorem 3.3 completes the proof. □

Remark 3.4 *The sufficient condition for global synchronization $l > 0.5$ can be conservative.*

Remark 3.5 *Corollary 3.2 implies that the PRC (3.11) improves the PRC in [62], in the sense that (3.11) allows global synchronization in the all-to-all case.*

3.2.2 The (strongly) Connected Graph Case

In this section we relax the assumption on the graph stated previously by only requiring the graph to be strongly connected in the unidirectional case, or connected in the bidirectional case.

Theorem 3.4 *Consider a network of PCOs interacting on a strongly connected graph $\mathcal{R}$. If at a time instant (T, J), the phases are contained in an arc of length $d < \pi$, $l \in (0, 1]$, and the PRC is given by (3.11), then the network (asymptotically) converges to the set $\mathcal{S}$.*

Proof. The proof follows the same lines as the proof of Theorem 3.3. Label the oscillators based on the phases at (T, J) in increasing order. Define a vector of distance functions as $V := [v_{1,2}, v_{2,3}, \ldots, v_{N-1,N}, v_{N,1}]^T \in [0, \pi]^N$, and define $I := \arg\max_q V_q(x(T, J))$. Consider a complete firing round. Note that oscillator N will fire first (*cf.* Figure 3.3), yet we do not know the next oscillator to fire since the phase ordering is not invariant to jumps in the non all-to-all case. Since the oscillators are contained in an arc of length $d < \pi$, we have that for every oscillator i at any time instant, $V_I = v_{i,I} + v_{i,I+1}$. Following the proof of Theorem 3.3, we know that when i fires, if $I \in \mathcal{N}^{i+}$ then $V_I^+ = v_{i,I}^+ + v_{i,I+1}^+ \leq (1-l)(v_{i,I}) + v_{i,I+1} < V_I$, and similarly if $I + 1 \in \mathcal{N}^{i+}$ then $V_I^+ = v_{i,I}^+ + v_{i,I+1}^+ \leq v_{i,I} + (1-l)(v_{i,I+1}) < V_I$. Since the graph is strongly connected we have that $I \in \cup_{i \in \mathcal{V}} \mathcal{N}^{i+}$ and $I + 1 \in \cup_{i \in \mathcal{V}} \mathcal{N}^{i+}$ hold. Therefore, after a complete round of firings $V_I^{+N} < V_I$; where, with abuse of notation, V_I^{+N} refers to the value of V_I after N firings. Since from Proposition 2.1 jumps are persistent, the network synchronizes asymptotically. □

Remark 3.6 *A similar result to Theorem 3.4 was proven in [101] when there is a dead zone r in the PRC, if $\max_{i,j \in \mathcal{V}} |x_i(0,0) - x_j(0,0)| < \Lambda \in (0, \pi]$ and $r \leq 2\pi - \Lambda$ hold.*

Remark 3.7 *The proof of Theorem 3.4 assumes that every oscillator fires once after some ordinary time interval. This is always true if the initial phases are contained in an arc of length $d < \pi$.*

Remark 3.8 *The case of a network of PCOs interacting on a connected bidirectional graph $\bar{\mathcal{R}}$ is included in the cases covered by Theorem 3.4.*

Theorem 3.4 is the analog of Theorem 3.3 for strongly connected graphs. Unfortunately, it is not possible to obtain an analog of Corollary 3.2 to claim global synchronization. We illustrate this fact with an example.

Example 2: Consider the star-like PCO network depicted in Fig. 3.2(b), note that the interaction graph is strongly rooted. Let x_1 be the phase of the root oscillator (in the middle) and $x_i, i \in \{2, \dots, N\}$ the phase of the peripheral oscillators. We will show that for any $l \in (0, 1]$ we can find a star-like network (with l-dependent number of oscillators N) and a set of initial conditions such that the network does not synchronize. To this end, first note that it is possible for the root to never fire, which requires $x_1(t, j) < \pi, \forall (t, j) \in \text{dom } x$ (since if $x_1 > \pi$, we cannot prevent x_1 reaching 2π and firing). Suppose $x_i(0, 0) = \frac{2\pi}{N-1}(i - 2)$ for $i \neq 1$ and $x_1(0, 0) \in [0, \pi - \frac{2\pi}{N-1})$. Then the phase x_1, ignoring the ordinary time index, satisfies:

$$x_1(j + 1) = \left(x_1(j) + \frac{2\pi}{N - 1} \right) (1 - l) \tag{3.33}$$

it is clear that the (exponentially stable) equilibrium point is given by $x_1 = \frac{2\pi(1-l)}{(N-1)l}$. Imposing $\frac{2\pi(1-l)}{(N-1)l} + \frac{2\pi}{N-1} < \pi$, gives $N > \left[\frac{l+2}{l}\right]$. Then, for any $l \in (0, 1]$ we can find an N such that we can construct the no-synchronization example with $N \to \infty$

as $l \to 0$. For example, for the network in Fig. 3.2(b) if $l > 0.5$ and $x_i(0,0) = \frac{2\pi}{N-1}(i-2)$ for $i \neq 1$, the network never synchronizes. □

Example 2 provides a no-global-synchronization example, where the problem is that the root oscillator never fires. However, if we restrict the family of graphs, a global synchronization result can be stated.

Corollary 3.3 *Consider a network of PCOs interacting on a strongly connected graph $\mathcal{R}$. If the communication graph $\mathcal{R}$ is such that:*

1. *it is strongly rooted, i.e., there exists $i^* \in \mathcal{V}$ such that $|\mathcal{N}^{i^*+}| = N - 1$*
2. *given $l > 0.5$, for every solution x there exists a positive $T_x < \infty$ such that at $(T_x, j) \in$ dom x, for some j, $x_{i^*} = 2\pi$*

then the network synchronizes from every initial condition $x(0,0) \in \mathcal{C}$.

Proof. Consider the state of the network, x, at $(T_x, j) \in$ dom x. x_{i^*} will fire and since $l > 0.5$, at $(T_x, j+1) \in$ dom x we have $x_i \in [0, 0.5\pi) \cup (1.5\pi, 2\pi]$, i.e., the oscillators are contained in an arc of length $d < \pi$. Invoking Theorem 3.4 yields synchronization. Hence, the network synchronizes from every initial condition $x(0,0) \in \mathcal{C}$, since from Proposition 2.1 there exist at least one solution from every initial condition. □

Corollary 3.3 gives a global synchronization result for strongly rooted graphs when the root oscillator always fires. Unfortunately, condition 2) is not straightforward to check from the graph topology. A less restrictive result can be given, if in turn, the graph is bidirectional.

Theorem 3.5 *Consider a network of PCOs interacting on a connected graph $\bar{\mathcal{R}}$. If the communication graph $\bar{\mathcal{R}}$ is such that:*

1. *for every solution x there exists a positive $T_x < \infty$ such that for all $t \geq 0$ every oscillator fires at least once in an ordinary time interval $[t, t + T_x]$*

and $l = 1$, then the network synchronizes from every initial condition $x(0, 0) \in \mathcal{C}$.

Proof. First note that from Proposition 2.1 every solution is complete, hence condition 1) can hold for all $t \geq 0$. In particular for $t = 0$. The main idea of the proof is to construct and monitor the set of unfired oscillators to show that this set being empty implies synchronization. Label the oscillators in firing order and suppose for simplicity and w.l.o.g. that $x_1(0, 0) = 2\pi$. Now let $\chi_{[0,\pi]}(0) := \{i \in \mathcal{V} : x_i(0, 0) \in [0, \pi]\}$ and $\chi_{(\pi,2\pi]}(0) := \{i \in \mathcal{V} : x_i(0, 0) \in (\pi, 2\pi]\}$. After 1 fires, since $l = 1$, it will create a firing storm as follows. Every $i \in \mathcal{N}^{1+} \cap \chi_{(\pi,2\pi]}$ will get a pulse, fire, and it will trigger firings on its child paths (neighbors, neighbors of neighbors, etc.). If the firing storm involves every oscillator, synchronization is achieved. If the firing storm involves a subset of the oscillators, then it must be the

case that the storm finished due to oscillators from the set $\chi_{[0,\pi]}$ getting pulses but not firing. Suppose that firing storms involve a subset of the oscillators and the system has evolved to the time instant (t_1, j_1), where $x_1(t_1, j_1) = \pi$, hence, several oscillators might have fired. Consider now the following sets: $\chi_F := \{i \in \mathcal{V} : \text{fired}\}$, $\chi_1 := \{i \in \mathcal{V} : \text{got pulse, unfired}\}$, and $\chi_U := \{i \in \mathcal{V} : \text{no pulse, unfired}\}$, and note that $\chi_{(\pi,2\pi]}(0) \subseteq \chi_F$, $\chi_U \subseteq \chi_{[0,\pi]}(0)$, and $\chi_1 \subseteq \chi_{[0,\pi]}(0)$, and also that $x_i(t_1, j_1) \in [0, \pi]$, $\forall i \in \chi_F \cup \chi_1$, and $x_i(t_1, j_1) \in (\pi, 2\pi]$, $\forall i \in \chi_U$, i.e., $\chi_{[0,\pi]}(t_1) = \chi_F \cup \chi_1$ and $\chi_{(\pi,2\pi]}(t_1) = \chi_U$; hence, every oscillator in χ_U will fire. Note that, by construction, $\mathcal{N}^{i+} \cap \chi_F = \emptyset$ for all $i \in \chi_U$, but since $\bar{\mathcal{R}}$ is connected $\mathcal{N}^{i+} \cap \chi_1 \neq \emptyset$ hold for at least one $i \in \chi_U$. Now, let the system evolve to a time instant (t_2, j_2) such that the last oscillator in χ_U has just fired. If firing storms are not complete, then oscillators in χ_1 got pulses while in $[0, \pi]$. We can update the set of unfired oscillators at (t_2, j_2) as $\chi_2 \subseteq \chi_1$, the set of oscillators in $[0, \pi]$ as $\chi_{[0,\pi]}(t_2) \supseteq \chi_U \cup \chi_2$ (might also include elements from χ_F and $\chi_1 \setminus \chi_2$), and the set of oscillators in $(\pi, 2\pi]$ as $\chi_{(\pi,2\pi]}(t_2) \subseteq \chi_F \cup (\chi_1 \setminus \chi_2)$. If $\chi_{(\pi,2\pi]}(t_2)$ is empty, then the system will evolve towards 2π and a complete firing storm will occur. For the moment, assume that $\chi_{(\pi,2\pi]}(t_2)$ is nonempty and note that, $\mathcal{N}^{i+} \cap \chi_{[0,\pi]}(t_2) \setminus \chi_2 = \emptyset$ for all $i \in \chi_{(\pi,2\pi]}(t_2)$ and due to $\bar{\mathcal{R}}$ being connected $\mathcal{N}^{i+} \cap \chi_2 \neq \emptyset$ hold for at least one $i \in \chi_{(\pi,2\pi]}(t_2)$. Letting the system evolve to a time instant (t_3, j_3) such that the last oscillator in $\chi_{(\pi,2\pi]}(t_2)$ has just fired, we see that the only option for firing storms not to

be complete is that oscillators in χ_2 got pulses while in $[0, \pi]$, and hence some of them have not fired. We can update the set of unfired oscillators at (t_3, j_3) as $\chi_3 \subseteq \chi_2 \subseteq \chi_1$. Iterating this argument, as the system evolves, the only option for having incomplete firing storms is to have a set of oscillators that never fire. Since, condition 1) requires every oscillator to have fired at (T_x, J), there exist a $t_n < T_x$ such that at (t_n, j_n) the set $\chi_{(\pi,2\pi]}(t_n)$ is empty and every oscillator i in the set χ_n from the collection $\chi_1, \chi_2, \ldots, \chi_n$ will fire and hence, after a complete firing storm, the network synchronizes. □

Remark 3.9 *Note that when the network synchronizes at $t_s > 0$, every oscillator fires exactly once in an ordinary time interval $[t, t + 2\pi]$ for every $t \geq t_s$.*

Condition 1) in Theorem 3.5 suggests that suppression of firing is the main obstacle for synchronization. Theorem 3.5 imposes a condition of periodic firing that is not straightforward to translate into conditions on the edges of $\bar{\mathcal{R}}$. To give a feeling of the types of networks included in Theorem 3.5, Fig. 3.4 shows two examples. Fig. 3.4(a) shows a connected network that do not fulfill conditions in Theorem 3.5. To see this, note that if the oscillators in the top row are uniformly distributed in the interval $[0, \pi)$, the oscillators in the middle row are initialized at 0, and the oscillators in the bottom row are initialized uniformly distributed in the interval $[\pi, 2\pi]$, the middle row never fires and the network never synchronizes. On the other hand, Fig. 3.4(b) shows a connected network that fulfills conditions

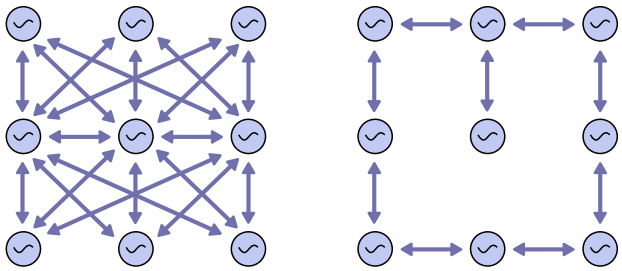


Figure 3.4: Connected bidirectional networks. (a): A network for which conditions in Theorem 3.5 do not hold since initial conditions can be chosen such that the middle row never fires; (b) A network for which conditions in Theorem 3.5 hold.

in Theorem 3.5 since every oscillator will fire periodically no matter the initial condition. It is worth noting that, as shown by the networks in Fig. 3.4, a higher number of edges does not imply higher likelihood of global synchronization.

Unfortunately, an analog result to Theorem 3.5 cannot be formulated for unidirectional graphs. We illustrate this fact with an example.

Example 3: Consider the 5-oscillator unidirectional cycle depicted in Fig. 3.2(c). Assume $l = 1$, $x_i(0,0) = \frac{2\pi}{N}i$, and $\mathcal{E}_{\mathcal{R}} = \{(1,5),(2,1),(3,2),(4,3),(5,4)\}$. Oscillator 5 fires first and the state after the firing is $x = [0, \frac{4\pi}{5}, \frac{6\pi}{5}, \frac{8\pi}{5}, 0]$. Oscillator 4 fires next and the state after the firing is $x = [\frac{2\pi}{5}, \frac{6\pi}{5}, \frac{8\pi}{5}, 0, 0]$; next, oscillator 3 fires, and the state after the firing is $x = [\frac{4\pi}{5}, \frac{8\pi}{5}, 0, 0, \frac{2\pi}{5}]$. Oscillator 2 fires next and the state after the firing is $x = [\frac{6\pi}{5}, 0, 0, \frac{2\pi}{5}, \frac{4\pi}{5}]$. Iterating, it is clear that the network never synchronizes, despite that every oscillator fires periodically and $l = 1$. □

3.2.3 Synchronization on Cycle Graphs

In this section we analyze the synchronization properties of PCO networks interacting on cycle graphs, which are a subset of the more general class of strongly connected graphs. We consider as a cycle graph the graph $\mathcal{R} = \{\mathcal{V}, \mathcal{E}_\mathcal{R}, \mathcal{A}_\mathcal{R}\}$, for which w.l.o.g. the edge set is given by $\mathcal{E}_\mathcal{R} = (1, N) \cup \bigcup_{i=1}^{N-1}(i+1, i)$, i.e., node $i+1$ can sense the firing of node i. Similarly, we define $\bar{\mathcal{E}}_\mathcal{R}$ as the bidirectional, or undirected, version of $\mathcal{E}_\mathcal{R}$, i.e., if $(i, j) \in \mathcal{E}_\mathcal{R}$ then (i, j) and $(j, i) \in \bar{\mathcal{E}}_\mathcal{R}$.

Although the conditions derived in the previous section still hold for cycle graphs, in this section we exploit the particular structure of the graph to obtain the exact value of the critical coupling strength l^* that enables global synchronization. Note that even though "easy" initial conditions can synchronize under weaker conditions, the following results give the weakest conditions for global synchronization.

Consider the following family of functions representing the distance to the synchronization set $\mathcal{S}$:

$$v_{i,i+1}(x) = \min\left(|x_i - x_{i+1}|, 2\pi - |x_i - x_{i+1}|\right) \tag{3.34}$$

note that $v_{i,i+1}(x)$ represents the length of the shortest segment joining oscillators i and $i+1$. Define the vector of distance functions as:

$$V := [v_{1,2}, v_{2,3}, \ldots, v_{N-1,N}, v_{N,1}]^T \in [0,\pi]^N \tag{3.35}$$

and the length of the cycle as $\mathbf{1}^T V$, where $\mathbf{1}$ is the N-dimensional column vector of all ones. We will refer to the component $q \in \{1, 2, \ldots, N\}$ of V as V_q. These components are continuous functions with respect to x, and positive definite with respect to $\mathcal{S}$. It is clear, since the oscillators have identical natural frequencies, that V_q remains unchanged during flows. Hence, the discrete-time dynamics (jumps) entirely determine the synchronization properties of the system. As in the previous sections, we will analyze the convergence properties of the underlying discrete-time system to prove synchronization.

Before stating the synchronization results, we need to introduce the concept of refractory period and a technical lemma that can be easily derived from Theorem 1 in [101] and Theorem 3.4.

Definition 3.3 (Refractory period) *A refractory period is an interval* $[0, r] \subseteq [0, 2\pi]$*, where* r *is the length of the refractory period, such that if the phase of an oscillator is inside the interval, it does not react to an incoming pulse, i.e., a refractory period of length* r *corresponds to a dead zone in the PRC in the interval*

$[0, r]$ *[22]. When a refractory period is introduced, the PRC* Q *is modified as* $Q(x) = \{0\}$ *for* $x \in [0, r)$ *and as* $Q(x) = \{0\} \cup \lim_{x \to r^+} Q(x)$ *for* $x = r$*. (cf. Figure 3.5).*

Lemma 3.1 *Consider a network of PCOs interacting on a cycle graph, either* $\mathcal{R}$ *or* $\bar{\mathcal{R}}$ *. If the initial phases are such that*

$$\max_{i,k \in \mathcal{V}} |x_i(0,0) - x_k(0,0)| < \pi,$$

$l \in (0, 1]$*, and the PRC is given by (3.11), then the network converges asymptotically to the set* $\mathcal{S}$ *even if there exists a refractory period in the PRC of length* $r \leq \pi$*.*

The following Theorems are the main results of this section and provide necessary and sufficient conditions for global synchronization of PCOs interacting on cycle graphs.

Theorem 3.6 *Consider the network of PCOs with dynamics* $\mathcal{H}$ *interacting on the bidirectional cycle graph* $\bar{\mathcal{R}}$*, and with PRC given by (3.11). The network synchronizes from every initial condition if and only if the coupling strength* l *is larger than the critical coupling* l^**, which is given by:*

$$l^* = \frac{N}{2} - \frac{\sqrt{N^2 - 4(N-2)}}{2} \tag{3.36}$$

A similar condition can be derived for the unidirectional graph $\mathcal{R}$.

Theorem 3.7 *Consider the network of PCOs with dynamics $\mathcal{H}$ interacting on the unidirectional cycle graph $\mathcal{R}$, and with PRC given by (3.11). Moreover, consider that there exists a refractory period of length $r = \pi$ in the PRC of 1 oscillator. The network synchronizes from every initial condition if and only if the coupling strength l is larger than the critical coupling l^*, which is given by:*

$$l^* = \frac{N-2}{N-1} \tag{3.37}$$

To prove Theorems 3.6 and 3.7 we rely on the following Lemma.

Lemma 3.2 *Consider the distance vector V defined in (3.30), the length of the cycle defined as $\mathbf{1}^T V$, and the PRC (3.11). At any time instant $(\bar{t}, \bar{j})$, let $i^* \in \mathcal{V}$ be the index of the oscillator with the largest phase and $i_* \in \mathcal{V}$ the index of the oscillator with the smallest phase. Define $\mathcal{U}_1 := \{x \in \mathcal{C} : x_i \geq x_{i+1} \forall i \in \mathcal{V} \setminus \{i_*\}\} \cap \{x \in \mathcal{C} : \mathbf{1}^T V = 2\pi\}$, $\mathcal{U}_2 := \{x \in \mathcal{C} : x_i \leq x_{i+1} \forall i \in \mathcal{V} \setminus \{i^*\}\} \cap \{x \in \mathcal{C} : \mathbf{1}^T V = 2\pi\}$, and $\mathcal{U} := \mathcal{U}_1 \cup \mathcal{U}_2$. The following claims hold:*

(a) If $\mathbf{1}^T V < 2\pi$, then $\exists\ i \in \mathcal{V} \setminus \min\{i^*, i_*\}$ such that $|x_i - x_{i+1}| > \pi$, or $|x_{i^*} - x_{i_*}| < \pi$

(b) If $\mathbf{1}^T V > 2\pi$*, then* $\exists\ i \in \mathcal{V}$ *such that when* $x_i = 2\pi$ *we have that* $x_{i+2} \in [0, x_{i+1})$ *and* $x_{i+1} \leq \pi$*, or* $x_{i+2} \in (x_{i+1}, 2\pi]$ *and* $x_{i+1} \geq \pi$*, or* $|x_{i+2} - x_{i+1}| > \pi$*; hence* $\mathbf{1}^T V$ *decreases after* i *jumps.*

(c) If $\mathbf{1}^T V = 2\pi$ *and* $x \notin \mathcal{U}$*, then there exists* $i \in \mathcal{V}$ *such that* $\mathbf{1}^T V$ *decreases after* i *jumps.*

(d) If $x \in \mathcal{U}$*, then* $|x_{i^*} - x_{i_*}| \geq \pi$ *and* $|x_i - x_{i+1}| \leq \pi$, $\forall i \in \mathcal{V} \setminus \{i^*, i_*\}$.

Proof. Define $\bar{i} := \min\{i^*, i_*\}$. To prove statement (a) note that $|x_i - x_{i+1}| \leq \pi \Rightarrow V_i = |x_i - x_{i+1}|$. If we sum over the V_is, the minimum value is reached when the segments are disjoint, i.e., phases are ordered either clock-wise, or counter clock-wise, and in this case $\sum_{i \in \mathcal{V} \setminus \{\bar{i}\}} V_i = |x_{i^*} - x_{i_*}|$. Now we proceed to prove by contraposition. Suppose $|x_i - x_{i+1}| \leq \pi\, \forall i \in \mathcal{V} \setminus \{i^*, i_*\}$ and $|x_{i^*} - x_{i_*}| \geq \pi$. Then, the minimum length of the cycle is equal to $\sum_{i \in \mathcal{V} \setminus \{\bar{i}\}} V_i + \min\left(|x_{i^*} - x_{i_*}|, 2\pi - |x_{i^*} - x_{i_*}|\right)$, which corresponds to the component measuring the length between x_{i^*} and x_{i_*} (note that for the length to be minimal, i^* and i_* must be neighbors). Then we have $\mathbf{1}^T V \geq \sum_{i \in \mathcal{V} \setminus \{\bar{i}\}} V_i + 2\pi - |x_{i^*} - x_{i_*}| = 2\pi$. Hence, (a) holds.

Regarding (b), note that for the length to be larger than 2π segments cannot be disjoint since from the proof of (a) we know that disjoint segments can add up to 2π, the length of the domain. Then, there must be the case that at least 2 segments, described by the components of V, intersect. Considering $x_i = 2\pi$,

the conditions $x_{i+2} \in [0, x_{i+1})$ and $x_{i+1} \leq \pi$, or $x_{i+2} \in (x_{i+1}, 2\pi]$ and $x_{i+1} \geq \pi$, or $|x_{i+2} - x_{i+1}| > \pi$ ensure that at least 2 segments affected by i intersect. Moreover, these conditions imply that the length of the cycle, $\mathbf{1}^T V$, will decrease after i jumps.

In the same line, when $\mathbf{1}^T V = 2\pi$ and $x \notin \mathcal{U}$ segments are not disjoint and then there exists i such that when $x_i = 2\pi$, we have $x_{i+2} \in [0, x_{i+1})$ and $x_{i+1} \leq \pi$, or $x_{i+2} \in (x_{i+1}, 2\pi]$ and $x_{i+1} \geq \pi$, or $|x_{i+2} - x_{i+1}| > \pi$ holds, implying that the length of the cycle, $\mathbf{1}^T V$, will decrease after i jumps. Hence, (c) holds.

Statement (d) follows by noting that the phase ordering implies that i^* and i_* are neighbors. Moreover, from $\mathbf{1}^T V = 2\pi$ we have $\sum_{i \in \mathcal{V} \setminus \{\bar{i}\}} V_i = |x_{i^*} - x_{i_*}|$ or $\sum_{i \in \mathcal{V} \setminus \{\bar{i}\}} V_i = 2\pi - |x_{i^*} - x_{i_*}|$ depending on whether $|x_{i^*} - x_{i_*}| \geq \pi$ or $|x_{i^*} - x_{i_*}| < \pi$ holds. We will proceed by contradiction. Suppose the latter is true and then $|x_i - x_{i+1}| < \pi$ holds for every oscillator $i \in \mathcal{V}$; furthermore, the phase ordering implies that segments are disjoint and then $\sum_{i \in \mathcal{V} \setminus \{\bar{i}\}} V_i = |x_{i^*} - x_{i_*}|$, which contradicts $\mathbf{1}^T V = 2\pi$. Hence $|x_{i^*} - x_{i_*}| \geq \pi$ must hold. Now if $|x_{i^*} - x_{i_*}| \geq \pi$ holds, either $|x_i - x_{i+1}| \leq \pi$ holds for every oscillator $i \in \mathcal{V} \setminus \{\bar{i}\}$ or $|x_i - x_{i+1}| > \pi$ holds for only one oscillator i (due to the phase ordering). Suppose the latter is true (note that for this to be feasible $|x_{i^*} - x_{i_*}| > \pi$ must hold), then we have $\sum_{i \in \mathcal{V} \setminus \{\bar{i}\}} V_i = 2\pi + |x_{i^*} - x_{i_*}| - 2|x_i - x_{i+1}| < |x_{i^*} - x_{i_*}|$, which again contradicts

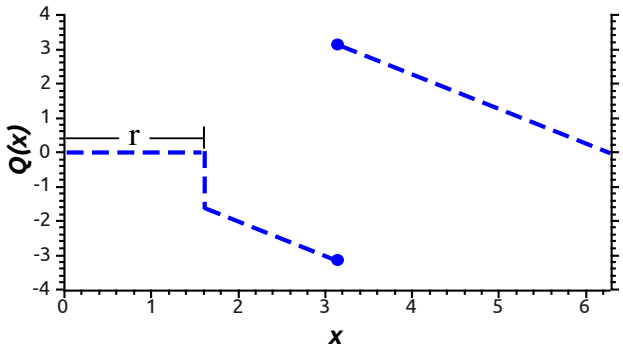


Figure 3.5: Effect of the addition of a refractory period on the PRC. The PRC Q is modified as $Q(x) = \{0\}$ for all $x \in [0, r]$, where r is the length of the refractory period.

$\mathbf{1}^T V = 2\pi$. Hence, $|x_i - x_{i+1}| \leq \pi$ must hold for every oscillator $i \in \mathcal{V} \setminus \{i^*, i_*\}$ and the Lemma is proven. □

Remark 3.10 *Note that Lemma 3.2(a) implies that if* $\mathbf{1}^T V < 2\pi$*, then conditions of Lemma 3.1 hold up to a rigid rotation of the oscillators. Hence, when* $\mathbf{1}^T V < 2\pi$ *the network always synchronizes. Moreover, conditions in Lemma 3.1 and Lemma 3.2(a) imply that the oscillators are contained in a semicircle, a problem equivalent to a consensus problem in* $\mathbb{R}^N$ *[60].*

Remark 3.11 *Statement* (b) *of Lemma 3.2 means that when* $\mathbf{1}^T V > 2\pi$*, the length will eventually decrease. Regarding global synchronization, initial conditions for which* $\mathbf{1}^T V > 2\pi$ *do not represent a problem since in these cases the length will decrease. In fact, we will show that the only problematic situation is when* $x(0, 0) \in \mathcal{U}$*.*

Now we proceed to prove Theorem 3.6.

Proof. To prove sufficiency, the strategy is to show that every solution is such that eventually $\mathbf{1}^T V < 2\pi$ and hence Lemma 3.1 yields synchronization of the network. Consider an arbitrary initial condition $x(0,0) \in \mathcal{C}$, we have four possible scenarios:

i) $x(0,0) \in \mathcal{C} : \mathbf{1}^T V < 2\pi$

In this case, directly applying Lemma 3.1 guarantees synchronization.

ii) $x(0,0) \in \mathcal{C} : \mathbf{1}^T V > 2\pi$

Lemma 3.2(b) guarantees that $\mathbf{1}^T V$ will decrease while $\mathbf{1}^T V > 2\pi$, then there exists a time instant (t_{ii}, j_{ii}) such that either $\mathbf{1}^T V = 2\pi, x(t_{ii}, j_{ii}) \notin \mathcal{U}$ or $x(t_{ii}, j_{ii}) \in \mathcal{U}$. At this point we can reinitialize the system in case *iii)* or *iv)*.

iii) $x(0,0) \in \mathcal{C} : \mathbf{1}^T V = 2\pi, x(0,0) \notin \mathcal{U}$

Lemma 3.2(c) ensures that the length will decrease and then there exists a time instant (t_{iii}, j_{iii}) at which $\mathbf{1}^T V < 2\pi$. At this point we can reinitialize the system in case *i)* and invoking Lemma 3.1 gives synchronization.

iv) $x(0,0) \in \mathcal{U}$

In this case, the situation is more complicated. To show that the system jumps outside $\mathcal{U}$, we analyze the change in V when an oscillator i jumps. Consider $x \in \mathcal{D}$, which is the union of the jump conditions for all x_i, suppose w.l.o.g. that node i is about to fire, denote the time as (t, j) and the state as $x(t, j)$. We drop the time indices t and j to facilitate the notation; however, the reader should be

aware that the time domain is a hybrid one, that $V^+(x)$ means $V(x(t, j+1))$, and $V_0 = V(x(0,0))$. We have that $x_i = 2\pi$ and $x_{i+1} \in [0, 2\pi]$, then:

$$V_i(x) = \min\{2\pi - x_{i+1}, x_{i+1}\} \tag{3.38}$$

$$V_i^+(x) = x_{i+1}(1-l) \text{ or } (2\pi - x_{i+1})(1-l) \tag{3.39}$$

depending on whether $x_{i+1} \in [0, \pi]$ or $x_{i+1} \in [\pi, 2\pi]$. Then, $V_i(x) > V_i^+(x) = (1-l)V_i(x)$ holds for any value of x_{i+1} before x_i jumps. Note that since the previous analysis is valid for all i we have $V_{i-1}(x) > V_{i-1}^+(x) = (1-l)V_{i-1}(x)$. Next we analyze the change in V_{i+1}. In this case we have:

$$V_{i+1}(x) = \min\{|x_{i+1} - x_{i+2}|, 2\pi - |x_{i+1} - x_{i+2}|\} \tag{3.40}$$

$$V_{i+1}^+(x) = \min\{|x_{i+1} - x_{i+2} + lQ(x_{i+1})|, 2\pi - |x_{i+1} - x_{i+2} + lQ(x_{i+1})|\} \tag{3.41}$$

Since $x(t,j) \in \mathcal{U}$, the phase ordering (either $x_i \geq x_{i+1}$ or $x_i \leq x_{i+1}$) and $|x_i - x_{i+1}| \leq \pi$ from Lemma 3.2(d) ensure that $V_{i+1}^+(x) = V_{i+1}(x) + lV_i(x)$ and $V_{i-2}^+(x) = V_{i-2}(x) + lV_{i-1}(x)$ hold, provided $V_{i+1}(x) + lV_i(x) < \pi$ and $V_{i-2}(x) + lV_{i-1}(x) < \pi$ (note that if the previous conditions do not hold, the length decreases and since $\mathbf{1}^T V < 2\pi$, the network synchronizes). The other components of V remain unchanged when i jumps. We can then write the change of V after i jumps in

matrix form by using the following transition matrices

$$\bar{C}_i = \begin{array}{c} \\ \\ \\ \\ \\ i^{th} \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \overset{\qquad\qquad\qquad\qquad i^{th}}{} \\ \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & \vdots & \vdots & \vdots & \vdots & \cdots & 0 \\ 0 & 0 & \ddots & l & \vdots & \vdots & \vdots & \cdots & 0 \\ 0 & 0 & \cdots & (1-l) & 0 & \cdots & \vdots & \cdots & 0 \\ \vdots & \vdots & \cdots & 0 & (1-l) & 0 & \vdots & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots & l & 1 & \vdots & \cdots & 0 \\ \vdots & \vdots & \cdots & 0 & \vdots & 0 & \ddots & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \end{array} \tag{3.42}$$

Then, when $V_{i+1}(x) + lV_i(x) < \pi$ and $V_{i-2}(x) + lV_{i-1}(x) < \pi$ hold, the value of V after i jumps is given by $V^+ = \bar{C}_i V$. Note that $\bar{C}_i$ are column stochastic matrices and then when $V^+ = \bar{C}_i V$, $\mathbf{1}^T V^+ = \mathbf{1}^T V$ holds, i.e., the length remains constant and the state remains in $\mathcal{U}$. In the following, we will use an auxiliary system $\tilde{V}^+ = \bar{C}_i \tilde{V}$ with $\tilde{V}_i \in \mathbb{R}$ and $\tilde{V}_0 = V_0$ (note that the elements of $\tilde{V}$ are not restricted to $[0, \pi]$ as the elements of V) to show that if $l > l^*$, the state will jump out of $\mathcal{U}$ and the network will synchronize. It is a well known fact from consensus theory [19] that an infinite product of column stochastic matrices with positive diagonal

entries, as $\bar{C}_i$, converges exponentially to a matrix of the form $\gamma \mathbf{1}^T$, where γ is a column vector such that $\mathbf{1}^T\gamma = 1$ [19]. By exploiting the particular structure of the $\bar{C}_i$ matrices, we can determine exactly the value of the vector γ as follows. Assume the system $\tilde{V}^+ = \bar{C}_i\tilde{V}$ is at the equilibrium $\tilde{V}^* = \gamma\mathbf{1}^T\tilde{V}_0 = 2\pi\gamma$ and w.l.o.g. $x \in \mathcal{U}_1$ and oscillator 1 is about to fire (note that in the bidirectional case, $x \in \mathcal{U}_1$ and $x \in \mathcal{U}_2$ are equivalent in terms of $\tilde{V}$). The phase ordering ensures that the firing sequence will be $1, 2, \ldots, N$ and since the system is at equilibrium, the $\bar{C}_i$ matrices induce a hard rotation on the elements of γ (since the length cannot decrease). Hence, assuming $l \in (0, 1)$, the vector γ must contain $N - 2$ identical elements δ, one element equal to $(1 - l)\delta$ and one element equal to $\frac{\delta}{(1-l)}$. Moreover, we have that

$$(N - 2)\delta + (1 - l)\delta + \frac{\delta}{(1 - l)} = 1 \tag{3.43}$$

holds. Since $l > l^* = \frac{N}{2} - \frac{\sqrt{N^2-4(N-2)}}{2}$, solving for $\frac{\delta}{(1-l)}$ gives $\frac{\delta}{(1-l)} > \frac{1}{2}$. Hence, if $l > l^*$ we have that, at the equilibrium, $\max \tilde{V}_i = \frac{\delta}{(1-l)}2\pi > \pi$. Then, a component of $\tilde{V}$ will converge exponentially fast [19] to a value larger than π, which in the original system, where $V_i \in [0, \pi]$, has to be interpreted as $|x_I - x_{I+1}| > \pi$ for some $I \in \mathcal{V}$ and hence we have $\mathbf{1}^T V < 2\pi$. At this point, we can take this as initial condition for Lemma 3.1. It should be noted that if $l = 1$, γ contains only one non zero entry $\delta = 1$, which ensures synchronization. Hence, the network synchronizes from every initial condition $x(0, 0) \in \mathcal{C}$. The 'only if' part follows easily by contra-

diction. First suppose that the network synchronizes from every initial condition and that $l \leq l^*$. Define the set $\bar{\mathcal{U}}^* := \Big\{ x \in \mathcal{U} : V_{i^*-1} = \frac{\delta}{(1-l)}, V_{i_*} = (1-l)\delta, V_i = \delta \; \forall i \in \mathcal{V} \setminus \{i^* - 1, i_*\}\}$ as the "worst case" set (note that this set contains the equilibrium of the system $\tilde{V}^+ = \bar{C}_i \tilde{V}$) the result follows by using $x(0,0) \in \bar{\mathcal{U}}^*$ as a counterexample. □

The proof of Theorem 3.7 is as follows.

