TESIS DE LA UNIVERSIDAD DE ZARAGOZA

2025 171

Eduardo Sebastián Rodríguez

Distributed Multi-robot Control: Physics, Geometry and Learning

Director/es
Sagüés Blázquiz, Carlos
Montijano Muñoz, Eduardo

ISSN 2254-7606

Prensas de la Universidad Universidad Zaragoza

Universidad de Zaragoza
Servicio de Publicaciones

ISSN 2254-7606

Universidad Zaragoza
1542

Tesis Doctoral

DISTRIBUTED MULTI-ROBOT CONTROL: PHYSICS, GEOMETRY AND LEARNING

Autor

Eduardo Sebastián Rodríguez

Director/es

Sagüés Blázquiz, Carlos
Montijano Muñoz, Eduardo

UNIVERSIDAD DE ZARAGOZA
Escuela de Doctorado

Programa de Doctorado en Ingeniería de Sistemas e Informática

2025

Universidad Zaragoza
1542

Tesis Doctoral

Distributed Multi-robot Control: Physics, Geometry and Learning

Autor

Eduardo Sebastián Rodríguez

Director/es

Eduardo Montijano Muñoz
Carlos Sagüés Blázquiz

Escuela de Ingeniería y Arquitectura

2024

Agradecimientos

John Keats, in his gravestone, summarized the spirit of the poets of the Romanticism with the following statement:

"HERE LIES ONE WHOSE NAME WAS WRIT IN WATER"

It is widely acknowledged that the intention behind the statement was for Keats to avoid having his name engraved on his burial site, as a final act of defiance against the concept of individualization. Similarly, this thesis is not the result of a single individual's efforts, but rather the culmination of the collective input of numerous people. Without their invaluable contributions, the journey traversed over the past few years would have been significantly different.

I would like to express my sincerest gratitude to Eduardo and Carlos, my advisors. I am largely indebted to them for my professional development. I first made Eduardo's acquaintance eight years ago and from the outset, it was evident that I wished to collaborate with him and be mentored by him. I was fortunate that my subsequent collaboration with Eduardo resulted in Carlos joining the thesis team. I cannot envisage a more suitable pair of advisors for one's academic and personal development. They have provided comprehensive guidance on all my decisions, including some challenging choices that not all advisors would be willing to support. I am indebted to them for their contribution to this dissertation.

I would also like to express my gratitude to all those who have contributed to the development of this thesis, whether directly or indirectly. After five years of graduate studies, I am still unsure which team, the I3A Team or the Ada Team, is the most welcoming and kind multi-agent system. This is a reflection of the amazing colleagues I have had the pleasure of working with. I would like to extend my gratitude to Diego and Rodrigo for their explicit collaboration on this thesis, and to Sara and Pablo, the parents of the I3A Team, for their implicit contributions. Furthermore, I would like to express my gratitude to Ana Cris, Mauro, Andrea, Rosario and the Luises for allowing me to learn from them. I would also like to acknowledge the contributions of the power electronics lab, from Miguel to Carlos, including Álvaro, Alejandro, Iván, Antonio, Estanis and Fran. I am thankful for the opportunity to participate in electronics projects and for the friendships that have developed outside of the professional collaboration.

I would like to extend my gratitude to the San Diego team for their exceptional support. Embarking on a new continent is never straightforward, and they provided invaluable support, both professionally and personally. I would like to take this opportunity to thank Nikolay, who has provided invaluable guidance and support as my third PhD advisor. I am confident that my current and future achievements are due in large part to his help and advice. The gratitude extends to Thai, who has provided invaluable guidance and support as a mentor and friend. I would further like to extend my gratitude to Adrià, Pol, Nikola, ... Their friendship provided a much-needed sense of companionship. I would be failing in my duty if I did not give a special mention to the members of Platypus in this paragraph: Octavio, Colin, Sebastian, Geelon, Drew, Dustin, Jack, thanks for your contributions to my world view.

The invaluable support of my family and friends deserves special credits, without whom the scientific and engineering principles underpinning this dissertation would not have been

possible. I am proud to say that I am undoubtedly one of the most fortunate individuals in the world. I have consistently received support from my family in the face of any life challenge, specially from my parents and sister. The second category of family, namely those friends that one encounters along the course of life, have also brought considerable happiness, constituting the core of my principles. They have been the perfect feedback loop, stabilizing my system dynamics.

The last words inevitably go to Rebeca, the necessary and sufficient condition of happiness throughout these years. Looking back, you are the person that brings light to the world. Everybody knows how fortunate I am of living along you. During the years encompassing this dissertation we moved in together, we traveled around the globe, and we got engaged. The only thing I can say is: thank you.

Abstract

Multi-robot systems emerge as a promising solution for tackling complex tasks that are beyond the capabilities of a single robot. Their inherent parallelism, robustness to individual failures, and ability to operate in large-scale environments make them particularly appealing for applications such as search and rescue, environmental monitoring, herding and agriculture, or warehouse automation. However, the coordination and control of multiple robots operating in a distributed infrastructure pose significant challenges. Specifically, this thesis aims to address four key aspects of distributed multi-robot systems: (i) effectiveness of coordination in highly nonlinear, volatile environments; (ii) fast and accurate reconstruction of collective information; (iii) scalability of the control policies in the number of robots; and (iv) power supply management. To address the four points, the thesis exploits three main tools: (i) physical properties of networked systems, (ii) geometrical control techniques and (iii) distributed optimization and machine learning methods.

The first part of the thesis addresses the problem of controlling input-nonaffine dynamical systems by means of a team of robots. The main motivation example is that of multi-robot herding, where robots leverage the highly nonlinear