Proof. The proof uses the same arguments as the proof of Theorem 3.6. Cases *i)*, *ii)*, and *iii)* follows the same arguments, yet case *iv)* is different. To show that the system jumps outside $\mathcal{U}$, first consider $x(0,0) \in \mathcal{U}_1$. In this case we have that when i fires, the phase ordering and Lemma 3.2(d) ensure that $x_{i+1} \in [\pi, 2\pi]$ and then the refractory period has no effect. Following the same reasoning as the one

for the proof of Theorem 3.6, the transition matrices are given by

$$C_i = \begin{array}{c} \\ \\ \\ \\ \\ i^{th} \\ \\ \\ \\ \\ \end{array} \overset{\begin{array}{ccccccccc} & & & & & i^{th} & & & \end{array}}{\begin{pmatrix} 1 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & \vdots & \vdots & \vdots & \vdots & \cdots & 0 \\ 0 & 0 & \ddots & 0 & \vdots & \vdots & \vdots & \cdots & 0 \\ 0 & 0 & \cdots & 1 & 0 & \cdots & \vdots & \cdots & 0 \\ \vdots & \vdots & \cdots & 0 & (1-l) & 0 & \vdots & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots & l & 1 & \vdots & \cdots & 0 \\ \vdots & \vdots & \cdots & 0 & \vdots & 0 & \ddots & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix}} \tag{3.44}$$

Note that the transition matrices are different from the bidirectional case due to the unidirectional nature of the graph. However, C_i are also column stochastic matrices and hence their infinite product converges exponentially to a matrix of the form $\gamma \mathbf{1}^T$. We will again consider an auxiliary system $\tilde{V} = C_i \tilde{V}$ to prove that the system jumps outside $\mathcal{U}_1$. Since $x \in \mathcal{U}_1$, the phase ordering ensures that the firing sequence will be $1, 2, \ldots, N$ and for the matrices C_i to induce a hard rotation on $\tilde{V}$ at the equilibrium, assuming $l \in (0, 1)$, the vector γ must contain

$N-1$ identical elements δ and one element equal to $\frac{\delta}{(1-l)}$. Moreover, we have that

$$(N-1)\delta + \frac{\delta}{(1-l)} = 1 \tag{3.45}$$

holds. Since $l > l^* = \frac{N-2}{N-1}$, solving for $\frac{\delta}{(1-l)}$ gives $\frac{\delta}{(1-l)} > \frac{1}{2}$. Hence, if $l > l^*$ we have that, at the equilibrium, $\max \tilde{V}_i = \frac{\delta}{(1-l)} 2\pi > \pi$, Then, a component of $\tilde{V}$ will converge exponentially fast to a value larger than π, which in the original system, where $V_i \in [0, \pi]$, has to be interpreted as $|x_I - x_{I+1}| > \pi$ for some $I \in \mathcal{V}$ and hence we have $\mathbf{1}^T V < 2\pi$. At this point, we can take this as initial condition for Lemma 3.1. It should be noted that if $l = 1$, γ contains only one non zero entry $\delta = 1$, which ensures synchronization. Hence, the network synchronizes from every initial condition $x(0,0) \in \mathcal{U}_1$.

Now consider $x(0,0) \in \mathcal{U}_2$, and w.l.o.g. that N will fire first. Suppose further that there is no refractory period. Note that in this case, the phase ordering ensures that the firing sequence will be $N, N-1, \ldots, 1$. Hence, to ensure a hard rotation at the equilibrium, the vector γ must contain $N-1$ identical elements δ and one element equal to $(1-l)\delta$. Moreover, we have that

$$(N-1)\delta + (1-l)\delta = 1 \tag{3.46}$$

holds. Then, the maximum feasible value for δ is $\frac{1}{N-l}$ and the network cannot synchronize, even if $l = 1$. However, when there is a refractory period in one node the network can synchronize. Recall that, due to the phase ordering, nodes get pulses when their phases are in $[0, \pi)$. Then, if the refractory period is in node i, when node $i-1$ jumps V_{i-1} is not affected; yet when node $i-2$ jumps, node $i-1$ is affected and V_{i-1} is increased by lV_{i-2}. Therefore after one round of firings V_{i-1} will have been increased by lV_{i-2}. Iterating this argument, $|x_{i-1} - x_i| > \pi$ will hold after a finite number of firing rounds, node i will react to node's $i-1$ firing event, and $x_i \in [0, \pi)\ \forall i \in \mathcal{V}$. Invoking again Lemma 3.1 completes the proof.

The 'only if' part follows by contradiction supposing that the network synchronizes from every initial condition and that $l \leq l^*$. Using the "worst case" initial condition $x(0,0) \in \mathcal{U}_1^*$ as counterexample yields a contradiction, where $\mathcal{U}_1^* := \left\{ x \in \mathcal{U}_1 : V_{i^*} = \frac{\delta}{(1-l)}, V_i = \delta\ \forall i \in \mathcal{V} \setminus \{i^*\} \right\}$. □

Remark 3.12 *The beneficial effects of a refractory period on the stability of PCO networks have been mentioned before [43, 101]. In the same sense, Theorem 3.7 states that the introduction of a refractory period enables global synchronization in the unidirectional case. It should be noted, however, that if more than one oscillator is affected by a refractory period, global synchronization cannot be guaranteed. To see this fact, consider the following example. Suppose we have a 3-node unidirectional cycle with interaction given by* $1 \to 2 \to 3 \to 1$ *Consider the initial*

condition $x = [\frac{\pi}{3}, \frac{2\pi}{3}, 2\pi]$ *and suppose further that every oscillator is affected by a refractory period of length* π. *After oscillator 3 jumps, 1 is not affected since it is in the refractory period and the state is given by* $x = [\frac{\pi}{3}, \frac{2\pi}{3}, 0]$. *Let the system evolve until oscillator 2 reaches* 2π. *Then, when 2 jumps, 3 is not affected since it is in the refractory period and the state is given by:* $x = [\frac{2\pi}{3}, 0, \frac{\pi}{3}]$. *Let the system evolve until oscillator 1 reaches* 2π *and note that oscillator 2 will not be affected by the firing since it is in the refractory period. The state after the jump is given by:* $x = [0, \frac{\pi}{3}, \frac{2\pi}{3}]$. *Iterating, it can be seen that the network never synchronizes. Alternatively, suppose we have a N-node bidirectional cycle, where 2 nodes have a refractory period of length* π. *Consider an initial condition given by 2 clusters, one at* π *and the other at* 2π *and suppose* $l = 1$. *It can be derived that the oscillators containing the refractory period will remain* π *apart while the other oscillators will jump back and forth.*

Remark 3.13 *Note that when* $l < l^*$ *the system stays in either* $\bar{\mathcal{U}}^*$ *or* $\mathcal{U}_1^*$, *depending on the structure of the graph. Both* $\bar{\mathcal{U}}^*$ *and* $\mathcal{U}_1^*$ *correspond to a phase-locked solution that leads to a constant inter-firing time. A similar strategy for communication scheduling, where the firing period can be divided into N "slots", one per node, is known as time-division-multiple-access (TDMA). PCO-based TDMA schemes are known to exist when the coupling is repulsive [23].*

3.2.4 The (strongly) Connected Graph Case Revisited

Based on the results obtained for cycle graphs, we can reformulate Theorem 3.5 to stress the necessity of the strongest coupling $l = 1$ to ensure global synchronization in connected graphs.

Theorem 3.8 *Consider a network of PCOs interacting on a connected graph $\bar{\mathcal{R}}$. If the communication graph $\bar{\mathcal{R}}$ is such that:*

1. *for every initial condition $x(0,0) \in \mathcal{C}$ there exists a positive $T < \infty$ such that for all $t \geq 0$ every oscillator fires at least once in an ordinary time interval $[t, t+T]$*

then the network synchronizes from every initial condition $x(0,0) \in \mathcal{C}$ if and only if $l = 1$.

Proof. The sufficiency part, is given in the proof of Theorem 3.5. For the necessity part, we proceed by contradiction using a counter-example. Suppose $\bar{\mathcal{R}}$ is connected, every oscillator fires periodically, the network synchronizes globally, and $l < 1$. From the results in Theorem 3.6, we know that if $\bar{\mathcal{R}}$ is a bidirectional cycle, the network synchronizes globally if and only if $l > \frac{N}{2} - \frac{\sqrt{N^2-4(N-2)}}{2}$. Then, since $l < 1$ we can construct a bidirectional cycle, with l-dependent number of oscillators, such that the network cannot synchronize globally, which contradicts the global synchronization assumption. Hence, the Theorem is proven. □

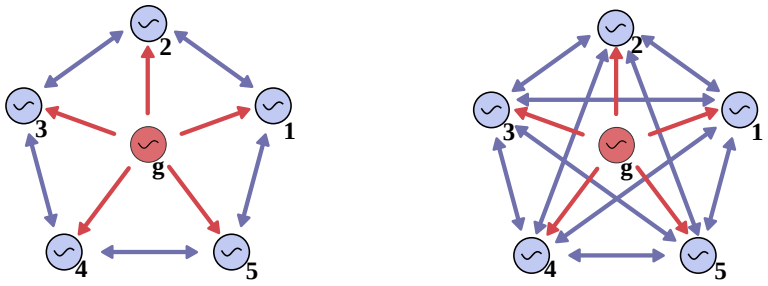


Figure 3.6: Networks with a global cue used in the numerical simulations. (a): Bidirectional 5-nodes cycle network plus an omnipresent global cue (red node in the middle); (b): All-to-all 5-nodes network plus an omnipresent global cue (red node in the middle).

3.3 Numerical Experiments

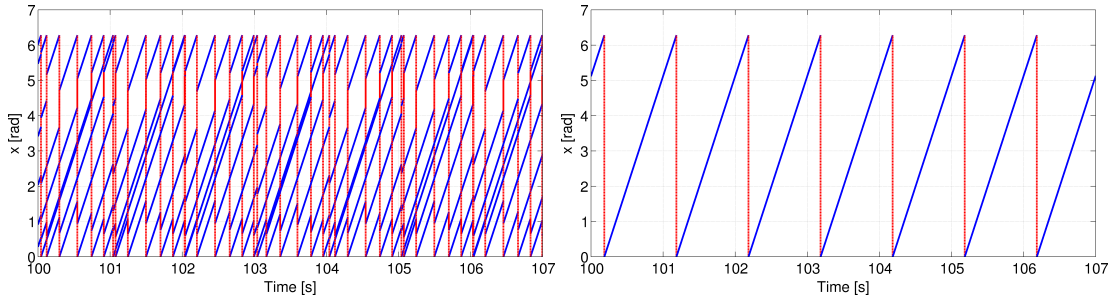
3.3.1 Centralized Networks

To illustrate the applicability and conservativeness of the analytical results previously derived for networks in the presence of a global cue, we use the hybrid systems simulator [82] to simulate the networks shown in Figure 3.6 when the PRC is given by (3.11).

We will examine the identical natural frequencies case in the first place. To this end, natural frequencies were set as $w_i = w = 2\pi$. As first example consider the network in Figure 3.6(a), which consists of a 5-slave-nodes bidirectional cycle plus an omnipresent global cue (in red). We simulate the network from a random initial condition and coupling given by $g = 0.1$ and $l = 0.4$. Figure 3.7(a) shows the simulation results. It can be seen that the network does not synchronize. In fact,

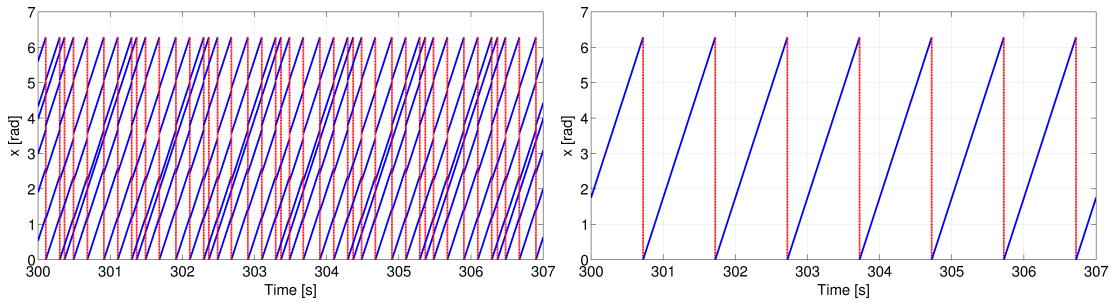
if we recall our previous result for bidirectional cycles in Theorem 3.6, the local coupling $l = 0.4$ is not strong enough to synchronize the cycle network globally. Moreover, the global coupling $g = 0.1$ is not strong enough to synchronize the slave oscillators to the global cue. Hence, the global coupling can be regarded as a periodic perturbation to the slave system that precludes convergence to a phase-locked state. Figure 3.7(b) shows the simulation results, from the same initial condition, when the coupling is given by $g = 0.5$ and $l = 0.4$. In this case the slave network synchronizes to the global cue. Although the local coupling is not strong enough to synchronize the cycle network locally, the global coupling is sufficiently attractive to preclude the slave system to converge to a phase-locked state and forces the slave system to follow the global cue.

As a second example, consider the the network in Figure 3.6(b), which consists of a 5-slave-nodes all-to-all network plus an omnipresent global cue (in red). We simulate the network from a random initial condition and coupling given by $g = 0.05$ and $l = 0.05$. The simulation results are shown in Figure 3.8(a). It can be seen that the network does not synchronize. As for the previous example, in this case the global coupling is too weak and it can be regarded as a perturbation precluding convergence to a phase-locked state. Figure 3.8b shows the simulation results when the coupling is given by $g = 0.1$ and $l = 0.05$. In this case, the global coupling is strong enough to force the slave system to follow the global cue.



(a) (b)

Figure 3.7: Simulation results for the network in Figure 3.6(a) for the identical natural frequencies case. (a): Results when the coupling is given by $g = 0.1$ and $l = 0.4$, since the global coupling is not strong enough the network do not synchronize; (b): Results when the coupling given by $g = 0.5$ and $l = 0.4$, the global cue is attractive enough and the network synchronizes.

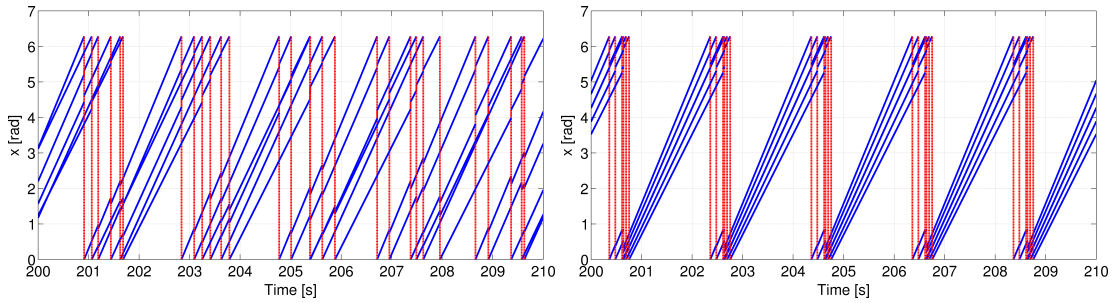


(a) (b)

Figure 3.8: Simulation results for the network in Figure 3.6(b) for the identical natural frequencies case. (a): Results when the coupling is given by $g = 0.05$ and $l = 0.05$, since the global coupling is not strong enough the network do not synchronize; (b): Results when the coupling is given by $g = 0.1$ and $l = 0.05$, the global cue is attractive enough and the network synchronizes.

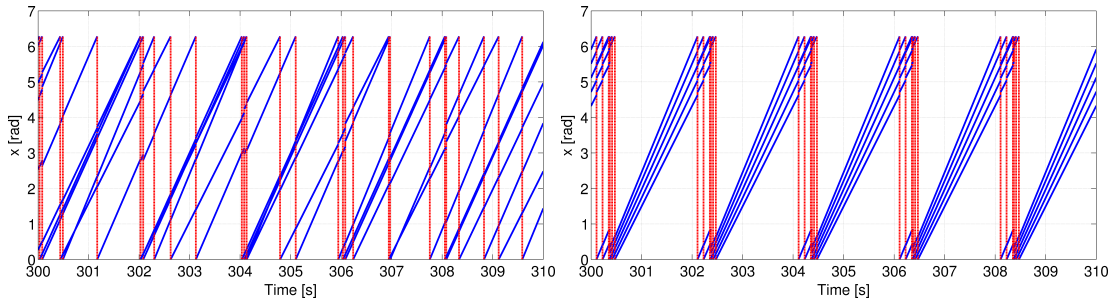
To examine the heterogeneous case, we set the non-identical natural frequencies as $[\pi, 1.1\pi, 1.05\pi, \pi, 0.95\pi, 0.9\pi]$. Figure 3.9(a) shows the simulation results for the network in Figure 3.6(a) when the coupling is given by $g = 0.1$ and $l = 0.1$. Hence, conditions in Theorem 3.2 do not hold. It can be seen that the network does not synchronize in frequency, i.e., each oscillator fires with a different frequency. Figure 3.9(b) shows the simulation results when the coupling is given by $g = 0.6$ and $l = 0.1$, i.e., conditions in Theorem 3.2 do hold. It can be seen that the network synchronizes in frequency and that every slave oscillator fires following the frequency of the global cue $w_g = \pi$.

As second example for the non-identical case, we simulate the network in Figure 3.6(b). The simulation results when the coupling is given by $g = 0.05$ and $l = 0.01$ (conditions in Theorem 3.2 do not hold) are shown in Figure 3.10a. It can be seen that the network does not synchronize in frequency. Figure 3.10(b) shows the simulation results when the coupling is given by $g = 0.6$ and $l = 0.05$. In this case, the global coupling is strong enough to force the slave oscillators to fire following the frequency of the global cue $w_g = \pi$.



(a) (b)

Figure 3.9: Simulation results for the network in Figure 3.6(a) for the non-identical natural frequencies case. (a): Results when the coupling is given by $g = 0.1$ and $l = 0.1$, conditions in Theorem 3.2 do not hold and the network do not synchronize in frequency; (b): Results when the coupling is given by $g = 0.6$ and $l = 0.1$, since conditions in Theorem 3.2 hold the network synchronizes in frequency.



(a) (b)

Figure 3.10: Simulation results for the network in Figure 3.6(b) for the non-identical natural frequencies case. (a): Results when the coupling is given by $g = 0.05$ and $l = 0.01$, conditions in Theorem 3.2 do not hold and the network do not synchronize in frequency; (b): Results when the coupling is given by $g = 0.6$ and $l = 0.05$, since conditions in Theorem 3.2 hold the network synchronizes in frequency.

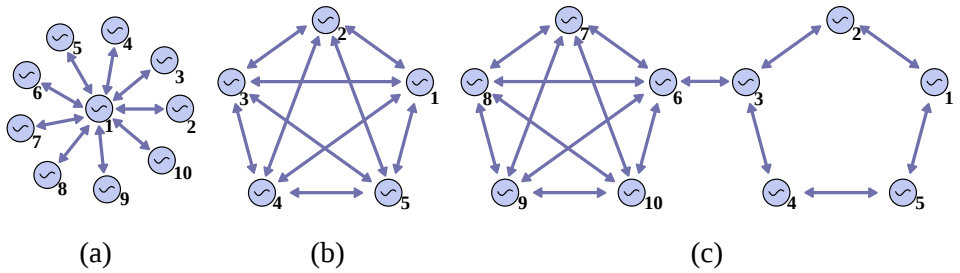


Figure 3.11: Networks used in the numerical examples. (a): Star network with 9 peripheral oscillators ($N = 10$); (b): All-to-all network with $N = 5$; (c): Bidirectional connection of an all-to-all network and a cycle network, $N = 10$.

3.3.2 Decentralized Networks

To illustrate the applicability and conservativeness of the analytical results previously derived for decentralized networks, we use the hybrid systems simulator [82] to simulate the networks shown in Figure 3.11.

As a first example, consider the strongly rooted network depicted in Figure 3.11(a). We simulate the network using the initial condition $x(0,0) = [2\pi, 1, 2, 3, \pi - \epsilon, \pi + \epsilon, 2\pi - 3, 2\pi - 2, 2\pi - 1, 2\pi - \epsilon]$, with $\epsilon \ll 1$; hence, the root oscillator will fire first (for any l). The left plot in Figure 3.12(a) shows the results when $l = 0.4$; it can be seen that even though the root oscillator (black curve in Figure 3.12(a)) fires, the network cannot synchronize. In fact, an example with N approaching infinity as l approaches 0.5 can be found. This stresses the necessity of the condition $l > 0.5$ in Corollary 3.3 to guarantee global synchronization. The right plot in Figure 3.12(a) shows the results when $l = 0.51$; it can be seen that since $l > 0.5$ the network synchronizes.

Now consider the all-to-all network in Figure 3.11(b). We ran simulations using the following initial condition $x_i(0,0) = \frac{2\pi}{5}i$. The left plot in Figure 3.12(b) presents the results when $l = 0.12$; it can be seen that the network does not synchronize since the coupling is too weak. On the other hand, the right plot in Figure 3.12(b) shows the results when $l = 0.51$. Since conditions in Corollary 3.2 are satisfied, the network synchronizes. It should be mentioned that we have found experimentally that the all-to-all network in Figure 3.12(b) synchronizes for $l > 0.13$, confirming the statement in Remark 4 claiming that Corollary 3.2 is conservative in some cases.

Finally, consider the network shown in Figure 3.11(c), which corresponds to the connection of an all-to-all and a cycle network. We use as initial condition $x_i(0,0) = \frac{2\pi}{5}i \mod 2\pi$. Note that, given the network topology, every oscillator fires periodically for every value of l. The left plot in Figure 3.12(c) presents the results when $l = 0.52$; it can be seen that the network does not synchronize. This stresses the fact that a strong coupling, $l = 1$, is required to guarantee global synchronization in general bidirectional PCO networks. On the other hand, the right plot in Figure 3.12(b) shows the results when $l = 1$. As predicted by Theorem 3.5, the network synchronizes.

To illustrate our analytical findings regarding cycle graphs, several numerical experiments were conducted using the hybrid systems simulator. Figure 3.13 shows

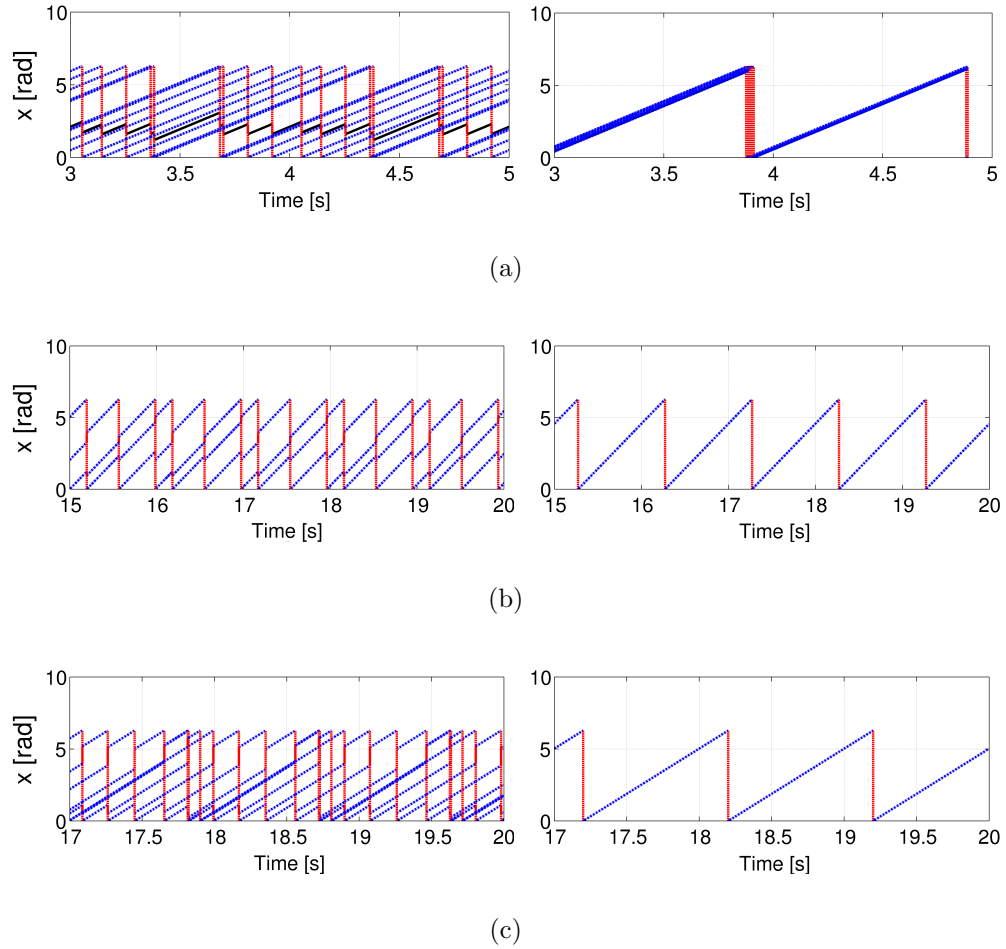


Figure 3.12: Simulation results for the networks of Figure 3.11. (a): Results for the star network in Fig. 3.11(a) when $l = 0.4$ (left) and $l = 0.51$ (right). The solid black curve denotes the root oscillator. As predicted by Corollary 3.3, the network in the right synchronizes. (b): Results for the all-to-all network in Fig. 3.11(b) when $l = 0.12$ (left) and $l = 0.51$ (right). As predicted by Theorem 3.3, the network in the right synchronizes. (c): Results for the network in Fig. 3.11(c) when $l = 0.52$ (left) and $l = 1$ (right). As predicted by Theorem 3.8, the network in the right synchronizes.

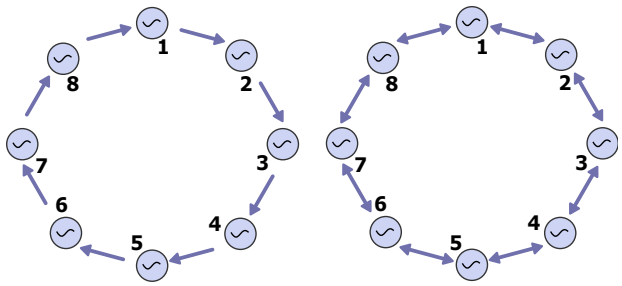


Figure 3.13: Network topologies used in the numerical experiments. Left: unidirectional ring of 8 nodes. Right: the bidirectional, or undirected, version of the ring of 8 nodes. Natural frequencies were set to $w = 2\pi$ for all the experiments.

the PCO networks used in the simulations consisting of 8 oscillators interacting on a bidirectional and a unidirectional graph. For all the experiments, natural frequencies were set to $w_i = w = 2\pi$.

Figure 3.14 shows the results for the bidirectional graph with initial condition $x(0,0) \in \bar{\mathcal{U}}^*$. Solving the condition in Theorem 3.6 gives a critical coupling strength of $l^* = 0.83772$. In the top plot the coupling strength is set below the critical value as $l = 0.8377$; hence, the network cannot synchronize and the oscillators distribute in the interval $[0, 2\pi]$. It can also be seen in the Figure, that $\bar{\mathcal{U}}^*$ is in fact a TDMA-like equilibrium for the system. On the other hand, when the coupling strength is increased to $l = 0.8378$, i.e., above the critical value, the network asymptotically synchronizes, as shown in the bottom plot of Figure 3.14

Figure 3.15 shows the results for the unidirectional graph with initial condition $x(0,0) \in \mathcal{U}_1^*$ when there is a refractory period of length $r = \pi$ in the PRC of

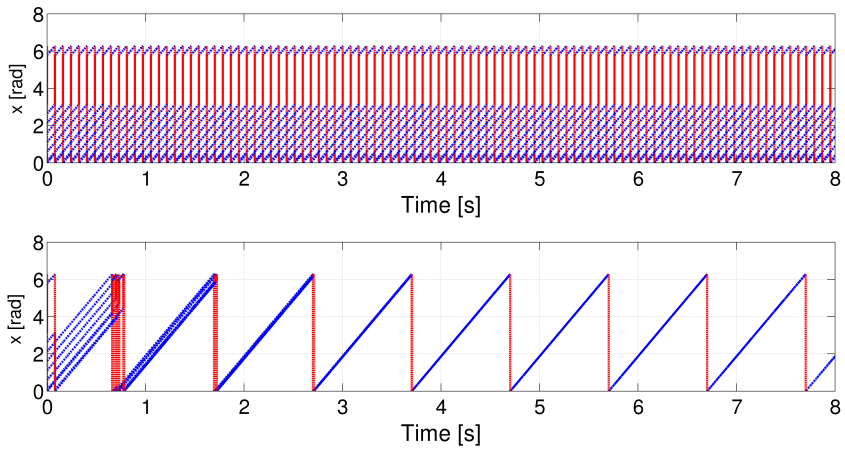


Figure 3.14: Simulation results for the **bidirectional** ring of Figure 3.13 and initial condition $x(0,0) \in \bar{\mathcal{U}}^*$. Red lines denote jump instants and blue lines denote phase values. On the top plot $l = 0.8377$; since from Theorem 3.6 we have $l^* = 0.83772$ the network cannot synchronize. On the bottom plot $l = 0.8378$; since in this case $l > l^*$ the network synchronizes.

oscillator 1. Solving the condition in Theorem 3.7 gives a critical coupling strength of $l^* = 0.8571$. It can be seen in the top plot that when $l = 0.857 < l^*$ the network cannot synchronize and the oscillators distribute in the interval $[0, 2\pi]$. Note that $\mathcal{U}_1^*$ is a TDMA-like equilibrium for the system when $l < l^*$. Increasing the coupling strength such that $l = 0.86 > l^*$ asymptotically synchronizes the network, as shown in the bottom plot. Figure 3.16 shows the results for the unidirectional graph when the initial condition $x(0,0) \in \mathcal{U}_2$ and there no oscillator is affected by a refractory period. In this case, the network cannot synchronize even when the coupling strength is $l = 1$ (the maximum possible value), which is shown in the top plot of Figure 3.16. The bottom plot shows the results when a refractory period of

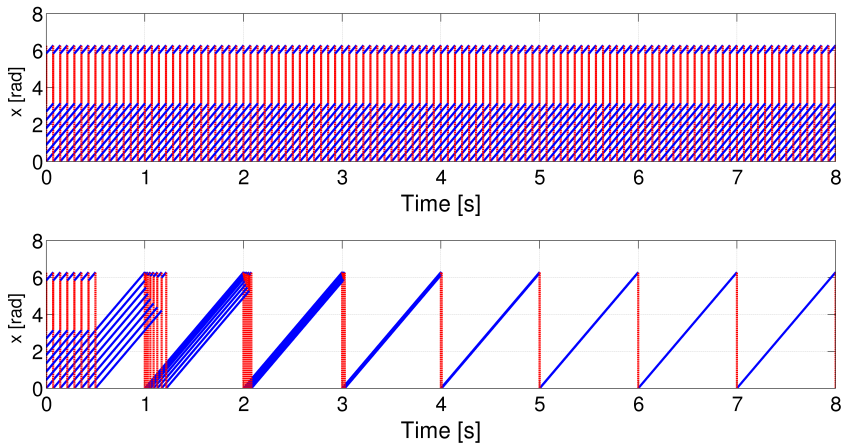


Figure 3.15: Simulation results for the **unidirectional** ring of Figure 3.13, initial condition $x(0,0) \in \mathcal{U}_1^*$ and there is a refractory period of length $r = \pi$ in node 1. Red lines denote jump instants and blue lines denote phase values. On the top plot $l = 0.857 < l^*$ hence the network cannot synchronize. On the bottom plot $l = 0.86$; since in this case $l > l^*$ the network synchronizes.

length $r = \pi$ is introduced in the PRC of oscillator 1. The network recovers the synchronization properties and synchronizes. It should be noted that since $l = 1$, an absorption phenomenon occurs yielding synchronization in finite time.

Figure 3.17 shows the critical strength l^* as a function of the number of oscillators N for both the unidirectional (blue curve) and bidirectional (red curve) cases. It can be seen (also deduced from the condition in the theorems) that l^* is always larger for unidirectional graphs and that, as the number of oscillators grows, the coupling strength goes to the maximal value 1. In fact, for $N = 250$ we have $l^* = 0.99598$ for the unidirectional case, and $l^* = 0.99597$ for the bidirectional case.

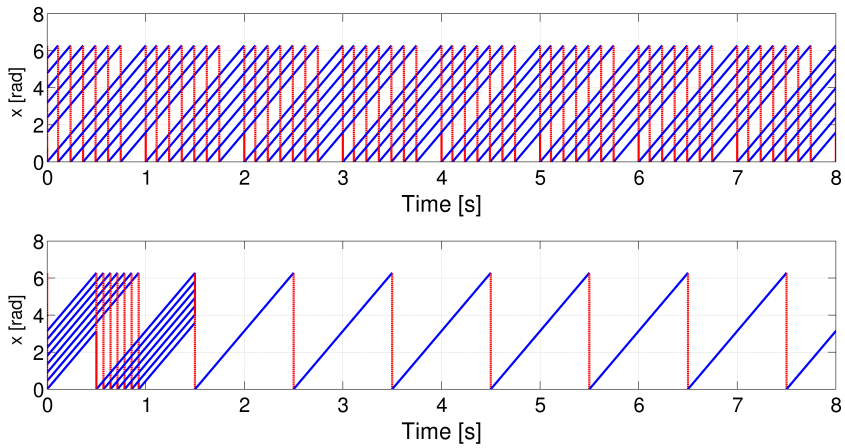


Figure 3.16: Simulation results for the **unidirectional** ring of Figure 3.13 and initial condition $x(0,0) \in \mathcal{U}_2$. Red lines denote jump instants and blue lines denote phase values. On the top plot $l = 1$ and there is no refractory period in any node; as was predicted the network cannot synchronize. On the bottom plot $l = 1$ and there is a refractory period of length $r = \pi$ in node 1. The network recovers the synchronization properties and synchronizes.

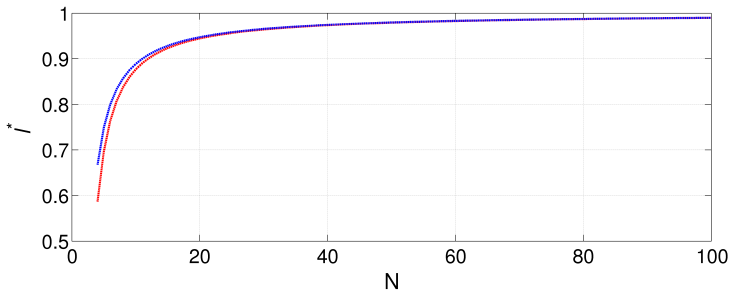


Figure 3.17: Critical coupling strength l^* as a function of N for the unidirectional (blue) and bidirectional (red) cases.

The analytical results obtained in this Chapter provide conditions to ensure synchronization in a variety of networks of PCOs. Inspired by these findings, in the next Chapter we will develop a PCO-based time synchronization protocol

for wireless sensor networks that enjoys all the convergence properties derived for PCOs. Moreover, as a consequence of the simple synchronization mechanism of PCOs, the protocol is simple and naturally scalable.

Chapter 4

PCO-Based Synchronization Protocol for Wireless Sensor Networks

The theoretical results obtained in the previous Chapter suggest that the PCO paradigm is an appealing synchronization method for networks of agents. This Chapter presents the design and evaluation of a new PCO-based synchronization protocol for wireless sensor networks.

The following concepts will be used throughout this chapter.

Definition 4.1 (Clock Drift) *Given a pair of clocks Ck_1 and Ck_2 running with natural frequencies f_{Ck_1} and f_{Ck_2} respectively. The clock drift is defined as the absolute difference of the natural frequencies (cf. Assumption 4 in Chapter 3), i.e., $|f_{Ck_1} - f_{Ck_2}|$*

Portions of this Chapter have been previously published in [101], [102], and [103].

Definition 4.2 (Clock Cycle Jitter) *Given a clock Ck with nominal natural frequency f_{Ck} and observed frequency over a cycle given by $\bar{f}_{Ck}$, the clock cycle jitter is defined as the deviation of the observed frequency from the nominal natural frequency, i.e., $\bar{f}_{Ck} - f_{Ck}$.*

Definition 4.3 (Clock Skew) *Given a pair of clocks with values $Ck_1(t)$ and $Ck_2(t)$ at a given time instant t, the skew at time t is given by their absolute difference. Similarly, given a network of N clocks with values $Ck_i(t)$, $i \in \{1, \ldots, N\}$, the network, or global, skew at time t is given by the maximum absolute difference over the set of clocks, i.e., $\max_{i,j \in \{1,\ldots,N\}} |Ck_i(t) - Ck_j(t)|$.*

4.1 Motivation

Providing a common notion of time is one of the most basic services in any distributed system. Several applications in wireless sensor networks rely on the existence of a precisely synchronized time [87], first because of energy efficiency. In order to increase the energy efficiency in the network, nodes minimize their duty cycle by alternating between sleep/awake cycles. Nodes must wake up at the precise instant in order to communicate with their neighbors. Secondly, time synchronization is critical for distributed monitoring and information fusion where

a precise ordering of events is needed. Without a precise time synchronization protocol, a wireless sensor network loses great part of its potential.

The time synchronization problem has been a subject of intense research since the massification of wireless sensor networks. The first approach to the problem was to design protocols mimicking those used in standard wired computer networks. As the understanding of the time synchronization problem in wireless sensor networks improved, new ad-hoc protocols have been proposed with variable success. In order to solve the time synchronization problem in a large scale network, it is important to count with distributed synchronization protocols that are simple and scalable [87]. Despite many centralized synchronization protocols have been proposed and currently conform the state-of-the-art, these strategies rely heavily on a leader reference node, and thus, they are not robust to many existing phenomena in wireless networks such as network reconfiguration and limited energy resources. Moreover, the use of a reference node makes the accuracy susceptible to an accumulative error that grows with the network dimension. These centralized protocols are inherently incompatible with the natural structure of wireless sensor networks that is ad hoc, time varying, and distributed. In the following sections, we will give a brief review of existing synchronization protocols and present a new PCO-based synchronization protocol, which enjoys the theoretical synchronization

properties of our PCO model and, furthermore, is simple, fully distributed, and naturally scalable.

4.2 Review of Existing Synchronization Protocols

Several synchronization protocols are currently available for wireless sensor networks. In the following we make a distinction based on the strategy to communicate information between nodes in the network. First, we consider packet-based protocols, which use the information inside the synchronization message to synchronize the network, i.e., the content of the synchronization packet is important for the protocol. We denote as pulse-based protocols those algorithms that do not rely on the content of the synchronization message to synchronize the network. Note that although the message is not necessarily a pulse (many physical layers are not able to handle pulses) but a short control packet, in pulse-based protocols the content of the message is not important, only the time instant at which the packet is received is used to synchronize the network, i.e., the interruption pulse coming from the physical layer.

4.2.1 Packet-Based Synchronization Protocols

Reference Broadcast Synchronization

The Reference Broadcast Synchronization (RBS) protocol [28] is a scheme in which nodes send reference beacons to their neighbors using physical layer broadcasts. RBS exploits the broadcast nature of the wireless physical channel used in wireless sensor networks to synchronize a set of receivers with one another. A reference node is elected to synchronize all other nodes using a sequence of broadcast messages. Since differences in the propagation times can generally be neglected in sensor networks, a reference message arrives at the same instant at all receivers. Each receiving node records a timestamp at the reception of a broadcast message, which is exchanged with neighboring nodes to calculate relative clock offsets. Although RBS is designed for single-hop time synchronization only, or within a single broadcast domain, nodes that participate in more than one broadcast domain can be employed to convert the timestamps between local clock values of different subnetworks. External clock references attached to one node, for example a GPS receiver, can be treated like reference broadcasts to transform the local timestamps into Coordinated Universal Time (UTC). The main disadvantage of the approach is that additional message exchange is necessary to communicate

the local time-stamps between the nodes, which creates a large communication overhead.

Timing-sync Protocol for Sensor Networks

The Timing-sync Protocol for Sensor Networks (TPSN) [32] aims to provide network-wide time synchronization by using a classical sender-receiver approach. The TPSN protocol works with a hierarchical structure created by first electing a root node and building a spanning tree of the network during the initial level discovery phase. In the synchronization phase of the algorithm, nodes synchronize to their parent in the tree by a two-way message exchange. Using the timestamps embedded in the synchronization messages, the child node is able to calculate the transmission delay and the relative clock offset. MAC layer time-stamping is used to reduce possible sources of uncertainty in the message delay; however, TPSN does not compensate for clock drifts making frequent resynchronization mandatory in order to keep the network properly synchronized. In addition, TPSN causes a high communication overhead since a two-way message exchange is required for each child node. It should be noted that, although unidirectional links are allowed to exists in the network, TPSN uses only bi-directional links to do pairwise synchronization between a set of nodes.

Flooding Time Synchronization Protocol

The shortcomings present in both RBS and TPSN are tackled by the Flooding-Time Synchronization Protocol (FTSP) [56]. In FTSP, a root node is elected which periodically floods its current time into the network forming an ad-hoc tree structure. MAC layer time-stamping is used to reduce possible sources of uncertainty in the message delay. Each node uses a linear regression table to convert between the local hardware clock and the clock of the reference node, which is dynamically elected by the network based on the smallest node identifier. After initialization, a node waits for a few rounds and listens for synchronization beacons from other nodes. Each node sufficiently synchronized to the root node starts broadcasting its estimation of the global clock. If a node does not receive synchronization messages during a certain period, it will declare itself the new root node and initiates a flooding of its local time. Although FTSP provides good global synchronization at low communication cost on small networks, it potentially incurs large skews between neighboring nodes due to the tree structure employed. Moreover, FTSP requires a larger amount of processing than RBS and TPSN and the maintaining of a table for performing linear regression that grows with the number of neighbors.

Gradient Time Synchronization Protocol

Gradient Time Synchronization Protocol (GTSP) [89] is a completely distributed time synchronization protocol, i.e., GTSP does not require neither a tree topology nor a reference node. GTSP focuses mainly on the synchronization error between neighbors. Existing time synchronization algorithms provide on average good synchronization between arbitrary nodes, however, neighbor nodes in a network may be poorly synchronized depending on the tree used to perform synchronization. In GTSP nodes periodically broadcast synchronization beacons to their neighbors trying to agree on a common logical clock by using a simple update algorithm based on consensus-like iterations for drift compensation and MAC layer time-stamping for reducing uncertainty in the message delays. It was shown analytically that by employing the GTSP algorithm, the logical clock of every participating node converges to a common virtual logical clock. GTSP relies on local information only, making it robust to node failures and changes in the network topology, which is a known drawback of tree-based protocols. Moreover, GTSP can improve the synchronization error between neighboring sensor nodes compared to tree-based time synchronization protocols, while maintaining a similar network-wide synchronization error.

PulseSync

The basic idea of PulseSync [53] is to distribute information regarding clock values as fast as possible, while minimizing the number of messages required to flood the network. In particular, nodes send messages only once in a given interval of time. Since a node cannot forward any information before getting a message, an intermediate node in a line topology has to wait for at least one message from a neighbor, and after its reception it must send a message as quickly as possible in order to not slow the flooding down. Thus, PulseSync develops a flooding of a pulse-like message through the network, implicitly building a breadthfirst search tree. This technique further implies that the pulse is originated at a given node, which becomes the root of the tree. The root node is the only node from which all nodes in the network obtain information, making its clock the reference for synchronization. To keep clock skews small at all times, nodes employ a drift compensation, also relative to the root node. Moreover, to reduce the effects of the random jitter, the drift estimates are based on a linear regression, similarly as the regression performed in FTSP. PulseSync is inherently a tree-based protocol and hence it suffers from the same drawbacks as the other tree-based algorithms although its convergence time is lower.

Glossy

Glossy [30] exploits constructive interference of IEEE 802.15.4 symbols for fast network flooding and implicit time synchronization. A timing requirement to make concurrent transmissions of the same packet interfere constructively is derived, allowing a receiver to decode the packet even in the absence of capture effects. To satisfy this requirement, Glossy temporally decouples flooding from other network activities. Unlike existing flooding schemes, Glossy's performance exhibits no noticeable dependency on node density, which facilitates its application in diverse real-world settings. However, although Glossy is an intriguing method, its use of the physical layer makes it harder to apply it in standard sensor networks where the user has no access below the MAC layer. Moreover, its formulation is based on the IEEE 802.15.4 physical layer and is not clear how Glossy can be adapted for networks operating using different physical layers.

4.2.2 Pulse-Based Synchronization Protocols

Reachback Firefly Algorithm

The Reachback Firefly Algorithm (RFA) [104] is inspired by the way neurons and fireflies spontaneously synchronize in nature. Initially based on the PCO model by Mirollo and Strogatz [62], the RFA protocol modifies the classical PCO model

to deal with non-deterministic delays and clock drifts. The authors recommend the use of a CSMA-like strategy at the MAC layer to avoid collisions that naturally occur when all the nodes aim to transmit at the same time. The flow of the protocol is as follows: each node periodically generates a pulse (simple empty message) and observes pulses from other nodes to adjust its own firing phase. It is proven analytically in [104] that every node converges to the common virtual time reference. RFA only provides local synchronization, i.e., nodes agree on the firing phases but do not have an absolute common notion of time. Another drawback of RFA is the fact that it has a high communication overhead. The RFA protocol has shown a slightly worse performance than FTSP in initial pilot implementations [104].

Scalable Sync

The Scalable Sync protocol [43] is also based in the PCO model. Specifically, it uses directly the leaky integrate and fire model by Peskin at each node inheriting the theoretical synchronization properties proven by Mirollo and Strogatz in [62]. As a key application mechanism, the authors propose a simple circuit for pulse detection at the physical layer, and as a minor, yet very important, modification to the PCO dynamics, the authors propose the use of a refractory period to ensure stability of the synchronized state and prevent firing storms. The main advantages

of the Scalable Sync protocol are its natural scalability, shown in numerous large scale simulations, and its simplicity, which comes from the identical treatment given to every received pulse irrespective of the origin. The main drawback of the protocol is that its convergence speed decreases dramatically when the network is sparse and large scale.

4.3 A New Pulse-Coupled Synchronization Protocol

Inspired by the biological principles behind synchronization of PCOs and supported by our theoretical findings, we designed a PCO-based time synchronization protocol for networks of agents interacting through wireless channels. Although our protocol shares the inspiration source with others pulse-based protocols such as RFA and Scalable Sync, our protocol outperforms previous PCO-based efforts since it uses the optimized PCO model presented in the previous chapters instead of the classical biological PCO model. In particular, the use of an optimal PRC gives faster synchronization and the use of a refractory period reduces the energy consumption.

4.3.1 Algorithmic Formulation

The algorithm behind the protocol is as follows. Every node in the network counts with a unique identifier, which can be assigned upon joining the network in a similar way to how IP addresses are assigned or can be an identifier assigned in the construction of the device, e.g., MAC address. Although the unique identifier is not used by the synchronization protocol, it might be needed for upper layer data processing or for conducting an initial flooding. For every node in the network, the internal phase variable is implemented using a counter that is initialized at 0 on startup and is continuously incremented at a constant frequency given by the processor's oscillating crystal or a fraction of it until it reaches the limit value CL. Once the counter overflows, it fires a counter event that makes the counter to reset and a synchronization message is broadcasted. When a message arrives from a neighbor oscillator, it triggers a message event that makes the counter to be updated based on the PRC and the coupling strength.

The protocol has several configuration parameters that need to be pre-set before initiating the protocol. This process can be done using a configuration file that is loaded at startup. The following parameters define the operation of the protocol:

- Upper limit of the phase counter: CL
- Role of the oscillator in the network (global cue or slave)

- Local and global coupling strengths: g, l
- PRC family (advance-only or advance-delay)
- Shape parameters for the different PRCs

The protocol that every node in the network implements can be summarized as follows.

Protocol 4.1 (PCO-Protocol) *Upon joining the network, each sensor i records its unique identifier, loads its configuration parameters, initializes its phase counter as $x_i = 0$ and starts increasing the counter at constant rate w.*

1. *At each phase event x_i = CL node i broadcasts a pulse-like message and resets the counter to $x_i = 0$.*
2. *Upon receiving a pulse, node i updates its phase following the PRC Q and the coupling strength l.*

In its current form, the PCO-Protocol can be implemented at any layer of the networking protocol stack. However, ideally the protocol should be implemented as low as possible to improve its accuracy. We propose to implement the protocol at the MAC layer to take advantage of the simple implementation requirements at the MAC layer (pure software) and the closeness to the physical layer that reduces non-deterministic delays from the packet processing at higher layers. Moreover, the

short control packets usually exchanged at the MAC layer in several MAC protocols (CDMA, ALOHA, etc.) represent an appealing alternative to emulate pulses. Note that implementing the impulsive updates at the MAC layer is equivalent to the classical MAC layer time-stamping used in many packet-based synchronization protocols. Figure 4.1(a) shows the flow diagram of the synchronization protocol and Figure 4.1(b) shows a version of the networking protocol stack suitable for wireless sensor networks that is an hybrid between the classical OSI model and the TCP-IP model. Our PCO-Protocol is implemented inside the Data-Link layer, at the MAC layer, as shown in Figure 4.1(b).

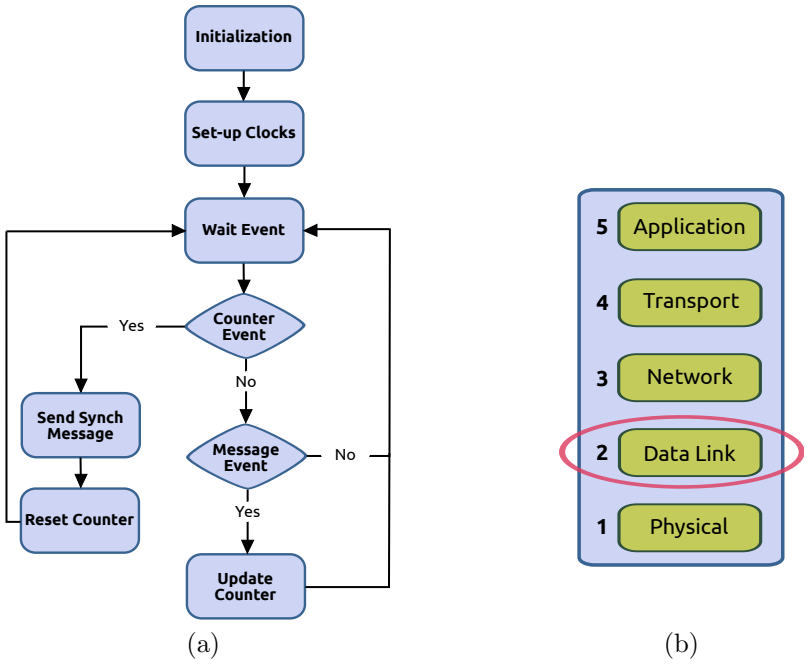


Figure 4.1: (a): Flow diagram of the PCO synchronization protocol (b): hybrid layered architecture; the protocol is located inside the data link layer, at the MAC layer.

4.3.2 Synchronization Properties of the Protocol

The synchronization protocol inherits all the synchronization properties derived for networks of PCOs in the previous chapters. The following results are a direct consequence of the synchronization results available for networks of PCOs. The proofs and a rigorous formulation of the problems can be found in [100, 103].

Theorem 4.1 *Consider a network of PCOs with a refractory period of length r in the PRC, if the initial phases are such that*

$$\max_{i,k\in\mathcal{V}} |x_i(0,0) - x_k(0,0)| < \Lambda \in (0,\pi],$$

$l \in (0,1]$ *and the interaction topology is strongly connected, then the oscillators can be perfectly synchronized for any* $r \leq \Lambda$.

The previous Theorem state that a network of PCOs can synchronize if the initial conditions are contained in a semicircle even if there exists a refractory period in the PRC of all the oscillators. The following Theorem relaxes the semicircle condition and it also covers synchronization over unreliable networks. In this case, however, synchronization is achieved with a given probability.

Theorem 4.2 *Consider a network of PCOs with a refractory period of length r in the PRC, interacting on a strongly connected graph $\mathcal{R}$. If the initial phases are*

independent and uniformly distributed in the interval $[0, 2\pi]$*, and every pulse transmission is associated with a successful delivery probability* p*, then the probability of synchronization is no less than*

$$\begin{aligned} P = \prod_{i \in \mathcal{V}} \Bigg\{ & 1 - \frac{1}{2} \left(\frac{\pi + r}{2\pi} + (1-p) \frac{\pi - r}{2\pi} \right)^{|\mathcal{N}^{i-}|} \\ & + \frac{1}{2} \left[\left(\frac{\pi + r}{2\pi} \right)^{|\mathcal{N}^{i-}|} - \left(\frac{1}{2} + (1-p) \frac{r}{2\pi} \right)^{|\mathcal{N}^{i-}|} \right] \\ & \times \left[1 - \left(\frac{\pi + r}{2\pi} + (1-p) \frac{\pi - r}{2\pi} \right)^{\min_{i \in \mathcal{V}} |\mathcal{N}^{i-}|} \right] \Bigg\} \end{aligned} \tag{4.1}$$

Remark 4.1 *It should be noted that Theorem 4.1 requires the phases to be contained in an interval of length* Λ*. However, in real deployments this constraint is unlikely to hold. One way to ensure the phases are contained in a small interval, and hence to exploit the largest possible refractory period to save energy, is to use an initial flooding as follows. Any sensor that wants to run the synchronization strategy broadcasts a reset packet (with its unique ID specified in the packet) and resets its phase to 0 to initiate a synchronization process. Every sensor receives the packet resets its phase to 0 and immediately passes the packet to its neighbors. A sensor having received the reset packet once will ignore all subsequently arriving identical reset packets to prevent broadcast storms. Note that although the reset packet cannot synchronize the network due to the existence of a non-deterministic processing time, it can reduce the phase difference.*

Remark 4.2 *Theorem 4.2 focuses mainly on large scale networks. The approximations needed in the proof of Theorem 4.2 ensure that if the network is large then the lower bound on the synchronization probability is close to the actual synchronization probability. On the other hand, if the network consists of a few nodes, then the lower bound could be to loose to be meaningful.*

4.4 Evaluation by Simulation

To test the strategy in a realistic environment, we selected Qualnet™ [85] as our simulation platform. Qualnet is a network simulation tool that can be used to simulate wireless and wired communication networks. It was first released in 2000 by Scalable Networks and has been widely used since to simulate MANETs, satellite networks, and sensor networks, among others. Figure 4.2 shows Qualnet's main interfaces.

The protocol was implemented as a standard wireless MAC layer protocol in Qualnet. Its operating philosophy is based on the CSMA and IEEE 802.11 protocols, combining carrier sense for collision avoidance and control packets to emulate the pulses used to communicate entrainment information. The current version works using IEEE 802.11b protocol at the physical layer and it includes data sending capabilities, as well as an adaptive sleep mode to save energy by reducing idle listening.

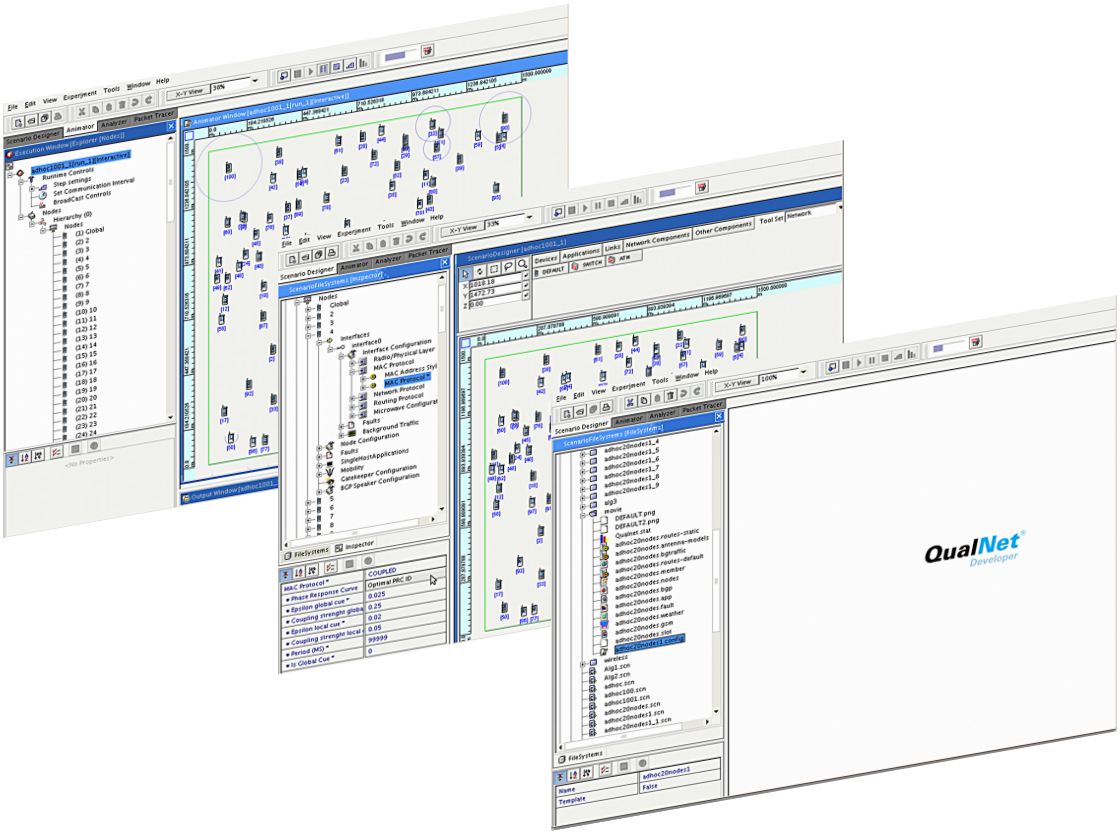


Figure 4.2: Qualnet's user interface. The scenarios for the simulations are first designed in the scenario designer interface and then simulated in the simulation interface. The PCO-based protocol is included as a standard MAC layer protocol in the protocol selection menu where all the configuration parameters can be modified by the user.

The algorithm can be selected as MAC layer protocol by the user in Qualnet's graphical interface, which allows the user to configure the following parameters of the algorithm:

- Natural period of the oscillator (counter limit)
- Role of the oscillator in the network (global cue or slave)
- Local and global coupling strengths
- Advance-only phase response curve family
- Advance-delay phase response curve family
- Shape parameters for the different PRCs

Simulations are realistic, in the sense that they include all phenomena found in real wireless networks: fading, transmission delays, collisions, limited transmission radius, among others. Moreover, they include the effects of quantization in both phase and PRCs. The complete algorithm consists of 3 operating modes: synchronization, data sending, and sleep (energy saving mode), as shown in Figure 4.3. The source code for implementing the protocol in Qualnet as a wireless MAC layer protocol is given in Appendix A.

In the following, we present simulations of our synchronization protocol for a variety of networks, some of them fulfill the assumptions in Theorems 4.1 and 4.2

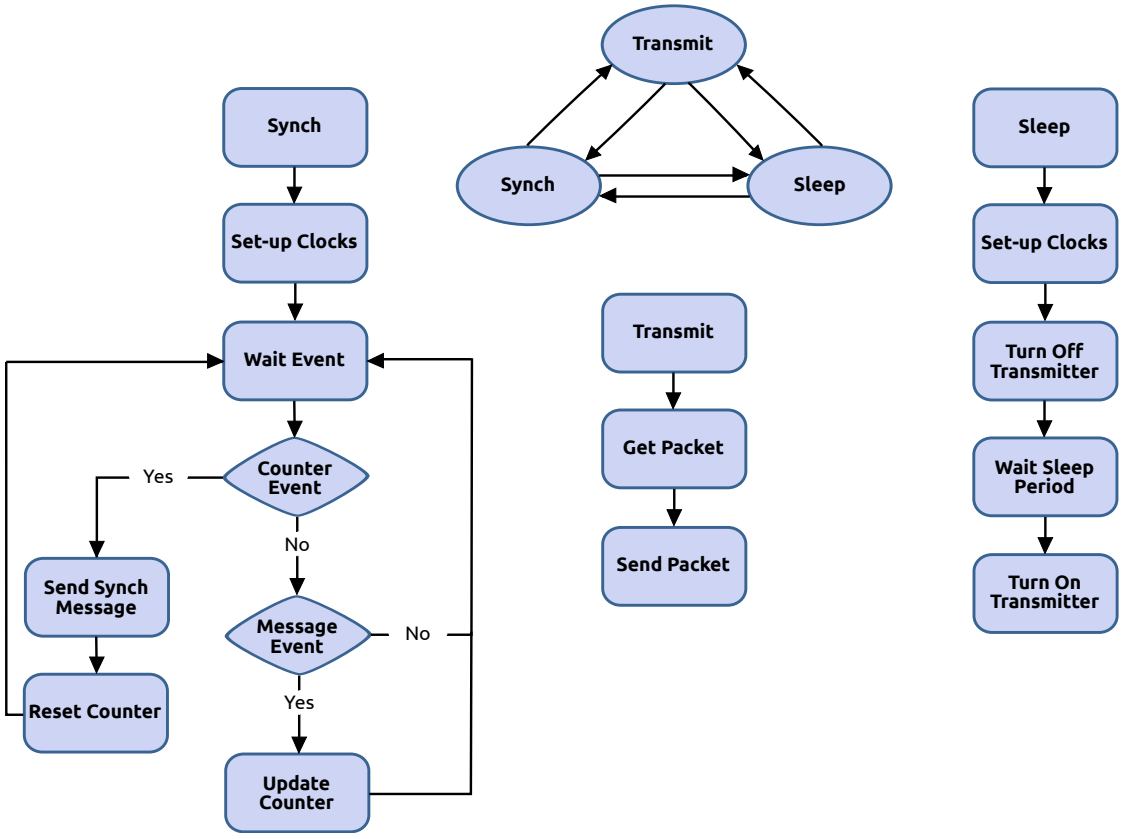


Figure 4.3: Flow diagram of the complete PCO-based wireless synchronization protocol. The protocol consists of three operating modes: synchronization, data sending, and sleep. When the nodes in the network are exchanging pulses to perform synchronization, the protocol works in synchronization mode. When nodes turn off the wireless antenna to save energy, the protocol switches to sleep mode. When nodes exchange information in the form of data packets to, for example, share sensor measurements, the protocol switches to data sending mode.

but we also study networks that are out of the scope of Theorems 4.1 and 4.2. We explore the effect of the protocol's parameters (PRC shape and coupling strength) on the synchronization properties of the algorithm focusing mainly on the time it takes and the amount of energy required to achieve synchronization.

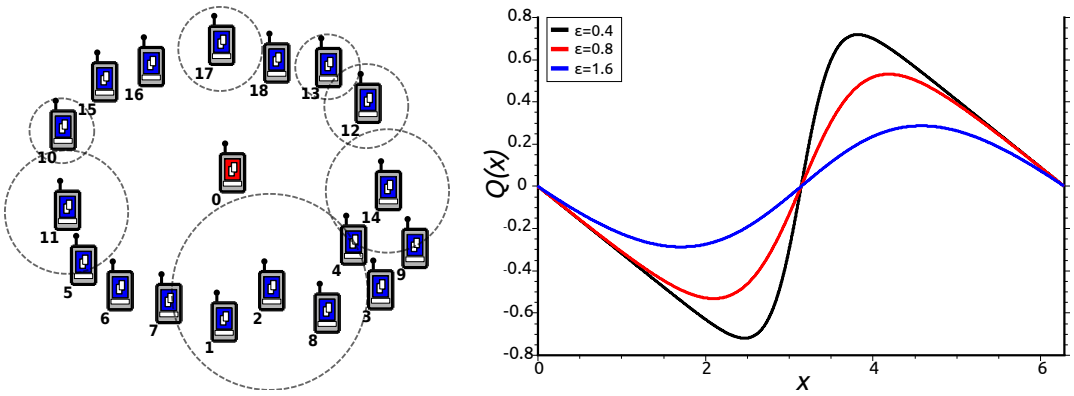
4.4.1 Synchronization to a Global Cue

Simulations were conducted to study the time to synchronization in a wireless network formed by 18 slave oscillators and 1 omnipresent global cue. Natural frequencies were set as $w = 2\pi$, i.e., every node fires with a natural period of $1s$. The agents interact following a static communication topology where each slave oscillator has between 8 and 10 neighbors. Figure 4.4(a) shows the network implemented in Qualnet.

The feedback strategy was implemented using a hyperbolic-like PRC, for both Q_g and Q_l, given by:

$$Q_q(x) = \frac{\tanh\left(\frac{x-\pi}{\epsilon_q}\right)}{\tanh\left(\frac{\pi}{\epsilon_q}\right)} - \frac{x-\pi}{\pi}, \quad q \in \{g, l\} \tag{4.2}$$

where ϵ_q is a tuning parameter controlling the shape of the PRC. Figure 4.4(b) shows graphs of the PRC (4.2) for different values of ϵ. Note that the graph of



(a) (b)

Figure 4.4: (a): Network consisting of 18 slave oscillators (blue) and 1 omnipresent global cue (red) used in the time to synchronization experiments. (b): PRC used in the time to synchronization experiments for different values of ϵ.

(4.2) is inside the set Ω for all the values of ϵ used. Moreover, Q_g and Q_l are such that $Q_q(x_i) > 0,\ q \in \{g, l\}$ if $x_i \in (\pi, 2\pi)$ and $Q_q(x_i) < 0,\ q \in \{g, l\}$ if $x_i \in (0, \pi)$. Therefore, condition 2) in Theorem 3.1 holds.

To analyze the influence of the tuning parameter ϵ on the time to synchronization, we conducted simulations in Qualnet for different values of ϵ_g and ϵ_l. Initial conditions were randomly chosen following a uniform distribution on the interval $[0, 2\pi]$ for each agent. Coupling strengths were selected as $g = l = 0.01$. Table 4.1 presents results of the simulations where each scenario was simulated 100 times and results were averaged. Results show that with a decrease in ϵ_g, the time to synchronization is reduced; while variations on ϵ_l can either reduce or increase the time to synchronization. A theoretical analysis regarding the effect of the tuning parameter on the time to synchronization is available in [102].

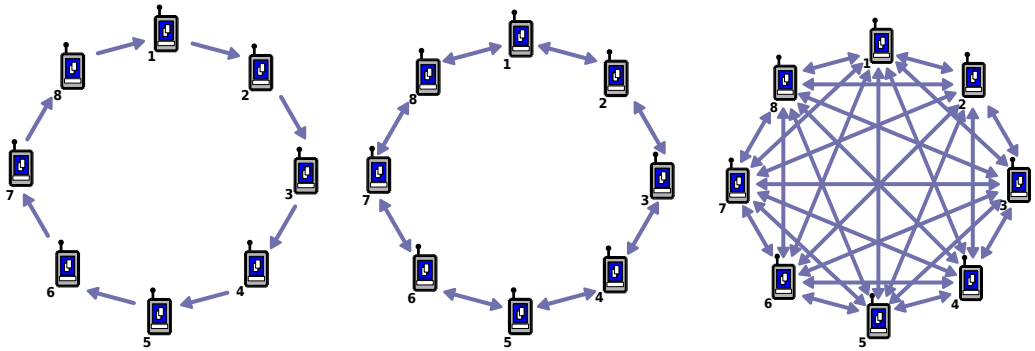
Table 4.1: Time to synchronization $[s]$ for the network in Figure 4.4(a) using the PRC (4.2) for different values of the parameter ϵ, and coupling given by: $g = l = 0.01$.

$\epsilon_g \setminus \epsilon_l$	0.05	0.1	0.2	0.4	0.8	1.6
0.4	22.93	23.17	23.14	22.58	21.53	22.44
0.8	24.95	25.21	25.36	24.23	23.63	24.34
1.6	30.14	31.92	31.75	30.35	28.15	29.09

Table 4.2: Time to synchronization $[s]$ for the network in Figure 4.4(a) under the PRC (4.2) for different values of the couplings g and l, and $\epsilon_g = 0.4$, $\epsilon_l = 0.05$.

$g \setminus l$	0.01	0.02	0.03	0.04	0.05	0.06
0.01	22.93	23.21	26.37	27.26	no sync	no sync
0.02	17.49	18.90	22.03	24.60	24.38	21.39
0.03	14.18	14.99	18.03	19.93	20.35	19.91

To analyze the influence of the coupling strengths on the time to synchronization, we conducted simulations in Qualnet for different values of g and l. Initial conditions were randomly chosen following a uniform distribution on the interval $[0, 2\pi]$ for each agent. The shape parameters of the PRCs were set as $\epsilon_g = 0.4$ and $\epsilon_l = 0.05$. Table 4.2 presents results of the study where each case was simulated and averaged over 100 runs. Results show that a larger g leads to faster synchronization, while a larger l does not necessarily imply faster synchronization. A larger l may inhibit synchronization when g is small, as suggested by Theorem 3.1.



(a) (b) (c)

Figure 4.5: Networks used in the numerical simulations. (a): Unidirectional 8-node network; (b) bidirectional 8-node network; (c): all-to-all 8-node network.

4.4.2 Decentralized Synchronization

Simulations were conducted to study the time to synchronization in a variety of simple networks when using the synchronization-rate-optimal PRC (3.11). Natural frequencies were set as $w = 2\pi$ for all the simulated scenarios. Figure 4.5 shows the networks used in the first set of simulations. Figure 4.5(a) shows an 8-nodes unidirectional network where each node has only 1 in-neighbor. Table 4.3 presents the time to synchronization for the network in Figure 4.5(a) when the length of the refractory period and the coupling strength are varied and each scenario was simulated and averaged over 100 runs. Initial conditions were randomly chosen from a uniform distribution. It can be seen that the length of the refractory period has no effect on the time to synchronization, while an increase in the coupling strength reduces the time the network takes to reach synchronization.

Table 4.3: Influence of the refractory period r and coupling strength l on the time to synchronization $[s]$ for the network in Figure 4.5(a).

$r \setminus l$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.2π	177.75	95.56	58.87	44.49	32.35	25.51	19.97	14.79	10.71
0.4π	175.50	93.96	61.84	44.47	33.74	25.11	19.41	14.81	10.48
0.6π	178.97	94.17	61.70	44.50	33.63	24.97	19.32	14.60	10.41
0.8π	178.97	94.17	61.70	44.50	33.63	24.97	19.32	14.60	10.41
1.0π	178.97	94.17	61.70	44.50	33.63	24.97	19.32	14.60	10.41
1.2π	178.97	94.17	61.70	44.50	33.63	24.97	19.32	14.60	10.41

Table 4.4: Influence of the refractory period r and coupling strength l on the time to synchronization $[s]$ for the network in Figure 4.5(b).

$r \setminus l$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.2π	81.80	46.60	34.05	27.43	22.39	18.65	16.88	10.36	8.34
0.4π	83.46	48.11	32.80	27.94	21.68	18.84	15.94	10.35	8.36
0.6π	85.83	47.47	32.80	27.81	21.74	18.59	15.76	10.35	8.39
0.8π	85.98	47.90	32.89	27.86	21.70	18.45	15.93	10.35	8.39
1.0π	85.98	47.90	32.89	27.86	21.70	18.45	15.93	10.35	8.39
1.2π	85.98	47.90	32.89	27.86	21.70	18.45	15.93	10.35	8.39

Figure 4.5(b) shows the bidirectional version of the network in Figure 4.5(a). Table 4.4 presents the time to synchronization for the network in Figure 4.5(b) when the length of the refractory period and the coupling strength are varied and each case was simulated and averaged over 100 runs. Initial conditions were randomly chosen from a uniform distribution. It can be seen that the length of the refractory period has no effect on the time to synchronization, while an increase in the coupling strength reduces the time the network takes to reach synchronization.

Table 4.5: Influence of the refractory period r and coupling strength l on the time to synchronization $[s]$ for the network in Figure 4.5(c).

$r \setminus l$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.2π	25.47	15.15	10.57	8.32	6.53	5.06	4.02	2.98	2.04
0.4π	27.16	16.03	11.15	8.40	6.53	5.06	4.02	2.98	2.04
0.6π	28.17	16.03	11.15	8.40	6.53	5.06	4.02	2.98	2.04
0.8π	28.17	16.03	11.15	8.40	6.53	5.06	4.02	2.98	2.04
1.0π	28.17	16.03	11.15	8.40	6.53	5.06	4.02	2.98	2.04
1.2π	28.17	16.03	11.15	8.40	6.53	5.06	4.02	2.98	2.04

Finally, Figure 4.5(c) shows an all-to-all connected 8-nodes network. Table 4.5 presents the time to synchronization for the network in Figure 4.5(c) when the length of the refractory period and the coupling strength are varied and each scenario was simulated and averaged over 100 runs. Initial conditions were randomly chosen from a uniform distribution. It can be seen that the length of the refractory period has no effect on the time to synchronization, while an increase in the coupling strength reduces the time the network takes to reach synchronization.

Results presented in Tables 4.3, 4.4 and 4.5 suggest that using a higher coupling strength is the best choice to obtain the lowest time to synchronization as possible. Similarly, the largest possible refractory period should be used since it does not affect the time to synchronization and it can reduce the energy consumed in the synchronization process. An important fact to note is that highly connected networks, all-to-all networks as the extreme case, present a lower time to synchronization.

Table 4.6: Influence of the refractory period r and coupling strength l on the time to synchronization $[s]$ for the faulty network in Figure 4.6(a).

$r \setminus l$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.2π	148.29	73.26	48.33	35.99	28.19	23.12	19.72	16.59	13.42
0.4π	149.27	74.23	48.23	36.01	28.78	23.06	19.77	16.87	13.10
0.6π	149.29	74.19	48.20	36.03	28.66	23.01	19.79	16.63	13.97
0.8π	148.41	73.44	48.20	36.03	28.06	23.01	19.79	16.45	13.92
1.0π	149.25	74.33	48.20	36.03	28.81	23.01	19.79	16.45	13.92
1.2π	149.31	74.19	48.20	36.03	28.66	23.01	19.79	16.45	13.92

To complement the simulations previously conducted, we consider a second set of networks that are shown in Figure 4.6. Figure 4.6(a) shows the 8-node bidirectional network in Figure 4.5(b) with a permanent link failure that makes node 2 to have only one in-neighbor. Table 4.6 presents the time to synchronization for the faulty network in Figure 4.6(a) when the length of the refractory period and the coupling strength are varied and each scenario is simulated and averaged over 100 runs. Initial conditions were randomly chosen from a uniform distribution. Once again, it can be seen that the length of the refractory period has no effect on the time to synchronization, while an increase in the coupling strength reduces the time the network takes to reach synchronization. However, the results are comparable to the ones shown in Table 4.3, i.e., for the unidirectional network in Figure 4.5(a). This fact suggests that the time to synchronization is related to the indegree of the network. An analytical argument for this fact can be found in [101].

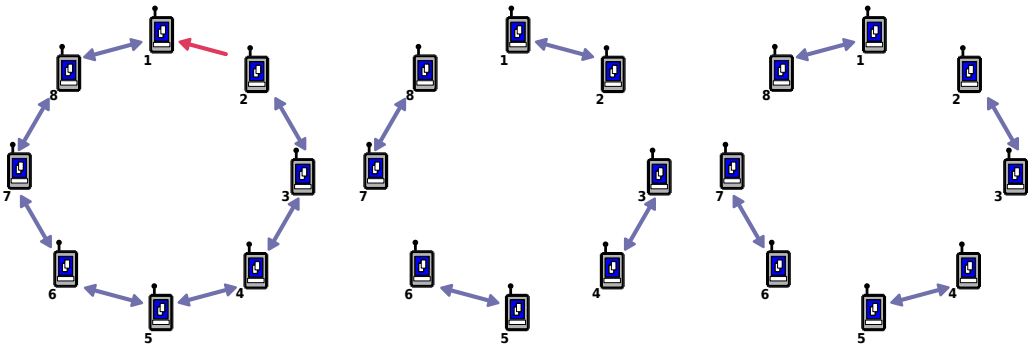


Figure 4.6: Networks used in the numerical simulations. (a): Bidirectional 8-node network with a permanent fault in one link; (b)-(c): unconnected complementary networks, switching between the two gives a connected network .

Figures 4.6(b) and 4.6(c) show an 8-nodes bidirectional network with disconnected communication topology. However, if the network switches continuously between the communication topologies shown, the union is connected. An analytical argument that ensures synchronization for switching networks with connected union graph can be found in [101]. To show that this is indeed the case, we conducted simulations of the switching network in Figures 4.6(b)-4.6(c). Table 4.7 presents the time to synchronization when the length of the refractory period and the coupling strength are varied and each setup is simulated and averaged over 100 runs. Initial conditions were randomly chosen from a uniform distribution. Results show that the length of the refractory period has no effect on the time to synchronization, while an increase in the coupling strength reduces the time the network takes to reach synchronization. It should be noted that the results are

Table 4.7: Influence of the refractory period r and coupling strength l on the time to synchronization $[s]$ for the switching network shown in Figures 4.6(b) and 4.6(c).

$r \setminus l$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.2π	172.46	94.58	65.38	50.60	42.65	33.95	31.51	23.86	19.17
0.4π	176.42	91.87	67.55	51.74	39.00	34.68	31.35	23.07	18.35
0.6π	178.14	94.03	67.79	50.86	39.00	34.09	30.75	22.20	18.09
0.8π	178.23	94.43	67.91	51.05	39.13	34.21	30.79	22.06	18.37
1.0π	178.23	94.43	67.91	51.05	39.13	34.21	30.79	22.06	18.37
1.2π	178.23	94.43	67.91	51.05	39.13	34.21	30.79	22.06	18.37

comparable to the ones shown in Table 4.3, i.e., for the unidirectional network in Figure 4.5(a). This is reasonable since at each time instant every node in the network has only one in-neighbor.

To show the potential of our protocol, we conducted simulations using the state-of-the-art protocol FTSP and compared the results against our protocol. Table 4.8 shows the results in terms of the network skew and the energy consumed during the synchronization process. We fixed the coupling strength as $l = 0.9$, the refractory period as $r = 1.2\pi$, and conduct 100 runs for each network. Initial conditions were randomly chosen from a uniform distribution. The results show that our protocol outperforms FTSP in all the simulated scenarios for both synchronization error and energy consumption. Taking into account that FTSP presents a synchronization error that grows exponentially with the size of the network [53], and that in the other hand, our protocol is naturally scalable and hence the error is independent

Table 4.8: Comparison of network skew and energy consumption between the PCO-protocol and FTSP for different network topologies.

Topology	Figure 4.5(a)		Figure 4.5(b)		Figure 4.5(c)		Figure 4.6(a)		Figure 4.6(b) & 4.6(c)	
Protocol	PCO	FTSP	PCO	FTSP	PCO	FTSP	PCO	FTSP	PCO	FTSP
Network skew [μs]	20.23	21.20	11.94	13.60	1.09	3.00	9.47	20.19	11.70	13.78
Energy consumption [mJ]	4.16	11.33	3.35	7.19	0.81	2.08	5.56	10.56	7.34	15.71

of the size of the network, results in Table 4.8 indicate that our protocol is suitable for modern large-scale wireless sensor networks.

To show the applicability of the synchronization protocol in more general mobile networks, we tested our algorithm on a mobile network using the random waypoint mobility model [9, 10, 84], which is the most used mobility model in the networking community. Figure 4.7 illustrates the operating principle of the random waypoint mobility model. At startup, each mobile node selects a new destination position at random and moves at constant speed following a straight line. Once a node reaches the target position, it remains there for a predetermined pause time. When the pause time has elapsed, the node selects a new position and repeats the process.

The network under study is formed by 9 nodes, whose initial distribution and interaction topology is shown in Figure 4.8. All the nodes were given the same transmission range of $250m$. For the first set of simulations we fix nodes 5, 6, and 7; while the others are free to move following the random waypoint mobility model. Table 4.9 presents the time to synchronization averaged over 100 runs when

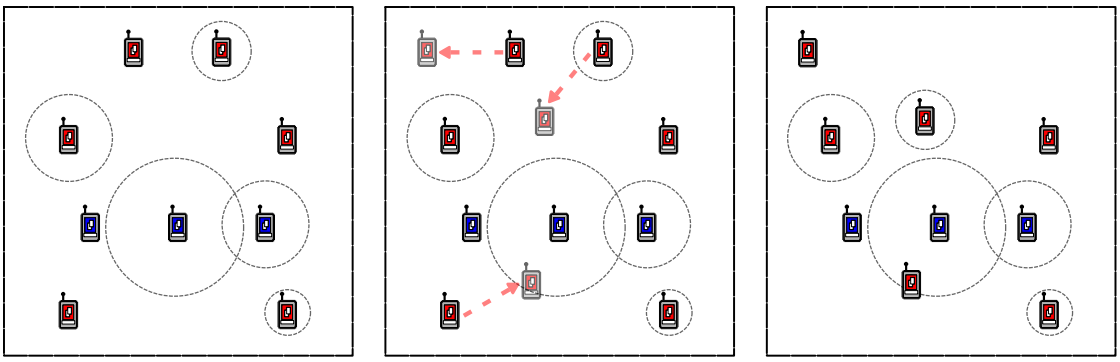


Figure 4.7: Illustration of the random waypoint mobility model for a 9-node mobile network. At a given time instant (left), mobile nodes (in red) pick a destination at random (middle) and then move to the destination at constant speed following a straight line (right). The process is repeated after a predetermined pause interval.

the length of the refractory period varies. It can be seen that the length of the refractory period has no effect on the time to synchronization.

For the second set of simulations, we start with the positions given in Figure 4.8 and consider each sensor static. We then enable mobility to each sensor one by one until all 9 sensors are able to move following the random waypoint mobility model. Table 4.10 shows the time to synchronization averaged over 100 runs for each case. It can be seen that as the number of mobile nodes increases, the time to synchronization decreases, which suggests that mobility facilitates the synchronization process. It should be noted, however, that the nodes distribution resulting from the random waypoint mobility model is biased towards the center of the deployment region [10, 84], which implies that mobile nodes tend to group in the center and create a highly connected network.

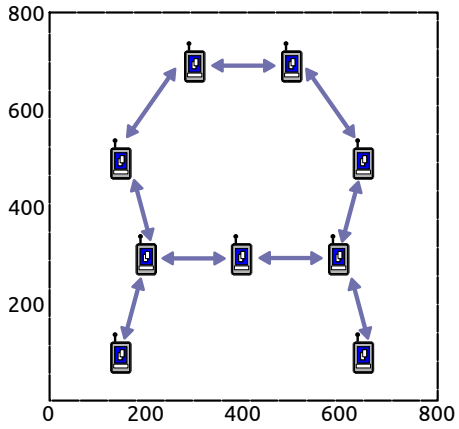


Figure 4.8: Mobile network used in the numerical simulations and the initial communication topology.

Table 4.9: Influence of the refractory period on the time to synchronization $[s]$ for the mobile network in Figure 4.8.

Refractory period l	0.2π	0.4π	0.6π	0.8π	1.0π	1.2π
Time to synchronization $[s]$	142.27	142.25	142.25	142.25	142.25	142.25

Table 4.10: Influence of mobility on the time to synchronization $[s]$ for the mobile network in Figure 4.8.

Number of mobile nodes	1	2	3	4	5	6	7	8	9
Time to synchronization $[s]$	290.91	279.73	260.30	244.18	175.46	174.52	174.52	153.56	136.10

4.4.3 Decentralized Synchronization over an Unreliable Network

The assumption of reliable communication at all times is somewhat difficult to achieve in a real deployment of a sensor network. To test our protocol in a more realistic scenario, we conducted simulations over an unreliable wireless network where each transmission has an associated probability of successful delivery p. For these simulations, we also choose initial conditions from a uniform distribution in the interval $[0, 2\pi]$. No initial flooding was carried out; therefore, conditions in Theorem 4.2 hold and we will be looking at the probability of synchronization in a variety of scenarios depending on the number of in-neighbors $|\mathcal{N}^{i-}|$, the value of p, and the length of the refractory period r.

For all the following simulations, we used the static grid network of 121 nodes shown in Figure 4.9. Additionally, we consider 79 mobile nodes whose initial positions are randomly chosen from a uniform distribution. Considering that each node is equipped with an omnidirectional antenna with a transmission range of $40m$, the probability of synchronization is larger than 0.999.

For the first set of simulations, we consider a wireless network with $p = 1$, i.e., reliable communication. Table 4.11 presents the simulation results averaged over 10000 runs when the length of the refractory period is varied from 0.1π to 0.5π.

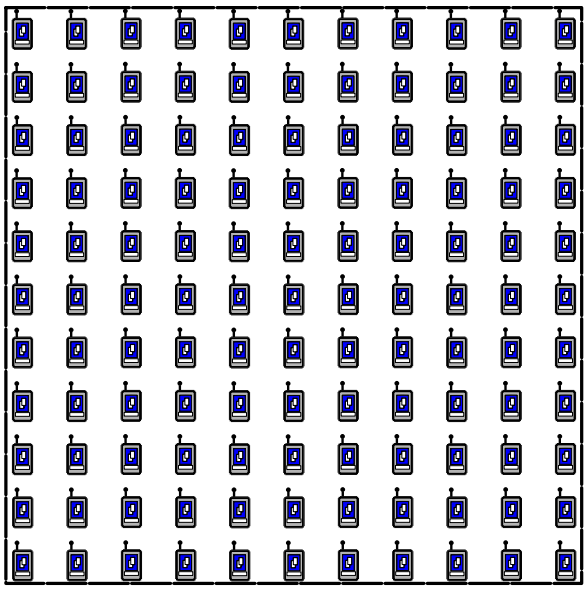


Figure 4.9: Grid static network used in the simulations over an unreliable network. Nodes are evenly placed keeping a distance of $10m$ with their closest neighbor in both horizontal and vertical directions.

Table 4.11: Influence of the refractory period on the probability of synchronization.

Refractory period r	0.1π	0.2π	0.3π	0.4π	0.5π
Fraction of synchronized runs	1.00	1.00	1.00	1.00	1.00

Results show that the length of the refractory period mildly affects the probability of synchronization, as predicted by Theorem 4.2.

For the second set of simulations, we consider an unreliable network with $p = 0.9$. Table 4.12 presents the simulation results averaged over 10000 runs when the length of the refractory period varies from 0.1π to 0.5π. Results show that the

Table 4.12: Influence of the refractory period on the probability of synchronization for an unreliable network with $p = 0.9$.

Refractory period r	0.1π	0.15π	0.2π	0.25π	0.3π
Fraction of synchronized runs	0.9987	0.9987	0.9981	0.9982	0.9985

Table 4.13: Probability and time to synchronization for different link reliabilities.

Probability p	0.9	0.85	0.8	0.75	0.7
Fraction of synchronized runs	1.0	1.0	1.0	1.0	1.0
Time to synchronization $[s]$	7.75	12.96	18.57	23.97	44.74

length of the refractory period mildly affects the probability of synchronization despite the network being unreliable.

For the following simulations, we removed the 79 mobile nodes and then the network under analysis is the 121-node grid network in Figure 4.9. We first study the influence of link reliability on the probability and time to synchronization. Table 4.13 presents the simulation results averaged over 1000 runs. It can be seen that the network synchronizes in all cases, yet the time it takes to reach synchronization increases with a decrease in the link reliabilities.

To illustrate the effect of the number of neighbors on the probability and time to synchronization, we vary the transmission range of the nodes. Table 4.14 shows the simulations results averaged over 10000 runs when the transmission range varies from $36.1m$ to $72.2m$. The results show that as the transmission range is increased, the network synchronizes faster.

Table 4.14: Time to synchronization for different transmission ranges and network indegrees.

Transmission range $[m]$	36.1	45.1	54.1	63.1	72.2
Indegree	13	22	30	30	49
Fraction of synchronized runs	1.0	1.0	1.0	1.0	1.0
Time to synchronization $[s]$	3.27	3.08	2.34	2.15	1.82

Table 4.15: Time to synchronization for different network sizes.

Network size	11×11	10×10	9×9	8×8	7×7	6×6
Fraction of synchronized runs	1.0	1.0	1.0	1.0	1.0	1.0
Time to synchronization $[s]$	3.27	2.48	1.67	1.21	1.06	0.96

Finally, we explore the effect of the network size on the probability and time to synchronization. Table 4.15 presents the simulation results averaged over 10000 runs when the network size is varied from 121 to 36 nodes. It can be seen that the network synchronizes faster as the size of the network is reduced.

The simulations presented indicate that the protocol behaves consistently with the available theoretical results. Moreover, for cases when the assumptions do not hold, the PCO-based protocol still synchronizes the network. Motivated by the simulation results, we implemented the protocol in real hardware to test the performance in sensors.

4.5 Pilot Implementation in Gumstix Overo Development Boards

To test the performance of the protocol in a real environment, we implemented the protocol in a testbed consisting of 3 Gumstix development boards [39]. The Gumstix Overo AirSTORM computer-on-module board is based on the Texas Instruments Sitara AM3703 Processor 800MHz ARM Cortex-A8 microprocessor. It is able to run several Linux distributions and it has an integrated 2×2 wireless 802.11-Bluetooth chipset. It is 17mm x 58mm x 4.2mm in size and its power consumption is typically less than 1W at full operation. Figure 4.10 shows the boards used in the evaluation.

The protocol was coded in user space at the application layer using a phase counter that is incremented at constant frequency. Pulses are simulated using UDP datagrams that are broadcasted after a counter overflow event. The source code used in the implementation of the protocol in the Gumstix boards is given in Appendix B.

Our objective is to use the pulse-coupled protocol to synchronize a 3-board network, while achieving a collective period of $1s$. Since in a real implementation there are processing and transmission delays, the natural period needs to be reduced to

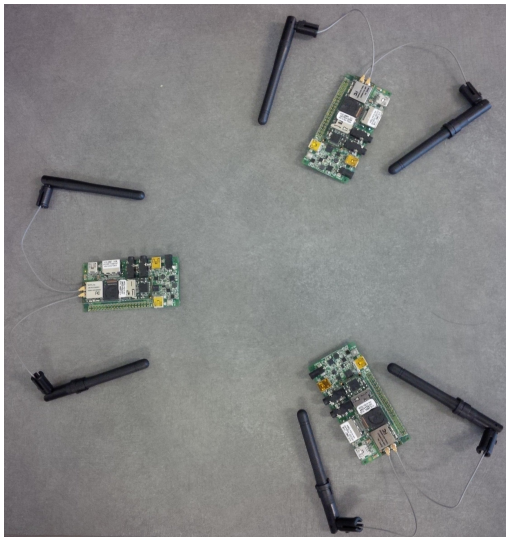


Figure 4.10: Gumstix Airstorm Overo development boards used in the pilot evaluation.

compensate them. We found experimentally that a natural period of $0.998s$ yields a collective period of $1s$ when the boards are coupled to form a network.

The stability of the natural period is an issue in a real hardware implementation since jitter is always present. In order to characterize the jitter present in the natural period, we run the boards uncoupled for approximately 1 hour and record the firing times to calculate the natural period. Results are given in Figure 4.11(a) where it can be seen that over 80% of the samples are within $\pm 0.125ms$ of the targeted natural period $T = 0.998s$.

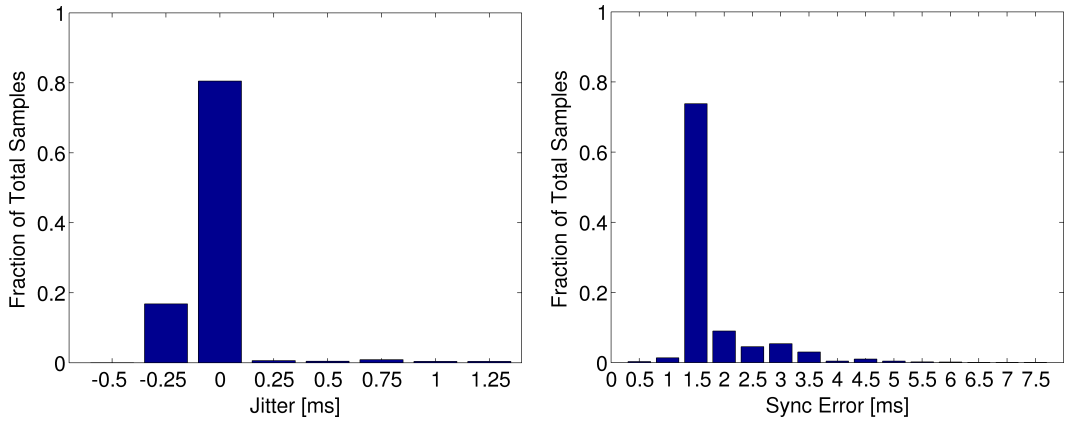


Figure 4.11: Results of the implementation in Gumstix boards. (a): Jitter present in the free running period; (b): Synchronization error for a 3-board all-to-all network.

Table 4.16 details the natural period experiments for the 3 boards. Results show an identical average natural period for the three boards up to the microsecond level and a standard deviation of less than 0.135 ms.

To evaluate the synchronization accuracy achieved in a 3-board all-to-all network, we run the coupled system for approximately 1 hour and record the firing times of each board. Comparing the firing times gives the actual skew in the network. Figure 4.11(b) presents the histogram of the recorded network skew. It can be seen that over 70% or the recorded samples are below $1.75ms$ with an average global skew of $1.72ms$. It is important to note that, although we consider the global skew acceptable, the results are significantly larger than the simulation results obtained in the previous section (*cf.* Table 4.8). We believe that the reason is

Table 4.16: Analysis of the natural period for the three Gumstix boards used in the implementation.

	Average Period $[s]$	Standard Deviation $[ms]$	Total Samples
Node A	0.99789	0.1317	3600
Node B	0.99789	0.1345	3592
Node C	0.99789	0.1322	3614

the (non-deterministic) processing time taken from reception to the actual update of the counter.

Table 4.17 details the collective period results for the 3-board all-to-all network. Results show an identical average collective period for the three boards up to the hundreds of microseconds level and a standard deviation of less than 1.3 ms. It should be noted that, although the average collective period is close to the target $T = 1s$, the standard deviation is significantly larger than the one obtained in the uncoupled case. A possible cause is the inevitable drop of some pulses. In fact, it can be seen in the fourth column in Table 4.17 that the number of received pulses for a given board is not equal to the sum of packets sent by the other two boards. This implies that some pulses were lost during the synchronization process thus affecting the collective period since in practice packet losses mean no coupling.

With our available testbed of 3 boards, besides the simplest all-to-all network we can also obtain a line network. We placed the boards to obtain a line communication topology (board A-board B-board C) and run the algorithm for approxi-

Table 4.17: Analysis of the collective period for a 3-board all-to-all network.

	Average Period $[s]$	Standard Deviation $[ms]$	Pulses sent / received
Node A	1.000313	1.2392	3509 / 6369
Node B	1.000311	1.2302	3501 / 6402
Node C	1.000305	1.2872	3488 / 6278

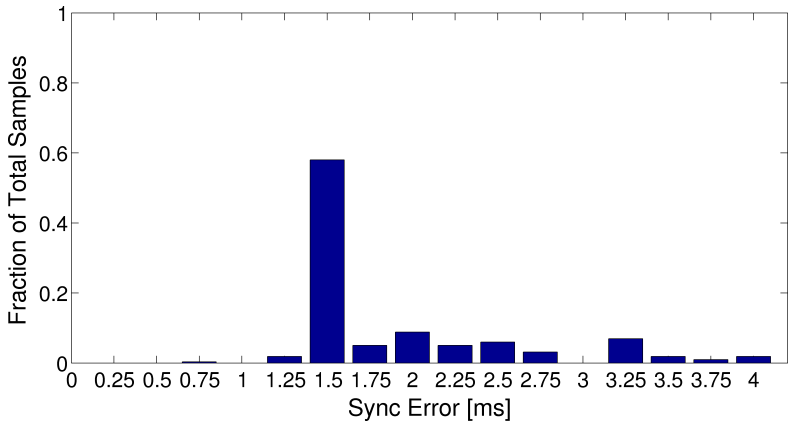


Figure 4.12: Synchronization error for a 3-board line network.

mately 1.5 hours and record the firing times of each board. Comparing the firing times gives the actual skew in the network. Figure 4.12 presents the histogram of the recorded network skew. It can be seen that over 70% or the recorded samples are below $1.75ms$ with an average global skew of $1.89ms$. It should be noted that the line topology is affected by the hidden terminal problem [93], which affects the stability of the collective period, as well as the network skew.

Table 4.18 details the collective period results for the 3-board line network. Results show an identical average collective period for the three boards up to the

Table 4.18: Analysis of the collective period for a 3-board line network.

	Average Period $[s]$	Standard Deviation $[ms]$	Pulses sent / received
Node A	0.999356	1.1078	5330 / 4790
Node B	0.999353	1.5704	5257 / 4353
Node C	0.999361	1.0539	5325 / 5164

hundreds of microseconds level and a standard deviation of less than 1.11 ms for the outside boards and less than 1.6 ms for the board in the middle. It should be noted that, although the average collective period is close to the target $T = 1s$, the standard deviation is significantly larger than the one obtained in the uncoupled case. The hidden terminal problem is responsible for the higher standard deviation in the middle board and the larger difference from the target period with respect to the all-to-all case. In fact, it can be seen in the fourth column in Table 4.18 that the number of received pulses for the middle board is much lower than the sum of packets sent by the other two boards, which is a direct effect of the hidden terminal problem [93]. The results obtained for the line topology show that in practice the communication topology impacts the performance of the algorithm, hence care should be taken when designing the network to avoid complicated topologies such as those susceptible to the hidden terminal problem.

To improve the performance of the algorithm, the implementation should be done at a level lower than the user space. For future implementations, we propose

to develop a kernel time synchronization module to embed the synchronization protocol in the operating system to have access to more precise time management functions and to reduce the processing delay present at the user space.

4.6 Pilot Implementation in BMS Acoustic Sensors

To test the performance of our algorithm in a real sensor network, we implemented the algorithm in a network of acoustic sensors provided by BioMimetic Systems, Inc. [11]. The BioMimetic Systems team has developed a range of advanced, biologically-inspired acoustic sensor systems for small arms fire detection, identification, and localization. The core of the Smart Neural Acoustic Processor (SNAP) technology has been developed over more than a decade of cooperative research between BMS, Boston University Hearing Research Center, U.S. Army ARDEC and ARL, DARPA, Office of Naval Research (ONR), and Joint Ground Robotics Enterprise (JGRE). The SNAP family of sensors is small, lightweight, low power devices easily configured for various array geometries, platforms, and acoustic targets. Figure 4.13 shows an schematic of the BMS sensors used in the evaluation of the algorithm.

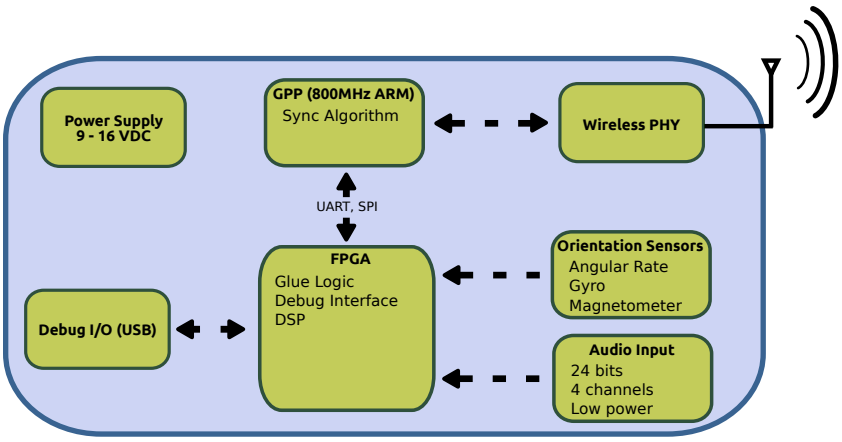


Figure 4.13: Schematic of the BMS acoustic sensors used in the pilot evaluation.

The objective of the tests is to study the performance of the PCO-protocol under real operating conditions where the acoustic sensors are acquiring and processing acoustic signals, orientation information, and run the synchronization algorithm concurrently. To this end, the BMS team performed tests in a firing range in two different scenarios. In the first setup, a single shooter fires single shots periodically and hence the sensors are subject to an average processing load. In the second setup, multiple shooters fire single and multiple shots and hence the sensors are subject to a severe processing load.

In the first test, we run the sensors uncoupled to study the free running period in both scenarios: average and severe processing load. Figure 4.15 shows the observed jitter present in the free running period. It can be seen that under an average processing load (Figure 4.14(a)) approximately 70% of the samples are within

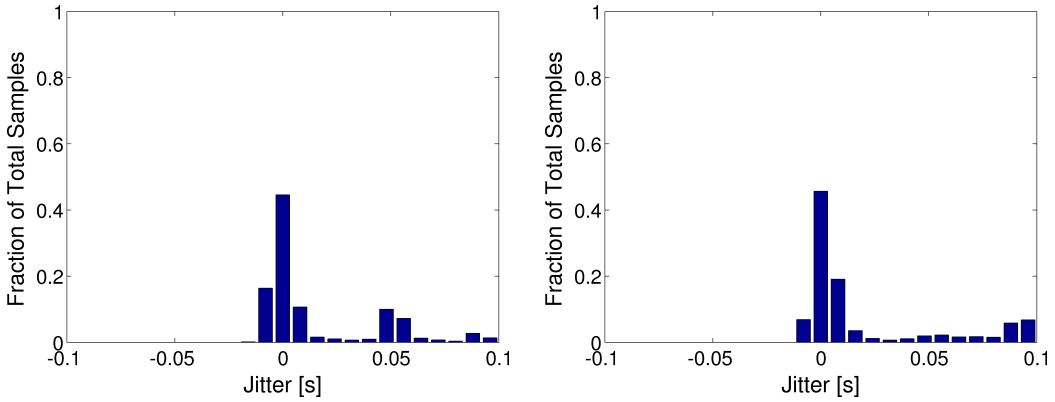


Figure 4.14: Results of the implementation in BMS sensors. (a): Jitter present in the free running period under average processing load; (b): Jitter present in the free running period under severe processing load.

$\pm 8ms$ of the target period, which is acceptable yet much higher than the results obtained in the Gumstix boards (*cf.* Figure 4.11(a)). When the sensors are faced with a severe processing load (Figure 4.14(b)), we also obtained approximately 70% of the samples within $\pm 8ms$ of the target period; however, the proportion of samples in the higher end ($100ms$) greatly increases with an increase in the processing load. Increasing the processing load increases the jitter since the PCO-algorithm is not given the proper amount of resources for correct operation.

In the second test, we study the synchronization error of a 4-sensors all-to-all network in both scenarios: average and severe processing load. Figure 4.15 shows the results. Under an average processing load (Figure 4.15(a)) approximately 50% of the samples are below $6.75ms$ while less than 10% are above $10ms$. Note that the network skew is much higher than the results obtained in the Gumstix boards (*cf.*

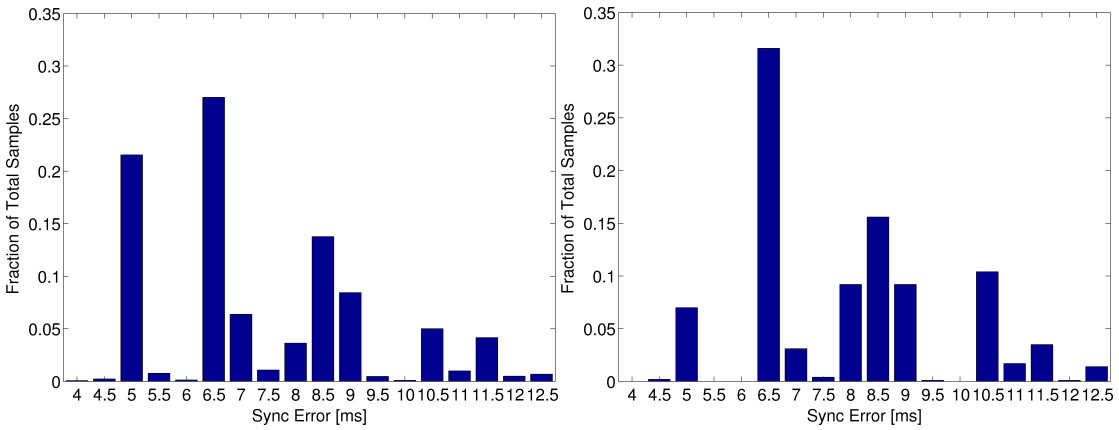


Figure 4.15: Synchronization results of the implementation in BMS sensors. (a): Synchronization error for a 4-sensors all-to-all network under average processing load; (b): Synchronization error for a 4-sensors all-to-all network under severe processing load.

Figure 4.11(b)). When the processing load is severe (Figure 4.15(b)), the samples below $6.75ms$ reduce to less than 40% while the samples above $10ms$ increase to more than 20%. As it was expected, based on the jitter results, a higher processing load increases the network skew.

The fact that a high processing load in the sensors damages the performance of the synchronization algorithm supports our plan to develop a kernel time synchronization module, to embed the synchronization protocol in the operating system thus ensuring the proper amount of resources.

The favorable results obtained motivates us to implement the PCO-protocol on a real practical system. As an example, in the next Chapter we implement the PCO-protocol on an acoustic event detection system.

Chapter 5

Pulse-Coupled Time Synchronization of Acoustic Event Detection Systems based on Mobile Wireless Sensor Networks

In this Chapter we provide an application example of our PCO-based synchronization protocol to acoustic event detection systems based on wireless sensor networks, whose localization accuracy dramatically deteriorates if the synchronization error is high.

5.1 Motivation

Localization systems based on fusing information from a collection of sensors have captured the attention of researchers due to their simple yet powerful operating principles. In sensor fusion localization algorithms, the first element is the data

Portions of this Chapter have been submitted for publication to IEEE Access [73].

association, where precise timestamps are required to ensure accurate localization. This poses the challenge of having an accurate common time notion among sensors. The concept of precision, however, is application-dependent and can vary from less than a microsecond up to seconds. Recently, several synchronization algorithms for sensor networks have been designed to provide a general framework for synchronization. The packet-based synchronization strategies (*cf.* Chapter 4) such as RBS [28], TPSN [32], FTSP [56], GTSP [89], PulseSync [53], and Glossy [30], have been recognized as powerful alternatives for performing periodic time synchronization in sensor networks and currently comprise the state-of-the-art standards, despite the existence of well known drawbacks (*cf.* Chapter 4). An appealing alternative approach to periodic time synchronization is *post facto* synchronization [80], in which the network synchronizes after a significant event has occurred thus reducing the network traffic needed in traditional periodic synchronization strategies. However, *post facto* synchronization requires either the existence of a third party leader node, or the existence and maintaining of a skew table and a routing protocol [80]. These extra requirements on both the network and the processing capabilities of the sensor nodes make the scalability of *post facto* synchronization difficult in an unreliable network of low capability nodes. For these reasons, pulse-coupled synchronization appears to be the natural option for performing time synchronization in an unreliable wireless sensor network.

Although pulse-coupled synchronization was first introduced concurrently with packet-based strategies, only recently has its applicability been explored in detail, since the packet-exchanging nature of communication networks facilitates the application of packet-based synchronization protocols. Nonetheless, the progress in radio technology and network standards has made pulse-coupled synchronization feasible to implement using ultra-wide bandwidth pulses, or preambles in a IEEE 802.11 network [101]. Moreover, by transmitting simple identical pulses instead of full length packet messages, the pulse-based synchronization strategy eliminates the imprecision due to high stack layer delays, protocol processing, or software implementation arising in traditional packet-based synchronization strategies. In addition, pulse-coupled synchronization considers each received pulse identically, since exchanged pulses are independent of their origin [43, 76], and is inherently a distributed strategy that does not require the selection of a root, or leader, node to flood the network with its local time. In contrast, in pulse-coupled synchronization, the common time of the network is agreed by all the participating nodes via simple local interactions, which makes synchronization robust to disconnections of any node. Despite the recent re-emergence of pulse-coupled synchronization, which has motivated both analytical- [100, 102, 74, 69, 70, 71] and general testbed-based [43, 76, 94] studies, an application of this technique to a practical functional system has not been reported yet.

In this Chapter, we present the application of pulse-coupled synchronization to an acoustic event detection system designed to locate the source of an acoustic event in a two-dimensional space, by using an acoustic-capable wireless sensor network. The particular acoustic event detection system of interest is inspired by military applications on locating sources of gunfire or explosions by soldier worn acoustic sensors [97, 17, 33, 81]. Initially, solutions were based on local measurements taken by an array of microphones, which allowed the system to locate the source based on angle-of-arrival (AoA) measurements. The appearance of sensor networks enabled a networked solution where localization is carried out using sensor fusion techniques. However, this poses the extra requirement of having a common time reference. A variety of approaches to tackle both sensor fusion and time synchronization have been investigated. The system in [97] is formed by a network of multi-channel sensors able to gather time-of-arrival (ToA) as well as AoA measurements. Fusion is conducted by a central node and time synchronization is achieved by using the *post facto* strategy given in [80]. A key assumption is that acoustic events are sporadic, and thus *post facto* synchronization allows resource savings by synchronizing the network only after an event has occurred, rather than keeping the network in sync all the time. Alternatively, the system presented in [81] uses single channel sensors that gather ToA measurements and fuses them in a central unit. It is assumed that each sensor is equipped with a GPS receiver and

then precise time synchronization is available; the main drawback of this approach is that the GPS signal needs to be accessible at all times. In this Chapter, localization is performed using ToA measurements from single channel sensors, while the time synchronization problem is solved using pulse-coupled synchronization.

The particular acoustic event detection system under consideration is a patrolling squadron, meaning that sensors are free to move yet they try to maintain a given geometric formation at all times. The system is able to implement a variety of formations and it performs localization using sensor fusion algorithms based on ToA measurements. Two approaches are proposed to solve the localization problem, a standard centralized estimator that fuses ToA measurements from all sensors at a central node, and a novel distributed approach where each sensor in the network solves a reduced localization problem using only a subset of the ToA measurements, and then the estimations are fused by means of distributed average consensus algorithms. The novel distributed approach presented is in line with our general aim to construct a fully distributed system where both synchronization and localization are achieved by means of simple interactions between neighboring sensors. To enable communication in the network, for both synchronization and measurement sharing, we propose a pure-broadcasting infrastructure-free ah-hoc network, for which pulse-coupled time synchronization is the natural choice.

5.2 Sensor Network based Acoustic Event Detection Systems

5.2.1 Sensor Network and Formations

The acoustic event detection system is comprised of N identical independent agents, distributed according to a given configuration, equipped with an acoustic sensor (microphone), a central processing unit, and a wireless transceiver in charge of establishing communication between sensors. For simplicity, we consider that the agents live in a two-dimensional Euclidean space, i.e., $\mathbb{R}^2$. The agents act as a patrolling squadron, meaning that they are free to move yet they try to maintain a geometric formation at all times, for security, tactical, operational or technical reasons. Formation maintaining control algorithms are under active study and are out of the scope of this work; we will assume that the agents are able to maintain their formation at all times up to a bounded error.

Throughout this Chapter, a graph theoretical formulation of the sensor network is used to characterize both the geometrical deployment (or formation) and the communication topology. To this end, we define the set of nodes (or agents) as $\mathcal{V} = \{1, \ldots, N\}$. The formation is modeled as a weighted undirected graph $\mathcal{F} = \{\mathcal{V}, \mathcal{E}_\mathcal{F}, \mathcal{A}_\mathcal{F}\}$, where the edge set is defined as $(i,j), (j,i) \in \mathcal{E}_\mathcal{F} \subseteq \mathcal{V} \times \mathcal{V}$ if and only if there is a distance constraint between nodes i and j. And the weighted

adjacency matrix $\mathcal{A}_{\mathcal{F}} = [a_{ij}] \in \mathbb{R}_{\geq 0}^{N \times N}$ is such that $a_{ij} > 0$ if and only if $(i, j) \in \mathcal{E}_{\mathcal{F}}$ and 0 elsewhere. In fact, if $a_{ij} > 0$, it is equal to the value of the distance between nodes i and j. Similarly, since we will consider identical transmission range, hence bidirectional communication, the communication topology is also modeled as an undirected graph $\bar{\mathcal{R}} = \{\mathcal{V}, \bar{\mathcal{E}}_{\mathcal{R}}, \bar{\mathcal{A}}_{\mathcal{R}}\}$, whose adjacency matrix $\bar{\mathcal{A}}_{\mathcal{R}} = [a_{ij}]$ is such that $a_{ij} = 1$ if and only if $(i, j) \in \bar{\mathcal{E}}_{\mathcal{R}}$, $a_{ii} = 1$ (self-loops), and 0 elsewhere. The reader is referred to [35] and [4] for an extensive treatment of the graph theoretical formulation used.

Note that the localization capabilities (considering perfect synchronization) are closely related to the geometrical deployment or formation graph, as will be clear in the following section, yet synchronization properties depend on the underlying communication graph induced by the formation and a given transmission range. We will analyze the interplay between these two graphs when estimating the location of the acoustic source.

In the following, let $p_i \in \Theta \subset \mathbb{R}^2$ be the position of sensor $i \in \mathcal{V}$, where Θ is the region of the space we wish to monitor. Consequently, let $P := [p_1 \dots p_N]^T$ be the position vector of the sensor network. We will say that the sensor network is in formation $\mathcal{F}$ if P satisfies all the distance constraints given in $\mathcal{A}_{\mathcal{F}}$, and we write $P \sim \mathcal{F}$. The characteristics of $\mathcal{F}$ and the number of sensors N play a key role in obtaining an accurate estimation of the location of the acoustic source. It has

been proven in [108] that ToA based localization systems can locate a source in $\mathbb{R}^2$ if and only if the sensors do not lie on a hyperbola. Moreover, the number of sensors required to perform single event localization is between 4 and 6. The work [95] extended the analysis for localization of simultaneous events finding analytically that a number of 9 sensors is sufficient for correct localization; nonetheless, it is experimentally conjectured that 8 sensors are enough to solve the problem. Regarding formations, [12] has identified conditions for optimal sensors' deployment when the position of the source is known, while [46] extended the optimal placement strategy when the source's position is unknown but a probability distribution for the position is available. However, when the sensors are placed in an optimal deployment the source lies in the convex hull of the sensor network, which is not realistic in the case we are interested since patrolling squads aim to locate events occurring outside its convex hull. This justifies our decision to not employ known optimal deployment strategies. The following standing assumption describes the characteristics of the formations used in this work, which ensures that the patrolling squad is able to locate the source uniquely at all times.

Assumption 5 *The formation graph $\mathcal{F}$ is such that if $P \sim \mathcal{F}$, then the position vector P defines an identifying sensor set (ISS) [108], i.e., no hyperbola in $\mathbb{R}^2$ passes through all components of P. Moreover, $\mathcal{F}$ is a globally rigid graph [4], i.e.,*

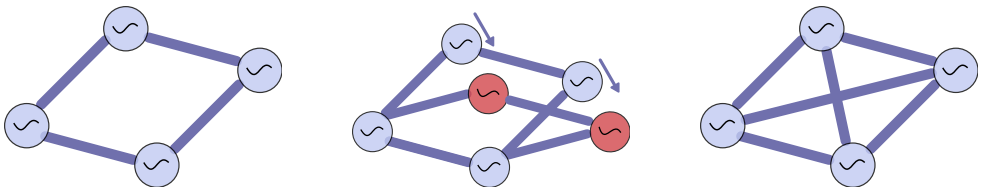


Figure 5.1: Illustration of the concept of a globally rigid graph (links represent the existence of a distance constraint); (a): a 4-node graph that is not rigid since it can be deformed smoothly to obtain the structure shown in (b); (c): a 4-node rigid graph that cannot be deformed.

if two generic position vectors are such that $P_1 \sim \mathcal{F}$ *and* $P_2 \sim \mathcal{F}$*, then* P_1 *and* P_2 *differ only by a combination of translation, rotation, and reflection in* $\mathbb{R}^2$*.*

Figure 5.1 illustrates the concept of a globally rigid graph. Links represent the existence of a distance constraint. The graph in Figure 5.1(a) is not rigid since it can be deformed without violating any distance constraint to obtain the structure shown in Figure 5.1(b). On the other hand, Figure 5.1(c) shows a globally rigid graph that cannot be deformed without violating a distance constraint. Hence, if two generic position vectors satisfy the distance constraints for Figure 5.1(c), they differ only by a combination of translation, rotation, and reflection in $\mathbb{R}^2$.

5.2.2 Centralized Localization

We focus on the localization of the source of acoustic events using as the primary information variable the time-of-arrival (ToA) of the blast wave, emitted by an acoustic source located at $\mathbf{S} \in \mathbb{R}^2$, to the sensor network. The strategy is to utilize

the arrival of the blast to estimate the relative position of the acoustic source with respect to the sensor network by combining measurements from the different sensors. Figure 5.2 shows a schematic of the situation, where an acoustic source emits a blast wave that propagates spherically at the speed of sound. The measured ToA of the blast, originated at $\mathbf{S}$ at a given time t_0, at a sensor i, located at p_i with radial distance r to the acoustic source, is given by:

$$\mathrm{ToA}_i = t_0 + \frac{r}{c} + \eta_i = t_0 + \frac{d(\mathbf{S}, p_i)}{c} + \eta_i \tag{5.1}$$

where c is the speed of sound, η_i is the measurement noise (accounting for the synchronization error between sensors and assumed to be a zero-mean white noise with variance σ^2), and $d(x, y) := ||x-y||_2$ is the Euclidean distance in $\mathbb{R}^2$. In order to accurately determine the position of the acoustic source $\mathbf{S}$, the measurement of a single sensor is insufficient, thus localization has to be performed by combining the measurements of the whole sensor network. Since the time at which the blast originated, t_0, is irrelevant, and furthermore it represents an additional unknown variable, localization can be performed by considering the time-difference-of-arrival (TDoA) between sensor i and a chosen reference sensor $r_f \in \mathcal{V}$ defined as:

$$\mathrm{TDoA}_i = \mathrm{ToA}_i - \mathrm{ToA}_{r_f} = \frac{d(\mathbf{S}, p_i) - d(\mathbf{S}, p_{r_f})}{c} + \eta_{ir_f} \tag{5.2}$$

where η_{ir_f} is the zero-mean difference measurement noise with variance $2\sigma^2$. Choosing, w.l.o.g., sensor 1 as reference, the maximum likelihood estimator (MLE) for the position of the acoustic source is given by [77]:

$$\hat{\mathbf{S}} = \arg\min_{z\in\Theta} \sum_{i=2}^{N} \left(\mathrm{TDoA}_i - \left(\frac{d(z,p_i)}{c} - \frac{d(z,p_1)}{c} \right) \right)^2 \tag{5.3}$$

which corresponds to the least squares minimization of η_{ir_f}. The optimization problem (5.3) is nonlinear and nonconvex, which makes it difficult to find the optimal solution. Several approaches for efficiently solving the MLE problem (5.3) have been developed with variable potential. An example is the second-order cone programming (SOCP) relaxation [109], which by introducing auxiliary variables and relaxing equality constrains is able to transform (5.3) into a quadratic convex problem. However, this approach suffers from a restrictive convex hull problem, i.e., the optimal solution lies within the convex hull of the sensors, which is not a problem when the sensors are deployed following an optimal configuration in the sense of [12, 46]; yet it is a limitation in the general case. An efficient approach that can avoid the convex hull problem is semidefinite program (SDP) relaxation, which has been proven to be effective in solving (5.3), even when there is uncertainty in the sensors' positions [109]. Recently, a two-step least squares algorithm, also based on semidefinite relaxation, was proposed to solve the localization problem

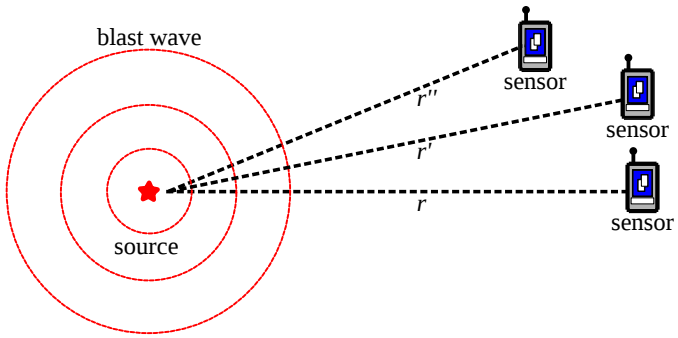


Figure 5.2: Propagation of the blast wave from the acoustic source to the sensor network. The blast is assumed to be propagating at the speed of sound c.

using directly the ToA measurements [107] to avoid the correlated noise terms that appear after subtracting the ToA measurements to obtain the TDoAs [95].

In practice, however, there is an unavoidable uncertainty associated with the position p_i, due to the intrinsic difficulty of maintaining a tight formation given that measurement/localization systems are inaccurate. Therefore, position uncertainties should be included in the formulation of (5.3) and then the estimator becomes:

$$\hat{\mathbf{S}} = \arg\min_{z\in\Theta} \sum_{i=2}^{N} \left(\mathrm{TDoA}_i - \left(\frac{d(z,\hat{p}_i)}{c} - \frac{d(z,\hat{p}_1)}{c} \right) \right)^2 \tag{5.4}$$

where $\hat{p}_i := p_i + \epsilon^i$ is the estimated position of node i with $\epsilon^i \in [-\bar{p}, \bar{p}]$ being the bounded position error.

New centralized sensor fusion techniques for solving the acoustic source localization problem using extra measurements are currently under active development [17, 33, 81, 38], however, they are out of the scope of this study.

5.2.3 Distributed Localization

In the previous section, the solution of the full information localization problem was stated. It should be noted that in the full information problem, the underlying communication network has no apparent effect since every sensor node needs to know all the ToA measurements, which implies some sort of flooding or all-to-all communication. However, in a fully distributed system a given sensor node only counts with the subset of ToA measurements gathered by its neighbor nodes $\mathcal{N}_i$ and flooding the network should be avoided. Hence, the communication network will play a crucial role in the distributed problem. In the rest of this section, we present the distributed localization problem where every sensor solves a local information problem and then local solutions are combined by means of averaging to obtain the global solution.

Based on the derivations made in the previous section, we can write the estimator for node i as:

$$\bar{\mathbf{S}}_i = \arg\min_{z\in\Theta} \sum_{j\in\mathcal{N}_i} \left(\mathrm{TDoA}_j - \left(\frac{d(z,\hat{p}_j)}{c} - \frac{d(z,\hat{p}_i)}{c} \right) \right)^2 \tag{5.5}$$

Note that (5.5) only uses information from neighbor nodes to estimate the location of the source. To ensure feasibility of the solution, we make the following assumption on the communication network.

Assumption 6 *For every node $i \in \mathcal{V}$ such that $P \sim \mathcal{F}$, the position vector corresponding to the sub-formation with nodes $\{i\} \cup \mathcal{N}_i$ defines an identifying sensor set. Consequently, $|\mathcal{N}_i| \geq 3$ holds for all i in $\mathcal{V}$ [108].*

Remark 5.1 *Note that if $\mathcal{F}$ is such that when $P \sim \mathcal{F}$, P defines an identifying sensor set, then for all i there always exists a set $\mathcal{N}_i \subset \mathcal{V}$, such that the sub-formation with nodes $\{i\} \cup \mathcal{N}_i$ defines an identifying sensor set. Clearly, this holds trivially when $\mathcal{N}_i = \mathcal{V} \setminus \{i\}$.*

Moreover, to ensure that the distributed averaging algorithm presented in the following converges properly, we make the following assumptions.

Assumption 7 *The communication delay is small enough to ensure that data transmitted by a node is received by all its destination nodes before any other node starts a new transmission. Moreover, there exists a positive finite T such that in a time interval of length T, every node broadcasts exactly once.*

Remark 5.2 *Note that the radio coverage of a sensor is usually short, hence Assumption 7 is not restrictive. Moreover, collision avoidance mechanisms will ensure that Assumption 7 holds.*

Assumption 8 *Every sensor $i \in \mathcal{V}$ knows its neighbor set $\mathcal{N}_i$ and, consequently, the number of neighbors it has $|\mathcal{N}_i|$.*

Remark 5.3 *Note that since the formation is known before deployment and, moreover, it plays a critical role in localization, it is reasonable to assume that, given a transmission range, every node knows how many nodes it can reach.*

In order to achieve a common estimate, nodes need to share and combine their local measurements. To this end, we consider a pure-broadcasting infrastructure-free ad-hoc network to establish communication. Using this network, each node will broadcast its local information every time a local timer x_i reaches a threshold value. Note that collision-avoidance mechanisms will ensure that only one node broadcasts at a given time. To average the local estimates $\bar{\mathbf{S}}_i$, we propose an averaging consensus algorithm inspired by asynchronous double linear iterations [54] that combines local estimates as follows.

Each node has two localization variables S_i and z_i, which initializes as $S_i = \bar{\mathbf{S}}_i$ and $z_i = 1$, and a timer variable x_i that is initialized as $x_i = 0$ and increased at a fixed rate. When the local timer reaches a threshold value $x_i = x_{\text{th}}$, node i updates its localization variables and broadcasts them. Upon reception, every node $j \in \mathcal{N}_i$ uses the received information to update its own localization variables. The update of the localization variables of the whole network, after an event $x_i = x_{\text{th}}$, is given

by the following updating law:

$$S_i^+ = \frac{S_i}{1 + |\mathcal{N}_i|} \tag{5.6a}$$

$$S_j^+ = S_j + a_{ij}\frac{S_i}{1 + |\mathcal{N}_i|} \tag{5.6b}$$

$$z_i^+ = \frac{z_i}{1 + |\mathcal{N}_i|} \tag{5.6c}$$

$$z_j^+ = z_j + a_{ij}\frac{z_i}{1 + |\mathcal{N}_i|} \tag{5.6d}$$

where a_{ij} is the corresponding entry of $\mathcal{A}_\mathcal{G}$. The local estimation of node i is given at any time instant by $\hat{\mathbf{S}}_i = \frac{S_i}{z_i}$. If every node updates its local estimate following the previous rule, the local estimations $\hat{\mathbf{S}}_i$ will converge to the average of the initial estimates $\bar{\mathbf{S}}_i$. To see this fact, we can model the system as a hybrid system and study its asymptotic behavior as follows.

Define three state variables per node S_i, z_i and x_i and stack them in a vector to obtain the vectors S, z and x, which contain the states of every node. Define the flow set as $\mathcal{C} := \mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R}^N_{\geq 0}$. The continuous time dynamics are given by:

$$\dot{S} = 0 \tag{5.7a}$$

$$\dot{z} = 0 \qquad (S, z, x) \in \mathcal{C} \tag{5.7b}$$

$$\dot{x} = w \tag{5.7c}$$

where $w > 0$ is the frequency of the timer x_i. Similarly, we define the jump set as $\mathcal{D}_i := \{(S, z, x) \in \mathcal{C} : x_i \geq x_{\text{th}}\}$, $\mathcal{D} := \cup_{i \in \mathcal{V}} \mathcal{D}_i$ and the jump map is given by $x_i^+ = 0$ and (5.6), which can be written in matrix form as:

$$S^+ = A_i S \tag{5.8a}$$

$$z^+ = A_i z \qquad (S, z, x) \in \mathcal{D}_i \tag{5.8b}$$

$$x^+ = (I_N - \text{diag}(e_i))x \tag{5.8c}$$

where e_i is the ith canonical vector and A_i is the $N \times N$ identity matrix with the ith column replaced by the ith column of the adjacency matrix $\mathcal{A}_\mathcal{G}$ scaled by $\frac{1}{1+|\mathcal{N}_i|}$. In the resulting system, the localization variables remain constant and the timer variables are incremented at a constant rate during flows, and once a timer variable reaches the limit, the system jumps, i.e., the localization variables are updated.

To show that the local estimates converge to the average, we only need to focus on the asymptotic behavior of the underlying discrete time dynamics, i.e., Equation (5.8), since during flows the localization variables remain unchanged.

At this point it is important to note that the A_i matrices are column stochastic, i.e., A_i has non-negative entries and column sum equal to 1, with positive diagonal entries. Moreover, from Assumption 7 we know that every node broadcasts once in a finite interval of time, in particular we can take $T = x_{\text{th}}$, and then we can analyze

the system after a complete round of broadcasts has taken place. In the following, let $\mathcal{P}_N$ be the set of permutations of the product of the N matrices A_i. Since both column stochastic and positive diagonal matrices are closed under multiplication, the elements of $\mathcal{P}_N$ are column stochastic matrices with positive diagonal entries. We can then write the update of the localization variables after N broadcasts, or equivalently, after T units of ordinary time as:

$$S^{+N} = P_i S \tag{5.9a}$$

$$z^{+N} = P_i z \tag{5.9b}$$

where $P_i \in \mathcal{P}_N$ and the $+N$ superscript refers to N updates (or after N jumps of the hybrid system). Since P_i corresponds to the product of N matrices A_i in arbitrary order, its underlying graph is the composition of the N graphs induced by the A_i matrices [19]. Since the composition graph contains the union of the edges of the individual A_i [19], then for every $P \in \mathcal{P}_N$ its underlying graph is connected. It is a well known fact from consensus theory that an infinite product of column stochastic matrices with positive diagonal and connected underlying graph converges exponentially fast to a matrix of the form $\gamma \mathbf{1}^T$ [19], irrespective of the order, although the value of γ might depend on the order. Then we have that

as time goes to infinity:

$$S \to \gamma \mathbf{1}^T \bar{\mathbf{S}} = \left(\sum_{i \in \mathcal{V}} \bar{\mathbf{S}}_i \right) \gamma \tag{5.10a}$$

$$z \to \gamma \mathbf{1}^T \mathbf{1} = N\gamma \tag{5.10b}$$

and hence for each sensor:

$$\frac{S_i}{z_i} = \frac{\gamma_i \sum_{i \in \mathcal{V}} \bar{\mathbf{S}}_i}{\gamma_i N} = \frac{1}{N} \sum_{i \in \mathcal{V}} \bar{S}_i \tag{5.11}$$

Since the convergence is asymptotic, in practice we need to define a tolerance ϵ to determine when the localization variables have reached their final value.

Remark 5.4 *It should be noted that solutions to hybrid systems, i.e., hybrid arcs, live in a hybrid time domain. Even though the time domain has not appeared explicitly in the previous analysis, its hybrid nature is important for the results. Hybrid time domains are a natural tool to deal with asynchronous systems that experience jumps at non-constant time intervals, as the one under study, since they keep track of both ordinary time and number of jumps independently.*

We can summarize the previous strategy in the form of a protocol that every node in the network implements upon detecting an acoustic event.

Protocol 5.1 *Upon detecting an acoustic event, each sensor i broadcasts its ToA and gathers ToA measurements from its neighbors, obtains $\bar{\mathbf{S}}_i$ by solving its local localization problem (5.5), initializes its localization variables as: $S_i = \bar{\mathbf{S}}_i$ and $z_i = 1$, and starts monitoring a timer variable x_i.*

1. *At each local timer event $x_i = x_{th}$, sensor i updates its localization variables using the rule:*

$$S_i^+ = \frac{S_i}{1 + |\mathcal{N}_i|} \tag{5.12a}$$

$$z_i^+ = \frac{z_i}{1 + |\mathcal{N}_i|} \tag{5.12b}$$

then, it broadcasts its updated values S_i^+ and z_i^+, and restarts the timer x_i

2. *Upon reception of a set of localization variables $\frac{S_j}{1+|\mathcal{N}_j|}$ and $\frac{z_j}{1+|\mathcal{N}_j|}$ from a neighboring node j, node i updates its localization variables using the rule:*

$$S_i^+ = S_i + \frac{S_j}{1 + |\mathcal{N}_j|} \tag{5.13a}$$

$$z_i^+ = z_i + \frac{z_j}{1 + |\mathcal{N}_j|} \tag{5.13b}$$

3. *When the tolerance is satisfied, i.e., $\frac{S_i^+}{z_i^+} - \frac{S_i}{z_i} < \epsilon$, i declares the problem solved and selects $\hat{\mathbf{S}}_i = \frac{S_i}{z_i}$ as the location of the acoustic source.*

Based on the previous analysis, the following Theorem is introduced.

Theorem 5.1 *Consider a sensor network based acoustic event detection system deployed in a given formation $\mathcal{F}$. Let $\hat{\mathbf{S}}_i$ be the local estimates and $\bar{\mathbf{S}}_i$ the initial local estimates obtained by solving (5.5). If Assumptions 6-8 hold and the nodes implement Protocol 5.1 upon detecting an acoustic event, then*

$$\hat{\mathbf{S}}_i \to \bar{\mathbf{S}}_{avg} := \frac{1}{N} \sum_{i \in \mathcal{V}} \bar{\mathbf{S}}_i, \quad \forall i \in \mathcal{V} \tag{5.14}$$

exponentially fast.

5.2.4 Performance Example

In this work, we consider a particular application of acoustic event detection systems to gunfire detection, taken from [17]. The acoustic localization of small arms' fire relies entirely on the sounds produced by the muzzle blast of the weapon [8]; hence, a TDoA-based approach is suitable. Figure 5.3 shows the four different formations that will be analyzed, each of them consisting of 8 sensors. $\mathcal{F}_1$ corresponds to a classical wedge formation [17]; $\mathcal{F}_2$ corresponds to a quad formation; $\mathcal{F}_3$ corresponds to a symmetric wedge formation; and $\mathcal{F}_4$ to a circle formation that is optimal in the sense of [12, 46] for sources inside the circle. It should be noted that for a fair comparison, all formations have approximately the same horizontal length ($92m$). The objective of this example is to analyze, using simulation, the

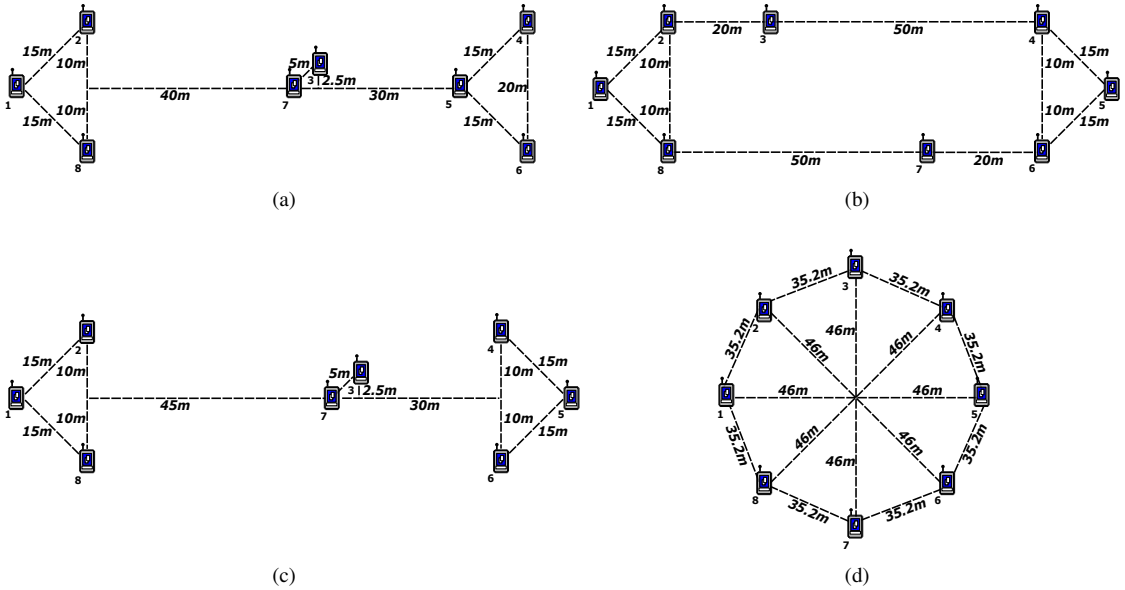


Figure 5.3: Formations used in the evaluation of the acoustic event detection system consisting of $N = 8$ sensors each. (a): $\mathcal{F}_1$; (b): $\mathcal{F}_2$; (c): $\mathcal{F}_3$; (d): $\mathcal{F}_4$. Note that distance constraints given are enough to determine a weighted adjacency matrix $\mathcal{A}_{\mathcal{F}_i}$ such that the formation graph is globally rigid.

localization error induced by the position uncertainty and the ToA error. Simulations were repeated 1000 times per formation, for different (random) position and ToA error values. The estimated location of the source was determined by solving (5.4) and (5.5) using the SDP technique described in [109], and using Protocol 5.1 for fusing local estimates in the distributed case.

Error modeling

To model both position and ToA errors, we use a truncated zero mean symmetric normal distribution $\mathrm{TN}(\bar{z}, \sigma^2)$ with probability density function (pdf) given

by [50]:

$$f(z) = \begin{cases} \frac{\frac{1}{\sigma}\phi\left(\frac{z}{\sigma}\right)}{2\Phi\left(\frac{\bar{z}}{\sigma}\right)-1} & , \ z \in [-\bar{z}, \bar{z}] \\ 0 & , \ z \notin [-\bar{z}, \bar{z}] \end{cases} \tag{5.15}$$

where ϕ and Φ are the standard normal pdf and cumulative density function (cdf), respectively. In the following, we will assume that the ToA error η_i follows a truncated distribution $\text{TN}(\bar{\eta}, 1)$ for all $i, j \in \mathcal{V}$. Similarly, we will assume that both the horizontal position error ϵ_x^i, and the vertical position error ϵ_y^i follow a truncated distribution $\text{TN}(\bar{p}, 1)$ for all $i \in \mathcal{V}$. Note that if ϵ_x^i and ϵ_y^i are i.i.d random variables with distribution $\text{TN}(\bar{p}, 1)$, then the Euclidean norm, i.e., the radial position error, follows a truncated Rayleigh distribution, which is consistent with the position error measured during field tests performed by the US ARMY RDECOM-ARDEC.

Impact on localization of position uncertainties

To test the impact of position uncertainties on the estimated location of the acoustic source, simulations were conducted in the absence of synchronization error. The position error components ϵ_x^i and ϵ_y^i are modeled as i.i.d random variables with distribution $\text{TN}(0.7m, 1m)$, i.e., a maximum error of $0.7m$ per component is assumed, which gives a radial position error of approximately $1m$ that is consistent with the order of accuracy of positioning sensors. Figure 5.4 shows centralized

detection simulations for the four formations in Figure 5.3 when there is a source of acoustic events (shooter) located at $\mathbf{S} = (0, -200)$, it can be seen that despite the position uncertainties all the four formations are able to locate the source position accurately. In fact, in all cases the Root-Mean-Square-Error (RMSE) defined as: $\sqrt{\frac{1}{1000}\sum d(\mathbf{S}, \hat{\mathbf{S}})^2}$ presents a value lower than $20m$ (see Table 5.1, $\bar{p} = 0.7$ case for details), i.e., below 10% of the distance from the shooter to the formation center. Similarly, Figure 5.5 shows distributed detection simulations for the four formations of Figure 5.3 and the same shooter location. We used the minimum transmission range such that Assumption 6 holds, hence each node performs localization using a subset of the ToA measurements. It can be seen in the figure, that despite the position uncertainties the system is able to localize the source accurately, with a RMSE below $17m$ and, moreover, outperforming the centralized results for all the formations tested. These localization results, both centralized and distributed, are considered acceptable for the particular system under analysis and represent a performance limit for the system with synchronization error.

Impact on localization of synchronization error

To test the impact of the synchronization error on the estimated location of the acoustic source, for a formation subject to a given level of position uncertainty, simulations were conducted considering: 1) that the position error components ϵ_x^i

and ϵ_y^i are i.i.d random variables with distribution TN$(0.7m, 1m)$; 2) the synchronization errors η_i are i.i.d random variables with distribution TN$(0.0025s, 1s)$ for the first set of simulations, and TN$(0.001s, 1s)$ for the second set of simulations. Note that a $\bar{\eta} = 0.0025s$ implies that the maximum skew of the network is $0.005s$. Figures 5.6 and 5.7 show centralized and distributed results for the case when $\bar{\eta} = 0.0025s$; it can be seen that the addition of a synchronization error greatly deteriorates the performance for all the four formations. Specifically, the RMSE increases by approximately 50% for the centralized case and by approximately 10% for the distributed case (see Table 5.1 for details). An interesting fact is that these results suggest that the consensus based distributed approach is more robust to synchronization errors than the classical centralized approach. However, the results obtained when $\bar{\eta} = 0.0025s$ are unacceptable in both cases, since the RMSE is larger than 10% of the distance from source to formation center. The fact that a network skew bounded by $0.005s$ produces a level of deterioration in localization performance that makes the system unusable, stresses the necessity of counting with a precise synchronization algorithm able to achieve a level of accuracy below this bound. Figures 5.8 and 5.9 show localization results when $\bar{\eta} = 0.001s$ for the centralized and distributed cases. In this case, detection results are very similar to the case when there is no synchronization error for both centralized and distributed localization (see Table 5.1 for details). These results suggest that a

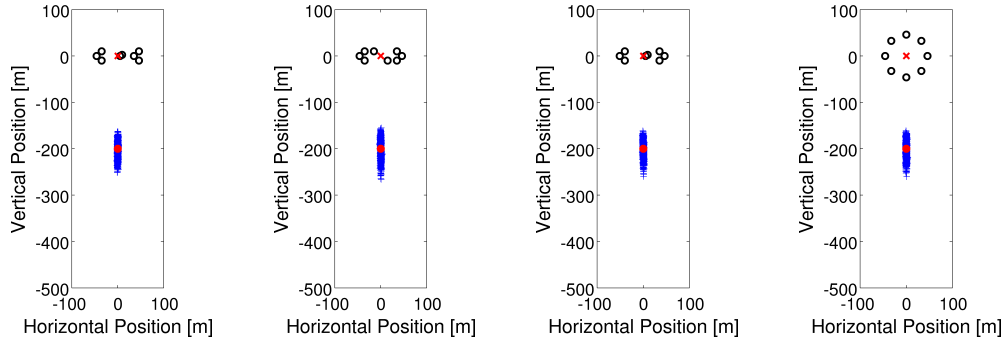


Figure 5.4: Results of the **centralized** acoustic source location estimation when $\bar{p}$ is $0.7m$ and the true position is $\mathbf{S} = (0m, -200m)$ for 1000 simulated events. Blue markers denote estimated locations while the red dot denote the actual source position; (a): $\mathcal{F}_1$; (b): $\mathcal{F}_2$; (c): $\mathcal{F}_3$; (d): $\mathcal{F}_4$.

synchronization error bounded near or below $0.001s$ should be sufficient for this particular application, i.e., a network skew below $0.002s$.

Table 5.1 summarizes centralized and distributed localization results obtained in the 3 cases analyzed in this section ($\bar{p} = 0.7m$, $\bar{\eta} = 0.001s$, and $\bar{\eta} = 0.0025s$) in terms of mean and standard deviation of the component errors defined as $x := \mathbf{S}_x - \hat{\mathbf{S}}_x$ and $y := \mathbf{S}_y - \hat{\mathbf{S}}_y$, as well as in terms of the RMSE. As previously mentioned, performance deteriorates as errors increase for all formations. $\mathcal{F}_1$, $\mathcal{F}_3$, and $\mathcal{F}_4$ show comparable performance, while the quad formation $\mathcal{F}_2$ presents the poorest result in all cases.

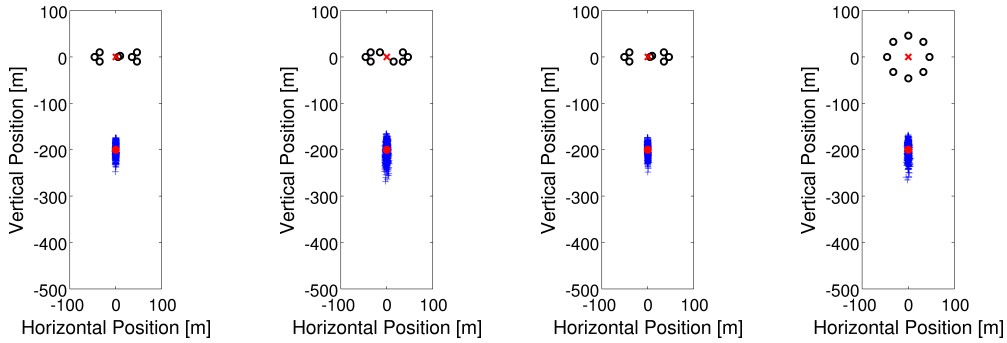


Figure 5.5: Results of the **distributed** acoustic source location estimation when $\bar{p}$ is $0.7m$ and the true position is $\mathbf{S} = (0m, -200m)$ for 1000 simulated events. Blue markers denote estimated locations while the red dot denote the actual source position; (a): $\mathcal{F}_1$; (b): $\mathcal{F}_2$; (c): $\mathcal{F}_3$; (d): $\mathcal{F}_4$.

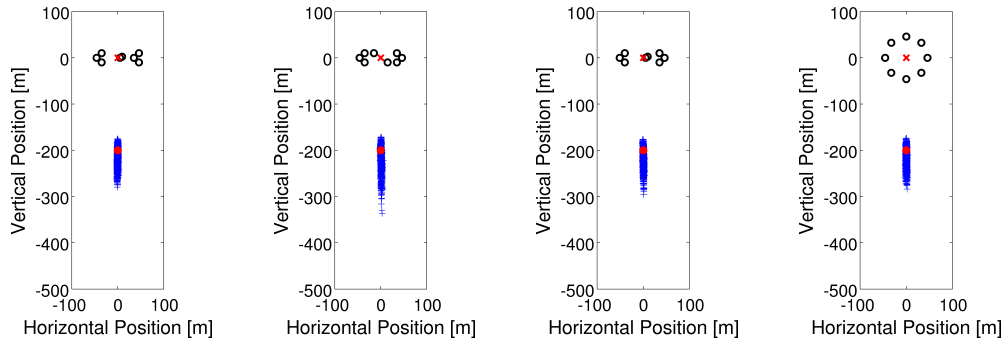


Figure 5.6: Results of the **centralized** acoustic source location estimation when $\bar{p} = 0.7m$, $\bar{\eta} = 0.0025s$ and the true position is $\mathbf{S} = (0m, -200m)$ for 1000 simulated events. Blue markers denote estimated locations while the red dot denote the actual source position; (a): $\mathcal{F}_1$; (b): $\mathcal{F}_2$; (c): $\mathcal{F}_3$; (d): $\mathcal{F}_4$.

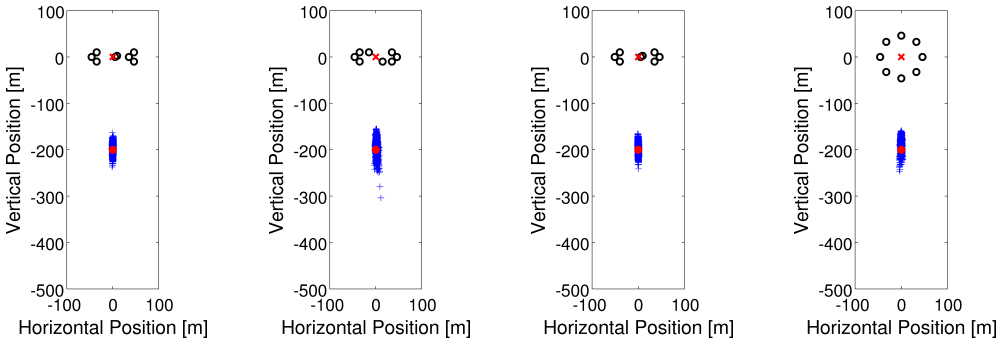


Figure 5.7: Results of the **distributed** acoustic source location estimation when $\bar{p} = 0.7m$, $\bar{\eta} = 0.0025s$ and the true position is $\mathbf{S} = (0m, -200m)$ for 1000 simulated events. Blue markers denote estimated locations while the red dot denote the actual source position; (a): $\mathcal{F}_1$; (b): $\mathcal{F}_2$; (c): $\mathcal{F}_3$; (d): $\mathcal{F}_4$.

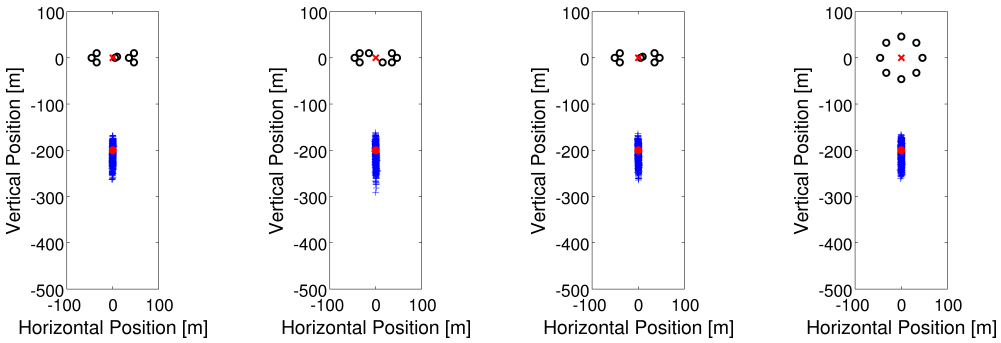
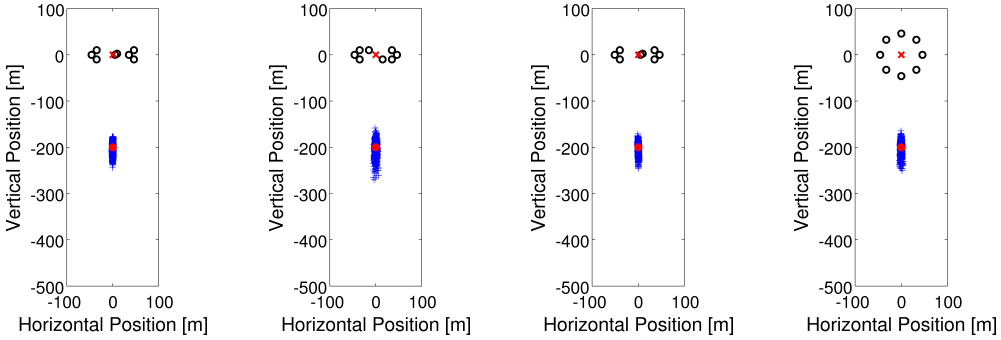


Figure 5.8: Results of the **centralized** acoustic source location estimation when $\bar{p} = 0.7m$, $\bar{\eta} = 0.001s$ and the true position is $\mathbf{S} = (0m, -200m)$ for 1000 simulated events. Blue markers denote estimated locations while the red dot denote the actual source position; (a): $\mathcal{F}_1$; (b): $\mathcal{F}_2$; (c): $\mathcal{F}_3$; (d): $\mathcal{F}_4$.



(a) (b) (c) (d)

Figure 5.9: Results of the **distributed** acoustic source location estimation when $\bar{p} = 0.7m$, $\bar{\eta} = 0.001s$ and the true position is $\mathbf{S} = (0m, -200m)$ for 1000 simulated events. Blue markers denote estimated locations while the red dot denote the actual source position; (a): $\mathcal{F}_1$; (b): $\mathcal{F}_2$; (c): $\mathcal{F}_3$; (d): $\mathcal{F}_4$.

Table 5.1: Summary of the localization results for the formations of Figure 5.3 for 1000 simulated events and different values of the error bounds $\bar{p}[m]$ and $\bar{\eta}[s]$. All the values are given in meters.

Setup		Centralized					Distributed				
		Mean x	Std x	Mean y	Std y	RMSE	Mean x	Std x	Mean y	Std y	RMSE
$\mathcal{F}_1$	$\bar{p} = 0.7$	-0.03	0.75	1.04	16.51	16.55	-0.02	0.73	0.92	11.77	11.83
	$\bar{\eta} = 0.001$	0.12	0.79	7.09	18.21	19.55	0.14	0.74	3.84	12.22	12.83
	$\bar{\eta} = 0.0025$	0.26	0.83	15.96	20.50	25.99	-0.07	0.77	-5.29	11.67	12.97
$\mathcal{F}_2$	$\bar{p} = 0.7$	-0.07	0.82	0.98	19.49	19.52	0.05	1.66	3.12	16.22	16.60
	$\bar{\eta} = 0.001$	-0.14	0.90	8.25	20.99	22.57	-0.01	1.66	4.55	16.83	17.51
	$\bar{\eta} = 0.0025$	-0.97	0.98	19.92	25.69	32.53	-1.91	1.76	-4.89	17.44	18.29
$\mathcal{F}_3$	$\bar{p} = 0.7$	-0.05	0.77	0.93	16.98	17.02	-0.41	0.69	0.83	11.99	12.03
	$\bar{\eta} = 0.001$	0.01	0.80	7.06	18.32	19.64	-0.01	0.70	3.68	12.52	13.06
	$\bar{\eta} = 0.0025$	-0.16	0.85	18.72	21.98	28.88	0.04	0.71	-4.82	12.23	13.16
$\mathcal{F}_4$	$\bar{p} = 0.7$	-0.06	0.82	0.97	17.84	17.87	-0.03	1.16	2.33	13.71	13.95
	$\bar{\eta} = 0.001$	-0.05	0.84	6.94	18.70	19.95	0.29	1.16	2.89	13.74	14.09
	$\bar{\eta} = 0.0025$	-0.21	0.91	16.53	21.23	26.92	0.16	1.19	-10.28	13.12	16.71

5.3 Evaluation of the Acoustic Event Detection System

In this section, we evaluate the acoustic event detection system using our PCO-protocol for time synchronization.

5.3.1 Transmission Range Assignment

In wireless sensor network-based systems, energy consumption is a critical issue, hence, it is necessary to choose carefully the transmission range given to each sensor in order to save energy. From Theorem 4.1 we know that the communication graph has to be connected for the synchronization protocol to work properly. Then, the transmission range $r \in (0, \bar{r}]$ must be such that if a formation graph $\mathcal{F}$ is given, it induces a connected communication graph $\mathcal{G}$. To find $\mathcal{G}$, first define the complete adjacency matrix of the formation graph as $\bar{\mathcal{A}}_{\mathcal{F}}$, which is obtained by finding the distances between every pair of nodes (note that since $\mathcal{F}$ is assumed to be globally rigid, its adjacency matrix $\mathcal{A}_{\mathcal{F}}$ contains all the information needed to obtain $\bar{\mathcal{A}}_{\mathcal{F}}$). Now consider a matrix-valued function $G : \mathbb{R}_{>0} \times \mathbb{R}^{N\times N} \to \mathbb{R}^{N\times N}$ such that:

$$[G(\alpha, M)]_{ij} = \begin{cases} 1, \text{ if } & [M]_{ij} \leq \alpha \\ 0, \text{ if } & [M]_{ij} > \alpha \end{cases} \tag{5.16}$$

and note that given $\mathcal{F}$ and r, the corresponding adjacency matrix of the communication graph is given by $\mathcal{A}_\mathcal{G} = G(r, \bar{\mathcal{A}}_\mathcal{F})$. From algebraic graph theory [35] it is well known that $\mathcal{G}$ being connected (when $\mathcal{G}$ includes all self-loops) is equivalent to ${\mathcal{A}_\mathcal{G}}^{N-1} > 0$, i.e., ${\mathcal{A}_\mathcal{G}}^{N-1}$ is a positive matrix. Moreover, we need to ensure that for every node i, $\{i\} \cup \mathcal{N}_i$ defines an identifying sensor set. Then, the problem of finding the smallest, considered optimal, r can be formulated as the following optimization problem:

$$\min r \tag{5.17}$$

subject to:

$$G(r, \bar{\mathcal{A}}_\mathcal{F})^{N-1} > 0 \tag{5.18a}$$

$$r \in (0, \bar{r}] \tag{5.18b}$$

$$\{i\} \cup \mathcal{N}_i \text{ is ISS} \tag{5.18c}$$

which always has a solution if $\bar{r}$ is large enough, in particular if $\bar{r}$ is larger than the largest element of $\mathcal{A}_\mathcal{F}$. Let r^* be the optimal transmission range, then $\mathcal{A}_\mathcal{G} = G(r^*, \bar{\mathcal{A}}_\mathcal{F})$. Solving for the particular formations used in this work (see Figure 5.3), and considering $\bar{r} = 100m$, we have: $r_1^* = 81.18m$, $r_2^* = 61.99m$, $r_3^* = 86.76m$, and $r_4^* = 65.05m$ for $\mathcal{F}_1$, $\mathcal{F}_2$, $\mathcal{F}_3$, and $\mathcal{F}_4$ respectively.

5.3.2 Implementation of the PCO-Based Protocol

To implement the pulse-coupled synchronization strategy presented in Chapter 4, each sensor constructs an embedded clock based on a counter, which increases at a given rate determined by the natural frequency (or a fraction of it) of the processor oscillating crystal from 0 to $x_{\text{th}} = CL$. The wireless transceiver of the sensor monitors the channel for incoming pulses. After initialization, an initial flooding is conducted; to this end, a sensor broadcasts a reset packet (with its unique ID specified in the packet) and resets its phase to 0 to begin the initial flooding process. Every sensor that receives the packet resets its phase to 0 and immediately passes the packet to its neighbors. A sensor having received the reset packet once will ignore all subsequently arriving identical reset packets to prevent broadcast storms. After the initial flooding, the pulse-based synchronization begins. During the refractory period, each sensor switches to sleep mode and turns off the wireless antenna to save energy. Once active, a node senses the channel for incoming pulses and it updates its internal phase according to the PRC and the coupling strength upon receptions. When the internal clock reaches the upper limit, a pulse is sent to the neighbors.

5.3.3 Integration of Time Synchronization and Localization

It was mentioned that pulse-coupled synchronization and the proposed distributed localization strategy integrate in harmony. The critical step is to switch the parameter x_{th} from the time synchronization constant CL to the localization constant $x_{\text{th}} = \frac{CL}{R}$, where R is the desired number of broadcasts per cycle for the localization algorithm. This is done using the following protocol that every node implements on startup.

Protocol 5.2 *On startup, nodes conduct a flooding process to ensure their phases are contained in half a circle. When this process is finished, nodes pick* $x_{th} = CL$*, initialize their phases as* $x_i = 0$ *and increase them at constant rate* w*.*

1. *At each phase event* $x_i = x_{th}$ *node* i *broadcasts a pulse. Upon receiving a pulse, node* i *updates its phase following the PRC and the coupling strength* l*.*

2. *Upon detecting an acoustic event, node* i *picks* $x_{th} = \frac{CL}{R}$*, switches to data sending mode and implements* Protocol 5.1*. Then, node* i *reports* $\hat{\mathbf{S}}$ *as solution, picks* $x_{th} = CL$*, switches to synchronization mode, and goes to 1).*

Remark 5.5 *It should be noted that while* Protocol 5.1 *is operating, time synchronization is not performed. This will lead to a drift produced by the non-identical natural frequencies. However, after restarting the synchronization algorithm, nodes will immediately go back in sync.*

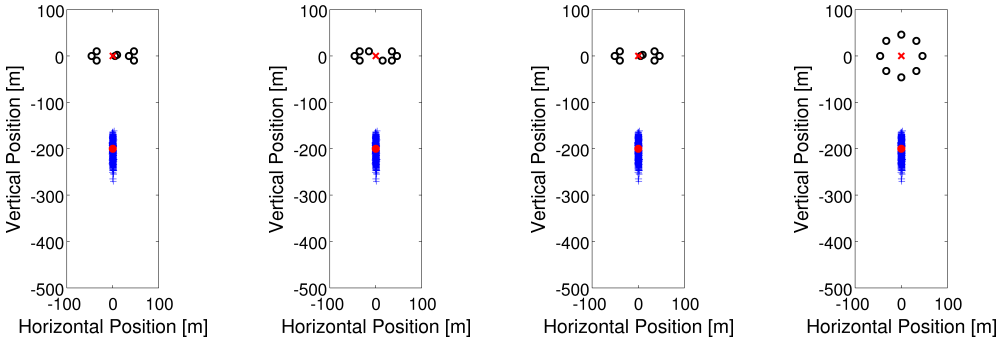
5.3.4 Evaluation for Nominal Formations

To evaluate the system as a whole, simulations of the acoustic event detection system using pulse-coupled synchronization were conducted for each formation. Simulations include position uncertainties bounded by $\bar{p} = 0.7m$ for both the horizontal and vertical components, and a source of acoustic events located at $\mathbf{S} = (0, -200)$ that generates events following a Poisson process with rate $\lambda = 24\frac{\text{events}}{\text{min}}$. To account for the drift found in real applications, natural periods of the internal sensors' clock were set as $(1 + \delta_i)s$ with $\delta_i \in [-0.001s, 0.001s]$ (note that common quartz crystals drift apart around $0.0001s/s$ [96], thus the considered error is reasonable for clocks built on top of an oscillating quartz crystal). The transmission range was set as $r_i^* + 1m$ to ensure connectivity, for every node in the formation $\mathcal{F}_i$. The system was tested in two different scenarios based on the value of the coupling strength used in the synchronization algorithm: $l = 0.8$ and $l = 1.0$. The drift and the initial synchronization error, which can be as large

as half natural period, i.e., $0.5s$, have to be compensated by the synchronization algorithm in order to enable accurate localization.

Figures 5.10 and 5.11 show centralized and distributed results obtained when $l = 0.8$. In this case, for all four formations in both the centralized and distributed cases, the RMSE values are below the values of the example when $\bar{\eta} = 0.001s$, the best result being achieved by $\mathcal{F}_1$ with a RMSE of $17.29m$ in the centralized case and of $12.10m$ in the distributed case, hence the results are considered acceptable (see Table 5.2 for details). Figures 5.12 and 5.13 show the results obtained for $l = 1.0$. In this case, all the formations present an acceptable RMSE below 10% of the distance from source to formation center (see Table 5.2 for details), the best being achieved by $\mathcal{F}_1$ with a RMSE of $16.97m$ in the centralized case and of $11.84m$ in the distributed case, which represent an improvement of approximately $0.3m$ with respect to the case with $l = 0.8$ for $\mathcal{F}_1$. The synchronization protocol is able to synchronize the network and compensate the drift present due to natural frequencies mismatch, thus enabling accurate localization.

Table 5.2 summarizes the results obtained in the experiments in terms of mean and standard deviation of the component errors and in terms of the RMSE. It can be seen that a stronger coupling strength results in a lower RMSE, suggesting that a strong coupling should be used. It should be noted that the distributed approach outperforms the centralized strategy in all cases. Considering that the distributed



(a) (b) (c) (d)

Figure 5.10: Results of the **centralized** acoustic source location estimation for 1000 events generated following a Poisson process with rate $\lambda = 24\frac{\text{events}}{\text{min}}$ and pulse-coupled synchronization with coupling strength $l = 0.8$, natural frequency $w = 2\pi$, position uncertainty bound $\bar{p} = 0.7m$, and true source position $\mathbf{S} = (0m, -200m)$. Blue markers denote estimated locations while the red dot denote the actual source position; (a): $\mathcal{F}_1$; (b): $\mathcal{F}_2$; (c): $\mathcal{F}_3$; (d): $\mathcal{F}_4$.

approach reduces the processing load at a single node, which is a drawback of centralized localization, and moreover, that distributed algorithms are scalable and robust to disconnections, distributed localization with pulse-coupled synchronization over a pure-broadcasting infrastructure-free ad-hoc network seems to be the ideal configuration to solve the acoustic source localization problem using a wireless sensor network.

5.3.5 Evaluation for Broken formations

Although the transmission range was selected to obtain a connected communication topology, in practical applications there are disturbances in the wireless channel that can disconnect the network. To evaluate the performance of the sys-

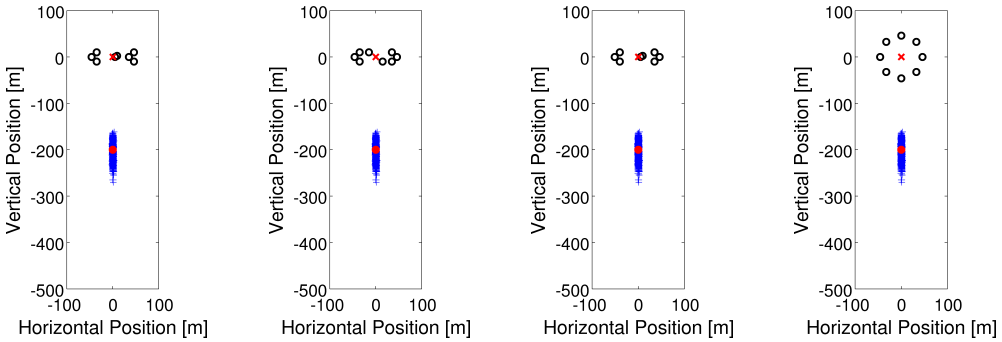


Figure 5.11: Results of the **distributed** acoustic source location estimation for 1000 events generated following a Poisson process with rate $\lambda = 24\frac{\text{events}}{\text{min}}$ and pulse-coupled synchronization with coupling strength $l = 0.8$, natural frequency $w = 2\pi$, position uncertainty bound $\bar{p} = 0.7m$, and true source position $\mathbf{S} = (0m, -200m)$. Blue markers denote estimated locations while the red dot denote the actual source position; (a): $\mathcal{F}_1$; (b): $\mathcal{F}_2$; (c): $\mathcal{F}_3$; (d): $\mathcal{F}_4$.

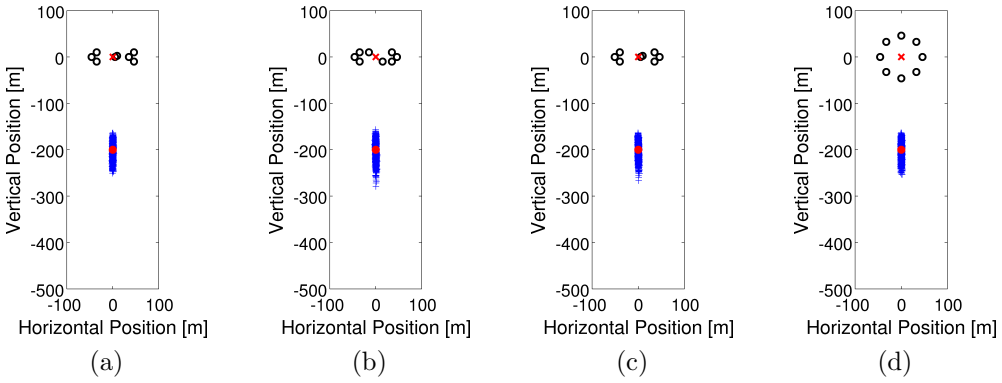


Figure 5.12: Results of the **centralized** acoustic source location estimation for 1000 events generated following a Poisson process with rate $\lambda = 24\frac{\text{events}}{\text{min}}$ and pulse-coupled synchronization with coupling strength $l = 1.0$, natural frequency $w = 2\pi$, position uncertainty bound $\bar{p} = 0.7m$, and true source position $\mathbf{S} = (0m, -200m)$. Blue markers denote estimated locations while the red dot denote the actual source position; (a): $\mathcal{F}_1$; (b): $\mathcal{F}_2$; (c): $\mathcal{F}_3$; (d): $\mathcal{F}_4$.

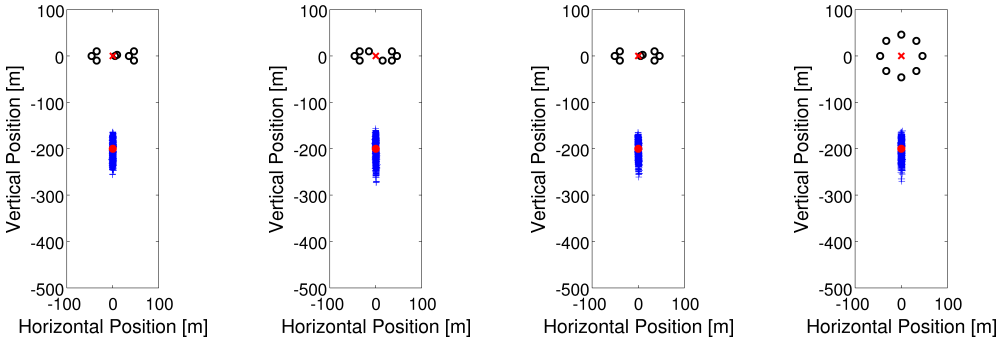


Figure 5.13: Results of the **distributed** acoustic source location estimation for 1000 events generated following a Poisson process with rate $\lambda = 24\frac{\text{events}}{\text{min}}$ and pulse-coupled synchronization with coupling strength $l = 1.0$, natural frequency $w = 2\pi$, position uncertainty bound $\bar{p} = 0.7m$, and true source position $\mathbf{S} = (0m, -200m)$. Blue markers denote estimated locations while the red dot denote the actual source position; (a): $\mathcal{F}_1$; (b): $\mathcal{F}_2$; (c): $\mathcal{F}_3$; (d): $\mathcal{F}_4$.

Table 5.2: Summary of the localization results for the formations of Figure 5.3 for 1000 events generated from a Poisson process and different values of the coupling strength l when $\bar{p} = 0.7m$. All the values are given in meters.

Setup		Centralized					Distributed				
		Mean x	Std x	Mean y	Std y	RMSE	Mean x	Std x	Mean y	Std y	RMSE
$\mathcal{F}_1$	$l = 0.8$	-0.18	0.76	2.66	17.07	17.29	-0.25	0.75	1.30	12.01	12.10
	$l = 1$	-0.14	0.76	1.91	16.85	16.97	-0.20	0.73	1.10	11.74	11.84
$\mathcal{F}_2$	$l = 0.8$	-0.27	0.83	2.66	20.33	20.51	-0.10	1.71	5.16	17.36	18.19
	$l = 1$	-0.23	0.85	1.60	19.49	19.56	-0.07	1.72	4.68	17.04	17.74
$\mathcal{F}_3$	$l = 0.8$	-0.27	0.78	2.98	17.07	17.34	-0.24	0.70	2.20	12.38	12.59
	$l = 1$	-0.21	0.78	2.39	17.16	17.33	-0.20	0.70	1.67	12.10	12.23
$\mathcal{F}_4$	$l = 0.8$	-0.31	0.83	3.30	18.30	18.61	-0.14	1.24	2.02	14.38	14.56
	$l = 1$	-0.24	0.81	2.47	17.88	18.06	-0.12	1.13	1.57	13.91	14.04

tem in this case, we consider a scenario where each formation is broken, from a communication point of view, into 2 sub-formations and perform distributed localization on each sub-formation. We broke the weakest communication links in each formation to obtain the following sub-formations: For the original formation $\mathcal{F}_1$, $\mathcal{F}_{11}$ is formed by nodes $3, 4, 5, 6, 7$ and $\mathcal{F}_{12}$ is formed by nodes $1, 2, 8$. For the original formation $\mathcal{F}_2$, $\mathcal{F}_{21}$ is formed by nodes $1, 2, 3, 8$ and $\mathcal{F}_{22}$ is formed by nodes $4, 5, 6, 7$. For the original formation $\mathcal{F}_3$, $\mathcal{F}_{31}$ is formed by nodes $3, 4, 5, 6, 7$ and $\mathcal{F}_{22}$ is formed by nodes $1, 2, 8$. Finally, for the original formation $\mathcal{F}_4$, $\mathcal{F}_{41}$ is formed by nodes $3, 4, 5, 6, 7$ and $\mathcal{F}_{42}$ is formed by nodes $1, 2, 8$. The sub-formations resulting from breaking the weakest link are illustrated in Figure 5.14, where blue circles denote $\mathcal{F}_{i1}$ sub-formations and green circles denote $\mathcal{F}_{i2}$ sub-formations. Figure 5.15 presents results obtained in the broken formation case for $l = 1.0$. It can be seen that performance deteriorates, yet each sub-formation is still able to locate the acoustic source with some level of coherence, specially sub-formations $\mathcal{F}_{11}$, $\mathcal{F}_{31}$, and $\mathcal{F}_{41}$ that present a RMSE below $30m$. Note that sub-formations $\mathcal{F}_{12}$, $\mathcal{F}_{32}$, and $\mathcal{F}_{42}$ do not define an identifying sensor set; hence, in some cases, they cannot uniquely identify the location of the acoustic source, and moreover, they are very sensitive to small synchronization and position errors as can be seen in Figure 5.15. Sub-formation $\mathcal{F}_{41}$ presents the best performance with a RMSE of $22.15m$.

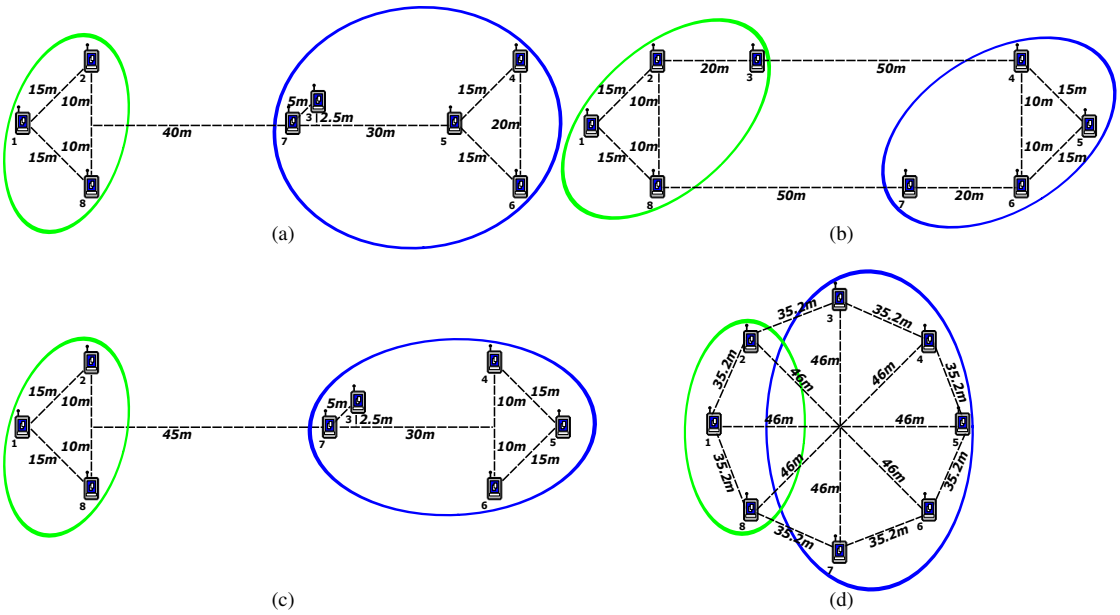


Figure 5.14: Formations used in the broken formation case. Blue circles denote $\mathcal{F}_{i1}$ sub-formations and green circles denote $\mathcal{F}_{i2}$ sub-formations. (a): $\mathcal{F}_1$; (b): $\mathcal{F}_2$; (c): $\mathcal{F}_3$; (d): $\mathcal{F}_4$.

Table 5.3 summarizes the results obtained in the broken formation case in terms of mean and standard deviation of the component errors and in terms of the RMSE.

The results obtained in this Chapter suggest that distributed localization with pulse-coupled synchronization over a pure-broadcasting infrastructure-free ad-hoc network, seems to be the ideal configuration to solve the acoustic source localization problem when using a wireless sensor network. The time synchronization protocol designed using the PCO synchronization mechanism has shown to be mature enough to be applied to a real practical problem.

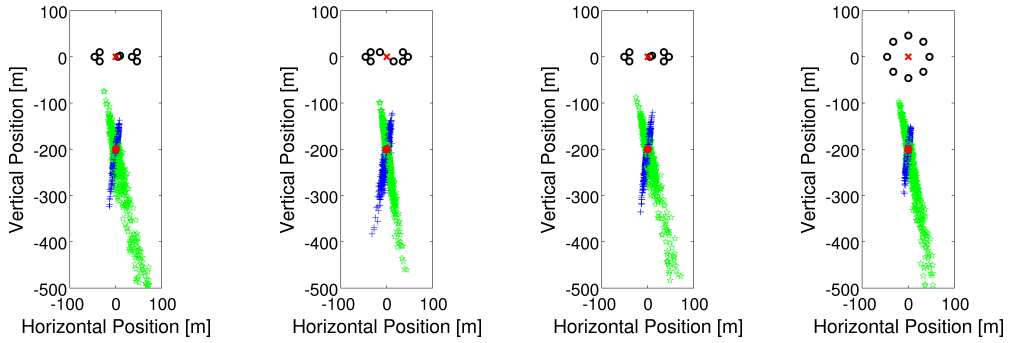


Figure 5.15: Results of the acoustic source location estimation in the broken formation case for 1000 events generated following a Poisson process with rate $\lambda = 24\frac{\text{events}}{\text{min}}$ and pulse-coupled synchronization with coupling strength $l = 1.0$, natural frequency $w = 2\pi$, and true source position $\mathbf{S} = (0m, -200m)$. Blue $+$ markers denote $\mathcal{F}_{i1}$ estimations and green $\star$ markers denote $\mathcal{F}_{i2}$ estimations, for $i = 1, \ldots, 4$. (a): $\mathcal{F}_1$; (b): $\mathcal{F}_2$; (c): $\mathcal{F}_3$; (d): $\mathcal{F}_4$.

Table 5.3: Summary of the localization results for the formations of Figure 5.3 for 1000 events generated from a Poisson process when the formation is broken into two sub-formations. All the values are given in meters.

Setup	Mean x	Std x	Mean y	Std y	RMSE
$\mathcal{F}_{11}$	-0.23	2.96	-1.27	22.45	22.67
$\mathcal{F}_{12}$	-10.23	27.24	59.39	157.31	170.57
$\mathcal{F}_{21}$	1.17	5.31	10.08	35.43	37.21
$\mathcal{F}_{22}$	-2.09	7.37	13.27	50.69	52.93
$\mathcal{F}_{31}$	0.66	3.53	6.87	28.96	29.97
$\mathcal{F}_{32}$	-8.15	25.51	41.73	133.35	142.20
$\mathcal{F}_{41}$	0.26	2.28	3.70	21.72	22.15
$\mathcal{F}_{42}$	-5.60	14.46	28.63	79.57	85.94

Chapter 6

Conclusions and Future Work

In this dissertation we studied synchronization of networks of PCOs and its application to the time synchronization problem in wireless sensor networks. The contributions of this work range from conditions to ensure synchronization for a variety of networks, on the theoretical side, to the design and implementation of a PCO-based time synchronization protocol for wireless sensor networks, on the application side. A complete summary of the topics treated and the contributions of this dissertation is given in the next section, followed by a set of related topics envisioned as future related work.

6.1 Summary

In Chapter 2 we presented the new model (2.7)-(2.11) for networks of PCOs that is able to handle naturally the impulsive nature of the coupling and the continuous time nature of the limit cycle oscillator. Moreover, the model includes

explicitly the structure of the underlying communication topology as part of the model and allows using discontinuous or multi-valued PRCs. Furthermore, our hybrid model allows using well established tools for the analysis of hybrid systems to study properties of networks without needing restrictive assumptions such as the existence of an invariant firing sequence or weak coupling.

In chapter 3 we presented the main theoretical contributions of this dissertation. We started by showing in Theorem 3.1 that networks of identical PCOs can synchronize to a global cue, or leader node, under mild conditions on the strength of the global coupling. A similar set of conditions was derived for non-identical PCOs in Theorem 3.2 when the feedback is given by the rate optimal PRC (3.11), stating that a network of non identical PCOs can synchronize in frequency to a global cue under mild conditions on the coupling strength. We continued our analysis deriving conditions to ensure global synchronization in the purely decentralized case for all-to-all (*cf.* Theorem 3.3 and Corollary 3.2), strongly rooted (*cf.* Corollary 3.3), and strongly connected networks (*cf.* Theorems 3.4 and 3.8). Finally, given the importance of cycle networks in both biological and engineering systems, we explored cycle networks in detail and found the exact value of the critical coupling that enables global synchronization for both bidirectional (*cf.* Theorem 3.6) and unidirectional cycles (*cf.* Theorem 3.7).

In Chapter 4 we presented the translation of the PCO paradigm into a functional time synchronization protocol for wireless sensor networks. The protocol (*cf.* Protocol 4.1) is given in general algorithmic form, enabling its implementation at any level of the networking protocol stack, although it is recommended to implement it as low as possible to improve the accuracy. An extensive evaluation by simulation in Qualnet is given to explore the synchronization properties of the algorithm in a variety of network structures and communication topologies. A comparison with FTSP, the state-of-the-art synchronization protocol, is also given showing that our protocol outperforms FTSP in terms of both accuracy and energy consumption (*cf.* Table 4.8). The pilot implementation in Gumstix development boards and in commercial acoustic wireless sensors given at the end of the Chapter illustrates the feasibility of implementing the protocol in real hardware platforms.

In Chapter 5 we presented the first implementation of pulse-coupled synchronization to an actual experimental system. As a side contribution, we proposed a new method for performing distributed localization in a network of acoustic sensors, which uses only local time-of-arrival measurements and then fuses individual estimates using consensus algorithms (*cf.* Theorem 5.1 and Protocol 5.1). Although pilot implementations of pulse-coupled synchronization are available in the literature, the restrictive assumption of having a dedicated network for time synchronization hinders the evaluation of the actual performance in a real shared

network. In contrast, we were able to combine pulse-coupled time synchronization with distributed localization over an infrastructure free wireless network is an harmonious way (*cf.* Protocol 5.2). Moreover, we showed that the use of pulse-coupled synchronization greatly improves the localization results (*cf.* Table 5.2) without imposing extra computational requirements on the sensors due to the simplicity and natural scalability of our PCO-based synchronization protocol.

6.2 Future Work

The results presented in this dissertation answer numerous questions regarding synchronization of networks of PCOs, but they also open new venues for research, many of them as natural extensions of our new results. Among many possible directions, the following are important topics that should be addressed.

Decentralized synchronization of heterogeneous networks of PCOs: A natural extension of the results given for decentralized networks is to consider heterogeneous PCOs. In Chapter 3 we stated conditions for synchronization in frequency for non-identical PCOs when there is a global cue. In the decentralized case we have observed in simulations that synchronization in frequency also emerges although the collective period is an extra unknown and, possibly, depends on the initial conditions. Finding conditions for synchronization in frequency for decentralized networks of non-identical PCOs is of great importance since in prac-

tical applications identical oscillators are highly unlikely to exist. Existing efforts in this direction have found conditions ensuring synchronization in frequency, yet results are very restrictive or for a particular network size. For example, the work in [92] presents an appealing method to predict synchronization in pairs of PCOs. However, the strategy cannot be applied to analyze synchronization in large networks. The work in [3] presents a strategy to ensure almost global synchronization in frequency in large networks of linear PCOs. However, the strategy is based on applying a strong coupling that forces the system to oscillate with a collective frequency equal to the largest frequency in the network, i.e., following the fastest oscillator in the network. A similar setup is analyzed in [13, 20] for the classical PCO model under all-to-all coupling, confirming that synchronization in frequency to the fastest oscillator is feasible under mild assumptions. However, such a result presents several drawbacks, among them are the non-robustness of the collective frequency, since any perturbation to the frequency of the fastest oscillator implies convergence of the network to a new collective frequency, and the huge damage that a Byzantine agent can cause to the network. Advantage should be taken of the consensus-like behavior observed in PCOs to obtain a robust collective frequency, yet extra research is needed to find the conditions that guarantee convergence and stability.

Effects of delays: From a practical point of view, transmissions in a wireless network are not instantaneous. Therefore, the presence of (possibly non-identical) delays must be included in the formulation in order to explore its effect. Existing works on this topic have given initial insights into the feasibility of synchronization when delays in the communication are present. In [65], a PRC similar to (3.11) is used to analyze local synchronization of PCO networks on aperiodic graphs with delays. The authors consider an allowable region for the range of the PRC that is similar to the set Ω; however, they do not allow discontinuities in the PRC, which is the key point to establish global convergence. In fact, the stability results in [65] are local and the authors use a probabilistic measure to evaluate the synchronization of the system. The work in [66] aims to generalize the results in [65] in a probabilistic setting by calculating the probability of synchronization of a network with arbitrary initial conditions. In [55] it is shown that heterogeneous delays lead to synchronization in a weakly coupled network of PCOs; however, no insights are given regarding the general coupling case. Although strong (global) results, such as our Theorem 3.8, exist for the delay-free case, global synchronization conditions for the delayed PCOs case are still to be obtained.

Mobility and topology control: The increasing trend to use mobile WSNs poses new difficulties to synchronization algorithms that need to deal with the effects of mobility on the resulting communication topology of the sensor network.

Partial or long-term disconnections are likely to happen in a mobile network, making network-wide synchronization impossible to achieve. In this setting, topology control algorithms [84] are needed to ensure that the network will maintain a proper level of connectivity throughout its operation. Topology control strategies for mobile networks are at an early development stage, yet an increasing number of strategies are being currently studied with promising results [84]. In the particular case of our PCO-based protocol, a topology control strategy will greatly benefit its performance. For example, condition 1) in Theorem 3.8 can be ensured by properly controlling the communication topology, thus guaranteeing global synchronization. Moreover, in our numerical experiments presented in Chapter 4, we have found that dense networks present lower network skew and time to synchronization. Furthermore, in our tests using Gumstix boards, we have found that the hidden terminal problem affects the stability of the collective period. A proper topology control strategy that can be incorporated into the PCO-based protocol will increase the applicability and improve performance; yet, such a strategy is still to be proposed.

Kernel level implementation of the synchronization protocol: As it was mentioned at the end of Chapter 4, an appealing option to reduce both the network skew and the jitter in the free running period is to implement the synchronization protocol as part of the operating system's kernel. The idea is to create a kernel

module that implements the protocol by directly writing pulse messages in the output queue of the MAC layer and capturing incoming messages just before they are processed by the MAC layer. Following this strategy, the sensors will benefit of the highest possible accuracy. Moreover, the operating system will perform time synchronization automatically since it will be part of the basic kernel-controlled processes. Hence, time synchronization will be transparent to the user and it will enjoy kernel-level priority. It should be noted that this strategy is not new; in fact, the network time protocol (NTP) is implemented in Linux systems in a similar way, as a kernel-controlled task.

Bibliography

[1] L. F. Abbott. Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain Research Bulletin*, 50(5-6):303–304, 1999.

[2] S. An, R. Harang, K. Meeker, D. Granados-Fuentes, C. A. Tsai, C. Mazuski, J. Kim, F. J. Doyle III, L. R. Petzold, and E. D. Herzog. A neuropeptide speeds circadian entrainment by reducing intercellular synchrony. *Proceedings of the National Academy of Sciences*, 110(46):E4355–E4361, 2013.

[3] Z. An, H. Zhu, X. Li, C. Xu, Y. Xu, and X. Li. Nonidentical linear pulse-coupled oscillators model with application to time synchronization in wireless sensor networks. *IEEE Transactions on Industrial Electronics*, 58(6):2205–2215, 2011.

[4] B. D. O. Anderson, C. Yu, B. Fidan, and J. M. Hendrickx. Rigid graph control architectures for autonomous formations. *IEEE Control Systems Magazine*, 28(6):48–63, 2008.

[5] D. Angeli and E. D. Sontag. Oscillations in I/O monotone systems under negative feedback. *IEEE Transactions on Automatic Control*, 53:166–176, 2008.

[6] M. Arcak. Passivity as a design tool for group coordination. *IEEE Transactions on Automatic Control*, 52(8):1380–1390, 2007.

[7] S. Barbarossa and G. Scutari. Bio-inspired sensor network design. *IEEE Signal Processing Magazine*, 24(3):26–35, 2007.

[8] J. Bedard and S. Pare. Ferret: a small arms fire detection system: localization concepts. In Edward M. Carapezza, editor, *SPIE*, volume 5071, pages 497–509, 2003.

[9] C. Bettstetter. Mobility modeling in wireless networks: categorization, smooth movement, and border effects. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(3):55–66, 2001.

[10] C. Bettstetter, H. Hartenstein, and X. Pérez-Costa. Stochastic properties of the random waypoint mobility model. *Wireless Networks*, 10(5):555–567, 2004.

[11] BioMimetic Systems, Inc. http://www.biomimetic-systems.com, Accessed June 2014.

[12] A. N. Bishop, B. Fidan, B. D. O. Anderson, K. Dogancay, and P. N. Pathirana. Optimality analysis of sensor-target localization geometries. *Automatica*, 46(3):479–492, 2010.

[13] S. Bottani. Synchronization of integrate and fire oscillators with global coupling. *Physical Review E*, 54:2334–2350, 1996.

[14] E. Brown, J. Moehlis, and P. Holmes. On the phase reduction and response dynamics of neural oscillator populations. *Neural computation*, 16(4):673–715, 2004.

[15] J. B. Buck. Synchronous rhythmic flashing of fireflies. *The Quarterly Review of Biology*, 13(3):301–314, 1938.

[16] J. B. Buck. Synchronous rhythmic flashing of fireflies II. *The Quarterly Review of Biology*, 63(3):265–289, 1988.

[17] G. Cakiades, S. Dasay, S. Deligeorges, B. Buckland, and J. George. Fusion solution for soldier wearable gunfire detection systems. In Edward M. Carapezza, editor, *SPIE*, volume 8388, 2012.

[18] C. C. Canavier and S. Achuthan. Pulse coupled oscillators and the phase resetting curve. *Mathematical Biosciences*, 226(2):77–96, 2010.

[19] M. Cao, A. S. Morse, and B. D. O. Anderson. Reaching a consensus in a dynamically changing environment: A graphical approach. *SIAM Journal on Control and Optimization*, 47(2):575–600, 2008.

[20] Y.-C. Chang and J. Juang. Stable synchrony in globally coupled integrate-and-fire oscillators. *SIAM Journal on Applied Dynamical Systems*, 7(4):1445–1476, 2008.

[21] J. Cortes, S. Martinez, and F. Bullo. Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Transactions on Automatic Control*, 51(8):1289–1298, 2006.

[22] S. Daan and C. S. Pittendrigh. A functional analysis of circadian pacemakers in nocturnal rodents. *Journal of Comparative Physiology*, 106(3):253–266, 1976.

[23] J. Degesys, I. Rose, A. Patel, and R. Nagpal. Desync: Self-organizing desynchronization and TDMA on wireless sensor networks. In *6th International Symposium on Information Processing in Sensor Networks, IPSN 2007*, pages 11–20, 2007.

[24] P. DeLellis, M. di Bernardo, and F. Garofalo. Novel decentralized adaptive strategies for the synchronization of complex networks. *Automatica*, 45(5):1312–1318, 2009.

[25] F. Dorfler and F. Bullo. On the critical coupling for Kuramoto oscillators. *SIAM Journal on Applied Dynamical Systems*, 10(3):1070–1099, 2011.

[26] R. O. Dror, C. C. Canavier, R. J. Butera, J. W. Clark, and J. H. Byrne. A mathematical criterion based on phase response curves for stability in a ring of coupled oscillators. *Biological Cybernetics*, 80(1):11–23, 1999.

[27] M. G. Earl and S. H. Strogatz. Synchronization in oscillator networks with delayed coupling: A stability criterion. *Physical Review E*, 67:036204, 2003.

[28] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. *SIGOPS Operating Systems Review*, 36(SI):147–163, 2002.

[29] B. Ermentrout. Type I membranes, phase resetting curves, and synchrony. *Neural Computation*, 8:979–1001, 1995.

[30] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. Efficient network flooding and time synchronization with glossy. In *10th International Conference on Information Processing in Sensor Networks, IPSN 2011*, pages 73–84, 2011.

[31] N. M. Freris, S. R. Graham, and P. R. Kumar. Fundamental limits on synchronizing clocks over networks. *IEEE Transactions on Automatic Control*, 56(6):1352–1364, 2011.

[32] S. Ganeriwal, R. Kumar, and M. B. Srivastava. Timing-sync protocol for sensor networks. In *1st ACM Conference on Embedded Networked Sensor Systems, SenSys '03*, pages 138–149, 2003.

[33] J. George and L. M. Kaplan. Shooter localization using soldier-worn gunfire detection systems. In *14th International Conference on Information Fusion, FUSION 2011*, pages 1–8, 2011.

[34] W. Gerstner and W. M. Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

[35] C. Godsil and G. Royle. *Algebraic Graph Theory*. Springer, 2001.

[36] R. Goebel, R. Sanfelice, and A. R. Teel. Hybrid dynamical systems. *IEEE Control Systems Magazine*, 29(2):28–93, 2009.

[37] R. Goebel, R. G. Sanfelice, and A. R. Teel. *Hybrid dynamical systems: modeling, stability, and robustness*. Princeton University Press, 2012.

[38] D. Grasing and S. Desai. Data fusion methods for small arms localization solutions. In *15th International Conference on Information Fusion, FUSION 2012*, pages 713–718, 2012.

[39] Gumstix, Inc. http://www.gumstix.com, Accessed September 2013.

[40] F. E. Hanson, J. F. Case, E. Buck, and J. Buck. Synchrony and flash entrainment in a new guinea firefly. *Science*, 174(4005):161–164, 1971.

[41] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4):500–544, 1952.

[42] Y. W. Hong and A. Scaglione. Time synchronization and reach-back communications with pulse-coupled oscillators for uwb wireless ad hoc networks. In *IEEE Conference on Ultra Wideband Systems and Technologies, 2003*, pages 190–194, 2003.

[43] Y. W. Hong and A. Scaglione. A scalable synchronization protocol for large scale sensor networks and its applications. *IEEE Journal on Selected Areas in Communications*, 23(5):1085–1099, 2005.

[44] F. C. Hoppensteadt and E. M. Izhikevich. *Weakly Connected Neural Networks*. Applied Mathematical Sciences. Springer, 1997.

[45] A.-S. Hu and S. D. Servetto. On the scalability of cooperative time synchronization in pulse-connected networks. *IEEE Transactions on Information Theory*, 14(6):2725–2748, 2006.

[46] J. T. Isaacs, D. J. Klein, and J. P. Hespanha. Optimal sensor placement for time difference of arrival localization. In *48th IEEE Conference on Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference, CDC/CCC 2009*, pages 7878–7884, 2009.

[47] E. M. Izhikevich. Weakly pulse-coupled oscillators, FM interactions, synchronization, and oscillatory associative memory. *IEEE Transactions on Neural Networks*, 10(3):508–526, 1999.

[48] E. M. Izhikevich. Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6):1569–1572, 2003.

[49] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[50] N. L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Univariate Distributions, Vol. 1.* John Wiley & Sons, 1994.

[51] K. Konishi and H. Kokame. Synchronization of pulse-coupled oscillators with a refractory period and frequency distribution for a wireless sensor network. *Chaos*, 18(3):033132, 2008.

[52] H. Kopetz and W. Ochsenreiter. Clock synchronization in distributed real-time systems. *IEEE Transactions on Computers*, C-36(8):933–940, 1987.

[53] C. Lenzen, P. Sommer, and R. Wattenhofer. Optimal clock synchronization in networks. In *7th ACM Conference on Embedded Networked Sensor Systems, SenSys '09*, pages 225–238, 2009.

[54] J. Liu and A. S. Morse. Asynchronous distributed averaging using double linear iterations. In *2012 American Control Conference (ACC)*, pages 6620–6625, 2012.

[55] E. Mallada and A. Tang. Weakly pulse-coupled oscillators: Heterogeneous delays lead to homogeneous phase. In *49th IEEE Conference on Decision and Control, CDC 2010*, pages 992–997, 2010.

[56] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi. The flooding time synchronization protocol. In *2nd ACM Conference on Embedded Networked Sensor Systems, SenSys '04*, pages 39–49, 2004.

[57] A. Mauroy. *On the dichotomic collective behaviors of large populations of pulse-coupled firing oscillators*. PhD thesis, University of Liège, Liège, Belgium, 2011.

[58] A. Mauroy, P. Sacre, and R. Sepulchre. Kick synchronization versus diffusive synchronization. In *51th IEEE Conference on Decision and Control, CDC 2012*, pages 7171–7183, 2012.

[59] A. Mauroy and R. Sepulchre. Clustering behaviors in networks of integrate-and-fire oscillators. *Chaos*, 18(3):037122, 2008.

[60] A. Mauroy and R. Sepulchre. Contraction of monotone phase-coupled oscillators. *Systems & Control Letters*, 61(11):1097–1102, 2012.

[61] M. Mesbahi and M. Egerstedt. *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[62] R. E. Mirollo and S. H. Strogatz. Synchronization of pulse-coupled biological oscillators. *SIAM Journal on Applied Mathematics*, 50(6):1645–1662, 1990.

[63] A. Nabi and J. Moehlis. Time optimal control of spiking neurons. *Journal of Mathematical Biology*, 64(6):981–1004, 2012.

[64] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis. On distributed averaging algorithms and quantization effects. *IEEE Transactions on Automatic Control*, 54(11):2506–2517, 2009.

[65] J. Nishimura and E. J. Friedman. Robust convergence in pulse-coupled oscillators with delays. *Physical Review Letters*, 106:194101, 2011.

[66] J. Nishimura and E. J. Friedman. Probabilistic convergence guarantees for type-ii pulse-coupled oscillators. *Physical Review E*, 86:025201, 2012.

[67] D. Noble and R. W. Tsien. Reconstruction of the repolarization process in cardiac purkinje fibres based on voltage clamp measurements of membrane current. *The Journal of Physiology*, 200(1):233–254, 1969.

[68] F. Núñez, Y. Wang, S. Desai, G. Cakiades, and F. J. Doyle III. Bio-inspired synchronization of wireless sensor networks for acoustic event detection systems. In *2012 International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication, ISPCS 2012*, pages 85–90, 2012.

[69] F. Núñez, Y. Wang, and F. J. Doyle III. Bio-inspired hybrid control of pulse-coupled oscillators and application to synchronization of a wireless network. In *2012 American Control Conference, ACC 2012*, pages 2818–2823, 2012.

[70] F. Núñez, Y. Wang, and F. J. Doyle III. Global synchronization of pulse-coupled oscillators interacting on cycle graphs. *Submitted*, 2013.

[71] F. Núñez, Y. Wang, and F. J. Doyle III. Synchronization of pulse-coupled oscillators on (strongly) connected graphs. *IEEE Transactions on Automatic Control (To appear)*, 2013.

[72] F. Núñez, Y. Wang, and F. J. Doyle III. Synchronization of pulse-coupled oscillators to a global pacemaker. *Submitted*, 2014.

[73] F. Núñez, Y. Wang, D. Grasing, S. Desai, G. Cakiades, and F. J. Doyle III. Pulse-coupled time synchronization for distributed acoustic event detection using wireless sensor networks. *Submitted*, 2014.

[74] F. Núñez, Y. Wang, A. R. Teel, and F. J. Doyle III. Bio-inspired synchronization of non-identical pulse-coupled oscillators subject to a global cue and local interactions. In 4^{th} *IFAC Conference on Analysis and Design of Hybrid Systems, ADHS12*, pages 115–120, 2012.

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

[76] R. Pagliari and A. Scaglione. Scalable network synchronization with pulse-coupled oscillators. *IEEE Transactions on Mobile Computing*, 10(3):392–405, 2011.

[77] N. Patwari, A. O. Hero III, M. Perkins, N. S. Correal, and R. J. O'Dea. Relative location estimation in wireless sensor networks. *IEEE Transactions on Signal Processing*, 51(8):2137–2148, 2003.

[78] C. S. Peskin. *Mathematical aspects of heart physiology*. Courant Institute Lecture Notes. Courant Institute of Mathematical Sciences, New York University, 1975.

[79] R. T. Rockafellar and R. J.-B. Wets. *Variational analysis*. Springer-Verlag, New York, 1998.

[80] J. Sallai, B. Kusý, Á. Lédeczi, and P. Dutta. On the scalability of routing integrated time synchronization. In *Third European Conference on Wireless Sensor Networks, EWSN 06*, pages 115–131, 2006.

[81] J. Sallai, Á. Lédeczi, and P. Völgyesi. Acoustic shooter localization with a minimal number of single-channel wireless sensor nodes. In *9th ACM Conference on Embedded Networked Sensor Systems*, *SenSys '11*, pages 96–107, 2011.

[82] R. G. Sanfelice. Simulating hybrid systems in Matlab/Simulink. http://www.u.arizona.edu/sricardo/software.html, Accessed May 2011.

[83] R. G. Sanfelice and A. R. Teel. Dynamical properties of hybrid systems simulators. *Automatica*, 46(2):239–248, 2010.

[84] P. Santi. Topology control in wireless ad hoc and sensor networks. *ACM Computing Surveys (CSUR)*, 37(2):164–194, 2005.

[85] Scalable networks inc. Qualnet 4.5 users guide. http://www.scalable-networks.com, Accessed November 2010.

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

[87] O. Simeone, U. Spagnolini, Y. Bar-Ness, and S. H. Strogatz. Distributed synchronization in wireless networks. *IEEE Signal Processing Magazine*, 25(5):81–97, 2008.

[88] R. M. Smeal, G. B. Ermentrout, and J. A. White. Phase-response curves and synchronized neural networks. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365(1551):2407–2422, 2010.

[89] P. Sommer and R. Wattenhofer. Gradient clock synchronization in wireless sensor networks. In *2009 International Conference on Information Processing in Sensor Networks*, IPSN '09, pages 37–48, 2009.

[90] G.-B. Stan and R. Sepulchre. Analysis of interconnected oscillators by dissipativity theory. *IEEE Transactions on Automatic Control*, 52(2):256–270, 2007.

[91] M. Stopfer, S. Bhagavan, B. H. Smith, and G. Laurent. Impaired odour discrimination on desynchronization of odour-encoding neural assemblies. *Nature*, 390(6655):70–4, 1997.

[92] S. S. Talathi, D.-U. Hwang, A. Miliotis, P. R. Carney, and W. L. Ditto. Predicting synchrony in heterogeneous pulse coupled oscillators. *Physical Review E*, 80:021908, 2009.

[93] A. Tsertou and D. I. Laurenson. Revisiting the hidden terminal problem in a csma/ca wireless network. *IEEE Transactions on Mobile Computing*, 7(7):817–831, 2008.

[94] A. Tyrrell, G. Auer, and C. Bettstetter. Emergent slot synchronization in wireless networks. *IEEE Transactions on Mobile Computing*, 9(5):719–732, 2010.

[95] S. Venkateswaran and U. Madhow. Localizing multiple events using times of arrival: a parallelized, hierarchical approach to the association problem. *IEEE Transactions on Signal Processing*, 60(10):5464–5477, 2012.

[96] J. R. Vig. Introduction to quartz frequency standards. revision. Technical report, SLCET-TR-92-1 Army Research Laboratory, Electronics and Power Sources Directorate, 1992.

[97] P. Volgyesi, G. Balogh, A. Nadas, C. B. Nash, and Á Ledeczi. Shooter localization and weapon classification with soldier-wearable networked sensors. In *5th International Conference on Mobile Systems, Applications and Services, MobiSys 07*, pages 113–126, 2007.

[98] T. J. Walker. Acoustic synchrony: Two mechanisms in the snowy tree cricket. *Science*, 166(3907):891–894, 1969.

[99] Y. Wang and F. J. Doyle III. On influences of global and local cues on the rate of synchronization of oscillator networks. *Automatica*, 47(6):1236–1242, 2011.

[100] Y. Wang and F. J. Doyle III. Optimal phase response functions for fast pulse-coupled synchronization in wireless sensor networks. *IEEE Transactions on Signal Processing*, 60(10):5583–5588, 2012.

[101] Y. Wang, F. Núñez, and F. J. Doyle III. Energy-efficient pulse-coupled synchronization strategy design for wireless sensor networks through reduced idle listening. *IEEE Transactions on Signal Processing*, 60(10):5293–5306, 2012.

[102] Y. Wang, F. Núñez, and F. J. Doyle III. Increasing sync rate of pulse-coupled oscillators via phase response function design: Theory and application to wireless networks. *IEEE Transactions on Control Systems Technology*, 21(4):1455–1462, 2012.

[103] Y. Wang, F. Núñez, and F. J. Doyle III. Statistical analysis of the pulse-coupled synchronization strategy for wireless sensor networks. *IEEE Transactions on Signal Processing*, 61(21):5193–5204, 2013.

[104] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal. Firefly-inspired sensor network synchronicity with realistic radio effects. In *3rd ACM Conference on Embedded Networked Sensor Systems*, *SenSys '05*, pages 142–153, 2005.

[105] A. T. Winfree. Biological rhythms and the behavior of populations of coupled oscillators. *Journal of Theoretical Biology*, 16(1):15–42, 1967.

[106] A. T. Winfree. *The geometry of biological time*. Springer-Verlag, 2001.

[107] E. Xu, Z. Ding, and S. Dasgupta. Source localization in wireless sensor networks from signal time-of-arrival measurements. *IEEE Transactions on Signal Processing*, 59(6):2887–2897, 2011.

[108] X. Xu, S. Sahni, and N. S. V. Rao. On basic properties of localization using distance-difference measurements. In *11th International Conference on Information Fusion, FUSION 2008*, pages 1–8, 2008.

[109] K. Yang, G. Wang, and Z. Luo. Efficient convex relaxation methods for robust target localization by a sensor network using time differences of arrivals. *IEEE Transactions on Signal Processing*, 57(7):2775–2784, 2009.

Appendix A

Source Code for Implementation in Qualnet

A.1 Header file: mac_coupled.h

```
// Coupled MAC protocol based on control packages.
// Version 4.10 1-19-2014
//  Multi PRC - dead zone
#ifndef MAC_COUPLED_H
#define MAC_COUPLED_H

enum
{
    COUPLED_STATUS_PASSIVE,
    COUPLED_STATUS_CARRIER_SENSE,
    COUPLED_STATUS_BACKOFF,
    COUPLED_STATUS_XMIT,
    COUPLED_STATUS_IN_XMITING,
    COUPLED_STATUS_YIELD
};


#define COUPLED_TX_DATA_YIELD_TIME         (20 * MICRO_SECOND)

/* Used to experiment with COUPLED timers only. */
#define COUPLED_LOCAL_DATA_YIELD_TIME      (0)
#define COUPLED_REMOTE_DATA_YIELD_TIME     (0)

#define COUPLED_BO_MIN          (20 * MICRO_SECOND)
#define COUPLED_BO_MAX          (16 * COUPLED_BO_MIN)
#define COUPLED_COUPLING_PERIOD        (1 * NANO_SECOND)
#define COUPLED_WRITING_PERIOD        (100 * MICRO_SECOND)
```

```
#define COUPLED_TIMER_SWITCH    0x1  /* bit 0000 0001 is used for ON/OFF*/
#define COUPLED_TIMER_ON        0x1
#define COUPLED_TIMER_OFF       0x0

#define COUPLED_TIMER_TYPE      0xE  /* bit 0000 1110 is used for Timer type */
#define COUPLED_TIMER_BACKOFF   0x0
#define COUPLED_TIMER_YIELD     0x2
#define COUPLED_TIMER_COUPLING  0x4
#define COUPLED_TIMER_UNDEFINED 0xE

#define SYNCH_PACKET            0x1
#define SYNCH_PACKET_GLOBAL     0x2
#define DATA_PACKET             0x0

#define TANH_PRC 0x1
#define OPTIMAL_PRC_ID 0xA
#define OPTIMAL_PRC_NID 0xB
#define PESKIN_PRC 0xC
#define MS_PRC 0xD

typedef struct COUPLED_timer
{
    Int32 seq;
    unsigned char flag;
} COUPLEDTimer;

typedef struct COUPLED_header_str {
    Mac802Address sourceAddr;
    Mac802Address destAddr;
    int priority;
    int pktype;
} COUPLEDHeader;

typedef struct struct_mac_COUPLED_str
{
    MacData* myMacData;

    Int32 status;           /* status of layer COUPLED_STATUS_* */
    Int32 BOmin;            /* minimum backoff */
    Int32 BOmax;            /* maximum backoff */
    Int32 BOtimes;          /* how many times has it backoff ? */

    Int32 pktsToSend;
    Int32 pktsLostOverflow;

    Int32 pktsSentUnicast;
    Int32 pktsSentBroadcast;
```

```
    Int32 pktsSentSynch;
    int counter;

    Int32 pktsGotUnicast;
    Int32 pktsGotBroadcast;
    Mac802Address bcadd;
    MacHWAddress dest;
    MacHWAddress dest1;
    MacHWAddress dest2;
    double epsilong;
    double epsilonl;
    double globalst;
    double localst;
    double epsiloni;
    int signum;
    int isglobal;
    int CounterLimit;
    int Prc;
    int deadz;
    int deadzinit;
    int top;
    int second;

    COUPLEDTimer timer;
    COUPLEDTimer timercoup;

    RandomSeed seed;        /* for setting backoff timer */
} MacDataCOUPLED;

/*
 * FUNCTION    MacCOUPLEDInit
 * PURPOSE     Initialization function for COUPLED protocol of MAC layer.
 *
 * Parameters:
 *     node:      node being initialized.
 *     nodeInput: structure containing contents of input file
 */
void MacCOUPLEDInit(
    Node *node, int interfaceIndex, const NodeInput *nodeInput);

/*
 * FUNCTION    MacCOUPLEDLayer
 * PURPOSE     Models the behaviour of the MAC layer with the COUPLED protocol
 *             on receiving the message enclosed in msgHdr.
 *
 * Parameters:
 *     node:     node which received the message
 *     msgHdr:   message received by the layer
```

```
 */
void MacCOUPLEDLayer(
    Node *node, int interfaceIndex, Message *msg);

/*
 * FUNCTION    MacCOUPLEDFinalize
 * PURPOSE     Called at the end of simulation to collect the results of
 *             the simulation of COUPLED protocol of the MAC Layer.
 *
 * Parameter:
 *     node:     node for which results are to be collected.
 */
void MacCOUPLEDFinalize(Node *node, int interfaceIndex);

/*
 * FUNCTION    MacCOUPLEDNetworkLayerHasPacketToSend
 * PURPOSE     To tell COUPLED that the network layer has a packet to send.
 */

void MacCOUPLEDNetworkLayerHasPacketToSend(Node *node, MacDataCOUPLED *COUPLED);

void MacCOUPLEDReceivePacketFromPhy(
    Node* node, MacDataCOUPLED* COUPLED, Message* msg);

void MacCOUPLEDReceivePhyStatusChangeNotification(
   Node* node,
   MacDataCOUPLED* COUPLED,
   PhyStatusType oldPhyStatus,
   PhyStatusType newPhyStatus);

#endif
```

A.2 Source file: mac_coupled.cpp

```
 // Coupled MAC protocol based on control packages.
// Version 4.10 1-19-2014
//  Multi PRC - dead zone

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <algorithm>
#include <cmath>

#include "api.h"
#include "mac_coupled.h"
#include "network_ip.h"
#include "partition.h"
#include "phy_802_11.h"

using namespace std;

static /*inline*/
PhyStatusType PhyStatus(Node* node, MacDataCOUPLED* COUPLED)
{
   return PHY_GetStatus(node, COUPLED->myMacData->phyNumber);
}

/*
 * NAME:        MacCOUPLEDHandlePromiscuousMode.
 *
 * PURPOSE:     Supports promiscuous mode sending remote packets to
 *              upper layers.
 *
 * PARAMETERS:  node, node using promiscuous mode.
 *              frame, packet to send to upper layers.
 *
 * RETURN:      None.
 *
 * ASSUMPTION:  node != NULL.
 */

static
void MacCOUPLEDHandlePromiscuousMode(Node *node,
                                     MacDataCOUPLED* COUPLED,
                                     Message *frame,
                                     Mac802Address prevHop,
                                     Mac802Address destAddr)
{
    MacHWAddress prevHopHWAddr;
```

```
    MacHWAddress destHWAddr;

    MESSAGE_RemoveHeader(node, frame, sizeof(COUPLEDHeader), TRACE_COUPLED);

    Convert802AddressToVariableHWAddress(node, &prevHopHWAddr, &prevHop);

    Convert802AddressToVariableHWAddress(node, &destHWAddr, &destAddr);

    MAC_SneakPeekAtMacPacket(node,
                             COUPLED->myMacData->interfaceIndex,
                             frame,
                             prevHopHWAddr,
                             destHWAddr);

    MESSAGE_AddHeader(node, frame, sizeof(COUPLEDHeader), TRACE_COUPLED);
}

/*
 * NAME:        MacCOUPLEDDataXmit.
 *
 * PURPOSE:     Sending data frames to destination.
 *
 * PARAMETERS:  node, node sending the data frame.
 *
 * RETURN:      None.
 *
 * ASSUMPTION:  node != NULL.
 */

static
void MacCOUPLEDXmit(Node *node, MacDataCOUPLED *COUPLED)
{
    Message *msg;
    MacHWAddress destHWAddr;
    int networkType;
    TosType priority;

    COUPLEDHeader         *hdr;

    assert(COUPLED->status == COUPLED_STATUS_XMIT);

    /*
     * Dequeue packet which was received from the
     * network layer.
     */

    MAC_OutputQueueDequeuePacket(
        node, COUPLED->myMacData->interfaceIndex,
```

```
        &msg, &destHWAddr, &networkType, &priority);

    if (msg == NULL)
    {
        #ifdef QDEBUG
            printf("COUPLED: Queue should not be empty...\n");
        #endif


        // The Queue can be empty if the packet was dropped forcefully by

        // routing protocol. Set the correct COUPLED state in this case

        if(COUPLED->BOtimes >0)

        {
            COUPLED->status = COUPLED_STATUS_BACKOFF;

        }
        else
        {
            COUPLED->status = COUPLED_STATUS_PASSIVE;

        }

        return;
    }

    COUPLED->status = COUPLED_STATUS_IN_XMITING;
    COUPLED->timer.flag = COUPLED_TIMER_OFF | COUPLED_TIMER_UNDEFINED;

    /*
     * Assign other fields to packet to be sent
     * to phy layer.
     */

    MESSAGE_AddHeader(node, msg, sizeof(COUPLEDHeader), TRACE_COUPLED);

    hdr = (COUPLEDHeader *) msg->packet;

    ConvertVariableHWAddressTo802Address(node, &destHWAddr, &hdr->destAddr);

    ConvertVariableHWAddressTo802Address(
                  node,
                  &node->macData[COUPLED->myMacData->interfaceIndex]->macHWAddr,
                  &hdr->sourceAddr);

    hdr->priority = priority;
```

```
    hdr->pktype = DATA_PACKET;

    PHY_StartTransmittingSignal(
        node, COUPLED->myMacData->phyNumber,
        msg, FALSE, 0);

    if (MAC_IsBroadcastMac802Address(&hdr->destAddr)) {
        COUPLED->pktsSentBroadcast++;
    }
    else {
        COUPLED->pktsSentUnicast++;
    }
}

/*
 * NAME:        MacCOUPLEDSetTimer.
 *
 * PURPOSE:     Set a timer for node to expire at time timerValue.
 *
 * PARAMETERS:  node, node setting the timer.
 *              timerType, what type of timer is being set.
 *              delay, when timer is to expire.
 *
 * RETURN:      None.
 *
 * ASSUMPTION:  node != NULL.
 */

static
void MacCOUPLEDSetTimer(
    Node *node, MacDataCOUPLED* COUPLED, int timerType, clocktype delay)
{
    Message      *newMsg;
    int         *timerSeq;

    COUPLED->timer.flag = (unsigned char)(COUPLED_TIMER_ON | timerType);
    COUPLED->timer.seq++;

    assert((timerType == COUPLED_TIMER_BACKOFF) ||
           (timerType == COUPLED_TIMER_YIELD));

    newMsg = MESSAGE_Alloc(node, MAC_LAYER, 0,
                           MSG_MAC_TimerExpired);
    MESSAGE_SetInstanceId(newMsg, (short)COUPLED->myMacData->interfaceIndex);

    MESSAGE_InfoAlloc(node, newMsg, sizeof(COUPLED->timer.seq));
    timerSeq  = (int *) MESSAGE_ReturnInfo(newMsg);
    *timerSeq = COUPLED->timer.seq;
```

```
    MESSAGE_Send(node, newMsg, delay);
}

/*
 * NAME:        MacCOUPLEDSetTimerCoupling.
 *
 * PURPOSE:     Set a timer for node to expire at time timerValue.
 *
 * PARAMETERS:  node, node setting the timer.
 *              timerType, what type of timer is being set.
 *              delay, when timer is to expire.
 *
 * RETURN:      None.
 *
 * ASSUMPTION:  node != NULL.
 */

static
void MacCOUPLEDSetTimerCoupling(
   Node *node, MacDataCOUPLED* COUPLED)
{
    Message      *newMsg;
    int         *timerSeq;

    COUPLED->timercoup.seq++;
    newMsg = MESSAGE_Alloc(node, MAC_LAYER, 0,
                           MSG_MAC_FrameStartOrEnd);
    MESSAGE_SetInstanceId(newMsg, (short)COUPLED->myMacData->interfaceIndex);

    MESSAGE_InfoAlloc(node, newMsg, sizeof(COUPLED->timer.seq));
    timerSeq  = (int *) MESSAGE_ReturnInfo(newMsg);
    *timerSeq = COUPLED->timer.seq;

    MESSAGE_Send(node, newMsg,10000*COUPLED_COUPLING_PERIOD);
}

static
void MacCOUPLEDSetTimerWriting(
   Node *node, MacDataCOUPLED* COUPLED)
{
    Message      *newMsg;
    int         *timerSeq;

    COUPLED->timercoup.seq++;
    newMsg = MESSAGE_Alloc(node, MAC_LAYER, 0,
                           MSG_SPECIAL_Timer);
    MESSAGE_SetInstanceId(newMsg, (short)COUPLED->myMacData->interfaceIndex);
```

```
    MESSAGE_InfoAlloc(node, newMsg, sizeof(COUPLED->timer.seq));
    timerSeq  = (int *) MESSAGE_ReturnInfo(newMsg);
    *timerSeq = COUPLED->timer.seq;

    MESSAGE_Send(node, newMsg, COUPLED_WRITING_PERIOD);
}

/*
 * NAME:        MacCOUPLEDYield.
 *
 * PURPOSE:     Yield so neighboring nodes can transmit or receive.
 *
 * PARAMETERS:  node, node that is yielding.
 *              holding, how int to yield for.
 *
 * RETURN:      None.
 *
 * ASSUMPTION:  node != NULL.
 */

static
void MacCOUPLEDYield(Node *node, MacDataCOUPLED *COUPLED, clocktype holding)
{
    assert(COUPLED->status == COUPLED_STATUS_YIELD);

    MacCOUPLEDSetTimer(node, COUPLED, COUPLED_TIMER_YIELD, holding);
}

/*
 * NAME:        MacCOUPLEDBackoff.
 *
 * PURPOSE:     Backing off sending data at a later time.
 *
 * PARAMETERS:  node, node that is backing off.
 *
 * RETURN:      None.
 *
 * ASSUMPTION:  node != NULL.
 */

static
void MacCOUPLEDBackoff(Node *node, MacDataCOUPLED *COUPLED)
{
    clocktype randTime;
    assert(COUPLED->status == COUPLED_STATUS_BACKOFF);

    randTime = (RANDOM_nrand(COUPLED->seed) % COUPLED->BOmin) + 1;
```

```
    COUPLED->BOmin = COUPLED->BOmin * 2;

    if (COUPLED->BOmin > COUPLED->BOmax) {
        COUPLED->BOmin = COUPLED->BOmax;
    }

    COUPLED->BOtimes++;

    MacCOUPLEDSetTimer(node, COUPLED, COUPLED_TIMER_BACKOFF, randTime);
}

/*
 * NAME:        UpdateCounter.
 *
 * PURPOSE:     Set timer for next coupling message at init.
 *
 * PARAMETERS:  node, node.
 *
 * RETURN:      None.
 *
 * ASSUMPTION:  node != NULL.
 */

static
void UpdateCounterOpId(Node *node, MacDataCOUPLED *COUPLED, double epsilon,
double strenght)
{
    int deltacounter;
    int currentcounter;
    int mid=0;
    currentcounter = COUPLED->counter;
    if (currentcounter < COUPLED->CounterLimit/2) {
mid=1;
}
    deltacounter = int(strenght*floor(COUPLED->CounterLimit-currentcounter-
    COUPLED->CounterLimit*mid));
    COUPLED->counter = currentcounter + deltacounter;
}

/*
 * NAME:        MacCOUPLEDSetSynchPk.
 *
 * PURPOSE:     Set a Synch Pk for node to Send at delay.
 *
 * PARAMETERS:  node, node setting the timer.
 *              timerType, what type of timer is being set.
 *              delay, when timer is to expire.
 *
```

```
 * RETURN:      None.
 *
 * ASSUMPTION:  node != NULL.
 */

static
void MacCOUPLEDSetSynchPk(
    Node *node, MacDataCOUPLED* COUPLED)
{
    Message      *newMsg;
    COUPLEDHeader      *hdr;
    COUPLED->pktsSentSynch++;
    COUPLED->counter = 0;
    COUPLED->status = COUPLED_STATUS_IN_XMITING;
    newMsg = MESSAGE_Alloc(node, 0, 0, 0);
    MESSAGE_PacketAlloc(node,
                        newMsg,
                        sizeof(COUPLEDHeader),
                        TRACE_COUPLED);

    hdr = (COUPLEDHeader *) MESSAGE_ReturnPacket(newMsg);
    ConvertVariableHWAddressTo802Address(node,
    &node->macData[COUPLED->myMacData->interfaceIndex]->macHWAddr, &hdr->sourceAddr);

    hdr->destAddr = COUPLED->bcadd;
    hdr->priority = 0;
    if (COUPLED->isglobal == 1){
    hdr->pktype = SYNCH_PACKET_GLOBAL;
    }
    else {
    hdr->pktype = SYNCH_PACKET;
    }
    PHY_StartTransmittingSignal(node, COUPLED->myMacData->phyNumber, newMsg,
    FALSE, 0);
    COUPLED->second++;
    clocktype curTime;
    char buf[80];
    curTime = getSimTime(node);
    ctoa(curTime,buf);
    char buf1[MAX_STRING_LENGTH];
    sprintf(buf1, "%uSEED%d",
            node->nodeId, node->globalSeed);
    FILE * pFile;
    pFile = fopen (buf1,"a");
    fprintf (pFile, "%s, ",buf);
    fclose (pFile);
}
```

```
static //inline//
void CheckPhyStatusAndSendOrBackoff(Node* node, MacDataCOUPLED* COUPLED) {
    /* Carrier sense response from phy. */

    if ((PhyStatus(node, COUPLED) == PHY_IDLE) &&
        (COUPLED->status != COUPLED_STATUS_IN_XMITING))
    {
        COUPLED->status = COUPLED_STATUS_XMIT;
        MacCOUPLEDXmit(node, COUPLED);
    }
    else {
        if (!MAC_OutputQueueIsEmpty(
                node, COUPLED->myMacData->interfaceIndex))
        {
            COUPLED->status = COUPLED_STATUS_BACKOFF;
            MacCOUPLEDBackoff(node, COUPLED);
        }
    }
}

/*
 * NAME:        MacCOUPLEDNetworkLayerHasPacketToSend.
 *
 * PURPOSE:     In passive mode, start process to send data; else return;
 *
 * RETURN:      None.
 *
 */

void MacCOUPLEDNetworkLayerHasPacketToSend(Node *node, MacDataCOUPLED *COUPLED)
{
    if (COUPLED->status == COUPLED_STATUS_PASSIVE) {
        CheckPhyStatusAndSendOrBackoff(node, COUPLED);
    }//if//
}

/*
 * NAME:        MacCOUPLEDPassive.
 *
 * PURPOSE:     In passive mode, check whether there is a local packet.
 *              If YES, send data; else return;
 *
 * PARAMETERS:  node, node that is in passive state.
 *
 * RETURN:      None.
 *
 * ASSUMPTION:  node != NULL.
 */
```

```
static
void MacCOUPLEDPassive(Node *node, MacDataCOUPLED *COUPLED)
{
    if ((COUPLED->status == COUPLED_STATUS_PASSIVE) &&
        (!MAC_OutputQueueIsEmpty(node, COUPLED->myMacData->interfaceIndex)))
    {
        MacCOUPLEDNetworkLayerHasPacketToSend(node, COUPLED);
    }
}

/*
 * NAME:        MacCOUPLEDPrintStats
 *
 * PURPOSE:     Print MAC layer statistics.
 *
 * PARAMETERS:  node.
 *
 * RETURN:      None.
 *
 * ASSUMPTION:  node != NULL.
 */

static
void MacCOUPLEDPrintStats(Node *node, MacDataCOUPLED* COUPLED, int interfaceIndex)
{
    char buf[MAX_STRING_LENGTH];

    sprintf(buf, "Packets from network = %d",
            COUPLED->pktsToSend);
    IO_PrintStat(node, "Mac", "COUPLED", ANY_DEST, interfaceIndex, buf);

    sprintf(buf, "Packets lost due to buffer overflow = %d",
            COUPLED->pktsLostOverflow);
    IO_PrintStat(node, "Mac", "COUPLED", ANY_DEST, interfaceIndex, buf);

    sprintf(buf, "UNICAST packets sent to channel = %d",
            COUPLED->pktsSentUnicast);
    IO_PrintStat(node, "Mac", "COUPLED", ANY_DEST, interfaceIndex, buf);

    sprintf(buf, "BROADCAST packets sent to channel = %d",
            COUPLED->pktsSentBroadcast);
    IO_PrintStat(node, "Mac", "COUPLED", ANY_DEST, interfaceIndex, buf);
    sprintf(buf, "UNICAST packets received = %d",
            COUPLED->pktsGotUnicast);
    IO_PrintStat(node, "Mac", "COUPLED", ANY_DEST, interfaceIndex, buf);

    sprintf(buf, "BROADCAST packets received = %d",
```

```
            COUPLED->pktsGotBroadcast);
    IO_PrintStat(node, "Mac", "COUPLED", ANY_DEST, interfaceIndex, buf);

    sprintf(buf, "CURRENT counter = %d",
            COUPLED->counter);
    IO_PrintStat(node, "Mac", "COUPLED", ANY_DEST, interfaceIndex, buf);

    sprintf(buf, "SYNCH packets sent = %d",
            COUPLED->pktsSentSynch);
    IO_PrintStat(node, "Mac", "COUPLED", ANY_DEST, interfaceIndex, buf);
}

/*
 * FUNCTION    MacCOUPLEDInit
 * PURPOSE     Initialization function for COUPLED protocol of MAC layer.

 * Parameters:
 *     node:      node being initialized.
 *     nodeInput: structure containing contents of input file
 */

void MacCOUPLEDInit(
   Node *node, int interfaceIndex, const NodeInput *nodeInput)
{
MacDataCOUPLED *COUPLED = (MacDataCOUPLED *) MEM_malloc(sizeof(MacDataCOUPLED));

    assert(COUPLED != NULL);

    memset(COUPLED, 0, sizeof(MacDataCOUPLED));
    COUPLED->myMacData = node->macData[interfaceIndex];
    COUPLED->myMacData->macVar = (void *)COUPLED;

    COUPLED->timer.flag = COUPLED_TIMER_ON | COUPLED_TIMER_UNDEFINED;
    COUPLED->timer.seq = 0;
    COUPLED->timercoup.flag = COUPLED_TIMER_ON;
    COUPLED->timercoup.seq = 0;

    COUPLED->status = COUPLED_STATUS_PASSIVE;
    COUPLED->BOmin = COUPLED_BO_MIN;
    COUPLED->BOmax = COUPLED_BO_MAX;
    COUPLED->BOtimes = 0;

    COUPLED->pktsToSend = 0;
    COUPLED->pktsLostOverflow = 0;

    COUPLED->pktsSentUnicast = 0;
    COUPLED->pktsSentBroadcast = 0;
    COUPLED->pktsSentSynch = 0;
```

```
COUPLED->pktsGotUnicast = 0;
COUPLED->pktsGotBroadcast = 0;
COUPLED->second = 0;

COUPLED->bcadd = ANY_MAC802;
COUPLED->top = 0;
BOOL temp;
Address address;
NetworkGetInterfaceInfo(node, interfaceIndex, &address, NETWORK_IPV4);
IO_ReadDouble(node->nodeId, &address, nodeInput,
"EPSILON_G", &temp, &COUPLED->epsilong);
IO_ReadDouble(node->nodeId, &address, nodeInput,
"EPSILON_L", &temp, &COUPLED->epsilonl);
IO_ReadDouble(node->nodeId, &address, nodeInput,
"COUPLED_GLOBAL_STRENGHT", &temp, &COUPLED->globalst);
IO_ReadDouble(node->nodeId, &address, nodeInput,
"COUPLED_LOCAL_STRENGHT", &temp, &COUPLED->localst);
IO_ReadDouble(node->nodeId, &address, nodeInput,
"EPSILON_I", &temp, &COUPLED->epsiloni);
IO_ReadInt(node->nodeId, &address, nodeInput,
"SIGNUM", &temp, &COUPLED->signum);
IO_ReadInt(node->nodeId, &address, nodeInput,
"COUPLED_IS_GLOBAL", &temp, &COUPLED->isglobal);
IO_ReadInt(node->nodeId, &address, nodeInput,
"COUPLED_COUNTER_LIMIT", &temp, &COUPLED->CounterLimit);
IO_ReadInt(node->nodeId, &address, nodeInput,
"COUPLED_PRC", &temp, &COUPLED->Prc);
IO_ReadInt(node->nodeId, &address, nodeInput,
"DEAD_ZONE", &temp, &COUPLED->deadz);
COUPLED->deadzinit = COUPLED->deadz;
printf("Node %u, %u", node->nodeId, COUPLED->Prc);
printf("Node %u, %u ", node->nodeId, address);
MAC_PrintHWAddr(&destiny);
RANDOM_SetSeed(COUPLED->seed,
               node->globalSeed,
               node->nodeId,
               MAC_PROTOCOL_COUPLED,
               interfaceIndex);

int randTime;
randTime = int(RANDOM_nrand(COUPLED->seed)%int(COUPLED->CounterLimit));
COUPLED->counter = randTime;
MacCOUPLEDSetTimerCoupling(node, COUPLED);
MacCOUPLEDSetTimerWriting(node, COUPLED);

char buf1[MAX_STRING_LENGTH];
sprintf(buf1, "%uSEED%d",
```

```
            node->nodeId, node->globalSeed);
    char buf3[MAX_STRING_LENGTH];
    sprintf(buf3, "%uSEEDTIMES%d",
            node->nodeId, node->globalSeed);
    char buf2[MAX_STRING_LENGTH];
    sprintf(buf2, "%uSEED%dRECEPTIONS",
            node->nodeId, node->globalSeed);
    FILE * pFile;
    pFile = fopen (buf1,"a");
 fprintf(pFile, "%lf, %lf, %d, %lf, %d, %d\n", COUPLED->epsilong,
 COUPLED->epsilonl, COUPLED->deadz, COUPLED->localst, COUPLED->isglobal,
 COUPLED->counter);
    fclose (pFile);
FILE * pFile1;
    pFile1 = fopen (buf2,"a");
 fprintf(pFile1, "%lf, %lf, %lf, %lf, %d, %d\n", COUPLED->epsilong,
 COUPLED->epsilonl, COUPLED->globalst, COUPLED->localst, COUPLED->isglobal,
 COUPLED->CounterLimit);
    fclose (pFile1);
    FILE * pFile2;
    pFile2 = fopen (buf3,"a");
 fprintf(pFile2, "%lf, %lf, %lf, %lf, %d, %d\n", COUPLED->epsilong,
 COUPLED->epsilonl, COUPLED->globalst, COUPLED->localst, COUPLED->isglobal,
 COUPLED->CounterLimit);
    fclose (pFile2);

#ifdef PARALLEL //Parallel
    PARALLEL_SetProtocolIsNotEOTCapable(node);
    PARALLEL_SetMinimumLookaheadForInterface(node, 0);
#endif //endParallel
}



void MacCOUPLEDReceivePacketFromPhy(
    Node* node, MacDataCOUPLED* COUPLED, Message* msg)
{
    if (COUPLED->status == COUPLED_STATUS_IN_XMITING) {
        MESSAGE_Free(node, msg);
        return;
    }//if//

    switch (COUPLED->status) {
    case COUPLED_STATUS_PASSIVE:
    case COUPLED_STATUS_CARRIER_SENSE:
    case COUPLED_STATUS_BACKOFF:
    case COUPLED_STATUS_YIELD: {
        int interfaceIndex = COUPLED->myMacData->interfaceIndex;
```

```
        COUPLEDHeader *hdr = (COUPLEDHeader *) msg->packet;

        MacHWAddress  destHWAddress;

        Convert802AddressToVariableHWAddress(node, &destHWAddress,
                                                          &hdr->destAddr);

       if (MAC_IsMyAddress(node, &destHWAddress)) {
            COUPLED->pktsGotUnicast++;

              if ((hdr->pktype == SYNCH_PACKET) && (COUPLED->isglobal == 0)) {

if (COUPLED->counter > COUPLED->deadz) {

        UpdateCounterOpId(node, COUPLED, COUPLED->epsilonl, COUPLED->localst);

}//deadzone
}//if noglobal
        }

       else if (MAC_IsBroadcastMac802Address(&hdr->destAddr))
        {
            COUPLED->pktsGotBroadcast++;

              if ((hdr->pktype == SYNCH_PACKET) && (COUPLED->isglobal == 0)) {

if (COUPLED->counter > COUPLED->deadz) {

    clocktype curTime;
char buf[80];
    curTime = getSimTime(node);
    ctoa(curTime,buf);
    char buf2[MAX_STRING_LENGTH];
    sprintf(buf2, "%u",
            &hdr->sourceAddr);
    char buf1[MAX_STRING_LENGTH];
    sprintf(buf1, "%uSEED%dRECEPTIONS",
            node->nodeId, node->globalSeed);
    FILE * pFile;
    pFile = fopen (buf1,"a");
    fprintf (pFile, "%s, %u, %s, 0\n",buf2,&hdr->destAddr,buf);
    fclose (pFile);
        UpdateCounterOpId(node, COUPLED, COUPLED->epsilonl, COUPLED->localst);
            //printf("Node %u, Counter Updated global\n", node->nodeId);

}//deadzone
}//if noglobal
        }
```

```
        if (MAC_IsMyAddress(node, &destHWAddress) ||
                            MAC_IsBroadcastHWAddress(&destHWAddress))
         {

              MacHWAddress srcHWAddress;
              Convert802AddressToVariableHWAddress(node, &srcHWAddress,
                                                     &hdr->sourceAddr);

          MESSAGE_RemoveHeader(node, msg, sizeof(COUPLEDHeader), TRACE_COUPLED);

             MAC_HandOffSuccessfullyReceivedPacket(node,
                COUPLED->myMacData->interfaceIndex, msg, &srcHWAddress);
         }
         else {
             if (node->macData[interfaceIndex]->promiscuousMode) {
                 MacCOUPLEDHandlePromiscuousMode(node,
                                          COUPLED,
                                          msg,
                                          hdr->sourceAddr,
                                          hdr->destAddr);
             }
             MESSAGE_Free(node, msg);
         }
         break;
     }
     default:
         MESSAGE_Free(node, msg);
         printf("MAC_COUPLED: Error with node %u, status %d.\n",
                node->nodeId, COUPLED->status);
         assert(FALSE); abort();
     }//switch//
}

void MacCOUPLEDReceivePhyStatusChangeNotification(
   Node* node,
   MacDataCOUPLED* COUPLED,
   PhyStatusType oldPhyStatus,
   PhyStatusType newPhyStatus)
{
   if (oldPhyStatus == PHY_TRANSMITTING) {
      assert(newPhyStatus != PHY_TRANSMITTING);
      assert(COUPLED->status == COUPLED_STATUS_IN_XMITING);

      COUPLED->BOmin = COUPLED_BO_MIN;
      COUPLED->BOmax = COUPLED_BO_MAX;
      COUPLED->BOtimes = 0;
      COUPLED->status = COUPLED_STATUS_YIELD;
```

```
      MacCOUPLEDYield(node, COUPLED, (clocktype)COUPLED_TX_DATA_YIELD_TIME);
   }//if//
}

/*
 * FUNCTION    MacCOUPLEDLayer
 * PURPOSE     Models the behaviour of the MAC layer with the COUPLED protocol
 *             on receiving the message enclosed in msg.
 *
 * Parameters:
 *     node:     node which received the message
 *     msg:      message received by the layer
 */

void MacCOUPLEDLayer(Node *node, int interfaceIndex, Message *msg)
{
    /*
     * Retrieve the pointer to the data portion which relates
     * to the COUPLED protocol.
     */

MacDataCOUPLED *COUPLED = (MacDataCOUPLED *)node->macData[interfaceIndex]->macVar;
    int seq_num;
    if (msg->eventType == MSG_MAC_FrameStartOrEnd) {
        MESSAGE_Free(node, msg);
        COUPLED->counter++;
        MacCOUPLEDSetTimerCoupling(node, COUPLED);
        if (COUPLED->counter > COUPLED->CounterLimit) {
        MacCOUPLEDSetSynchPk(node, COUPLED);
        }
    }
    else if (msg->eventType == MSG_SPECIAL_Timer) {
    clocktype curTime;
    char buf[80];
    curTime = getSimTime(node);
    ctoa(curTime,buf);
    char buf1[MAX_STRING_LENGTH];
    sprintf(buf1, "%uSEEDTIMES%d",
            node->nodeId, node->globalSeed);
    FILE * pFile;
    pFile = fopen (buf1,"a");
    fprintf (pFile, "%s, ",buf);
    fprintf(pFile, "%d, ", COUPLED->second);
    fprintf(pFile, "%d\n", COUPLED->counter);
    fclose (pFile);

    seq_num = *((int *) MESSAGE_ReturnInfo(msg));
    MESSAGE_Free(node, msg);
```

```
    MacCOUPLEDSetTimerWriting(node, COUPLED);
    }

    else {

    assert(msg->eventType == MSG_MAC_TimerExpired);

    seq_num = *((int *) MESSAGE_ReturnInfo(msg));

    MESSAGE_Free(node, msg);
    if ((seq_num < COUPLED->timer.seq) ||
        ((COUPLED->timer.flag & COUPLED_TIMER_SWITCH) == COUPLED_TIMER_OFF)) {
        return;
    }

    if (seq_num > COUPLED->timer.seq) {
        assert(FALSE);
    }

    assert(((COUPLED->timer.flag & COUPLED_TIMER_TYPE) == COUPLED_TIMER_BACKOFF) ||
           ((COUPLED->timer.flag & COUPLED_TIMER_TYPE) == COUPLED_TIMER_YIELD));

    switch(COUPLED->timer.flag & COUPLED_TIMER_TYPE) {
    case COUPLED_TIMER_BACKOFF:
    {
        COUPLED->timer.flag = COUPLED_TIMER_OFF | COUPLED_TIMER_UNDEFINED;

        CheckPhyStatusAndSendOrBackoff(node, COUPLED);

        break;
    }

    case COUPLED_TIMER_YIELD:
        COUPLED->timer.flag = COUPLED_TIMER_OFF | COUPLED_TIMER_UNDEFINED;
        COUPLED->status = COUPLED_STATUS_PASSIVE;
        MacCOUPLEDPassive(node, COUPLED);
        break;

    default:
        assert(FALSE); abort();
        break;
    }/*switch*/
   }
}

/*
 * FUNCTION    MAC_Finalize
 * PURPOSE     Called at the end of simulation to collect the results of
```

```
 *             the simulation of the COUPLED protocol of MAC Layer.
 *
 * Parameter:
 *     node:     node for which results are to be collected.
 */

void MacCOUPLEDFinalize(Node *node, int interfaceIndex)
{
MacDataCOUPLED* COUPLED = (MacDataCOUPLED *)node->macData[interfaceIndex]->macVar;

    if (node->macData[interfaceIndex]->macStats == TRUE) {
        MacCOUPLEDPrintStats(node, COUPLED, interfaceIndex);
    }
    char buf1[MAX_STRING_LENGTH];
    sprintf(buf1, "%uSEED%d",
            node->nodeId, node->globalSeed);
    FILE * pFile;
    pFile = fopen (buf1,"a");
    fprintf(pFile, "\n%d\n", COUPLED->counter);
    fclose (pFile);

}
```

Appendix B

Source Code for Implementation in Gumstix Boards and BMS Sensors

B.1 Header file: sync.h

```
extern pthread_mutex_t count_mutex;
extern pthread_cond_t c_var;

void *UDP_send(void *ptr);

void *UDP_rec(void *ptr);

int spawn_thread(int argc, char *argv[]);

long double get_time();

void *counter(void *args);
```

B.2 Source file: udp.cpp

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#include <netdb.h>
#include <stdlib.h>
#include <string.h>
#include <string>
#include <unistd.h>
#include <sys/time.h>
#include <pthread.h>
#include <time.h>
#include "sync.h"

#define MAXCOUNT 10000
#define lambda 0.8
#define delta 2
#define half MAXCOUNT/2

bool x;

pthread_mutex_t count_mutex = PTHREAD_MUTEX_INITIALIZER, count_mutex2 =
PTHREAD_MUTEX_INITIALIZER;
pthread_cond_t c_var = PTHREAD_COND_INITIALIZER, c_var2 =
PTHREAD_COND_INITIALIZER;

struct info{
  std::string IP;
  int port;
};

struct timespec tim = {0, 1*100*1000};

long int seconds = 0;
int i = 0;

int spawn_thread(int argc, char *argv[]){
  //Argument checker; requires a port to broadcast to.
  if (argc !=2) {
    printf("syntax: <port_number> \n");
    exit(-1);
  }
  info* a = new info;
  a->port = (atoi(argv[1]));
  info* b = new info;
```

```
  b->port = (atoi(argv[1]));
  pthread_t t1, t2, t3;
  long* t;
  pthread_cond_init (&c_var, NULL);
  pthread_mutex_init(&count_mutex, NULL);
  pthread_create(&t1, NULL, UDP_send, a);
  pthread_create(&t2, NULL, UDP_rec, b);
  pthread_create(&t3, NULL, counter, t);
  return 0;
}

void *counter(void *args){
  struct tm *current;
  struct timeval detail_time;
  time_t now;
  bool overflow = false;
  sleep(1);
  while(1){
    while((x == false) && (overflow == false)){
      //THIS FOR LOOP CONTROLS STEP SIZE
      for(int x=0;x<10227;x++)
;
      i++;
      if(i >= MAXCOUNT)
overflow = true;
    }

    if(x == true){
      printf("counter: %i\n", i);
      if(i <delta)
;
      else if(i < half){
i = (i - (i*lambda));

      }
      else{
i = i+(lambda*(MAXCOUNT-i));

      }
      x = false;

    }
    if(overflow == true){
      pthread_cond_signal(&c_var);
      seconds++;
      i = 0;
      overflow = false;
      //THIS IF STATEMENT CONTROLS TIMER.
```

```
      if(seconds > 3600){
pthread_cond_signal(&c_var2);
      }
    }
  }
}


void *UDP_rec(void *ptr){
  struct info *c = (struct info *) ptr;
  sockaddr_in si_me, si_other;
  int s;
  s=socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);
  int broadcast=1;
  char buf[40];
  memset(&si_me, 0, sizeof(si_me));
  si_me.sin_family = AF_INET;
  si_me.sin_port = htons(c->port);
  si_me.sin_addr.s_addr = INADDR_ANY;

  bind(s, (sockaddr *)&si_me, sizeof(sockaddr));
  setsockopt(s, SOL_SOCKET, SO_BROADCAST, &broadcast, sizeof broadcast);
  int sel = 0;
  fd_set readfds;
  struct timeval tv;
  unsigned int slen = sizeof(struct sockaddr_in);
  FD_ZERO(&readfds);
  FD_SET(s, &readfds);
  while(1){
    tv.tv_sec = 3;
    sel = select(s+1, &readfds, NULL, NULL, &tv);
    if (sel == 1){
      recvfrom(s, buf, sizeof(buf)-1, 0, (sockaddr *)&si_other, &slen);
      x = true;
    }
  }
}

void *UDP_send(void *ptr){
  struct info *c = (struct info *) ptr;
  std::string str = "r";
  int datalen = 1;
  const char *databuf;
  databuf = str.c_str();
  time_t now;
  struct tm *current;
  struct timeval detail_time;
  int sock;
```

```
  struct sockaddr_in broadcastAddr;
  char *broadcastIP;
  unsigned short broadcastPort;
  char *sendString;
  int broadcastPermission;
  int sendStringLen;

  if ((sock = socket(PF_INET, SOCK_DGRAM, IPPROTO_UDP)) < 0){
    fprintf(stderr, "socket error");
    exit(1);
  }
  char loopch=0;
  if (setsockopt(sock, IPPROTO_IP, IP_MULTICAST_LOOP,
 (char *)&loopch, sizeof(loopch)) < 0) {
    perror("setting IP_MULTICAST_LOOP:");
    close(sock);
    exit(1);
  }
  broadcastPermission = 1;
  if (setsockopt(sock, SOL_SOCKET, SO_BROADCAST, (void *) &
  broadcastPermission,sizeof(broadcastPermission)) < 0){
    fprintf(stderr, "setsockopt error");
    exit(1);
  }
  memset(&broadcastAddr, 0, sizeof(broadcastAddr));
  broadcastAddr.sin_family = AF_INET;
  broadcastAddr.sin_addr.s_addr = inet_addr("10.42.43.255");
  broadcastAddr.sin_port = htons(c->port);
  while(1){
    pthread_cond_wait(&c_var, &count_mutex);
    sendto(sock, databuf, 1, 0, (struct sockaddr *)&
    broadcastAddr, sizeof(broadcastAddr));
    now = time(0);
    current = localtime(&now);
    gettimeofday(&detail_time,NULL);
    printf("sent packet at %i.%06lu seconds\n",current->tm_sec,
   detail_time.tv_usec);
  }
}

long double get_time(){
  float decimal = i;
  long double time = seconds;
  decimal = decimal / 10000;
  time = time + decimal;
  return time;
}
```

B.3 Usage Example file: test.cpp

```
include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#include <netdb.h>
#include <stdlib.h>
#include <string.h>
#include <string>
#include <unistd.h>
#include <sys/time.h>
#include <pthread.h>
#include <time.h>
#include "sync.h"


int main(int argc, char *argv[]){
  long double time;
  spawn_thread(argc, argv); //starts the wireless synchronization algorithm
  while(1){
    sleep(3) //sleep for three seconds
    time = get_time(); //variable time now has local time
  }
  return 0;
}
```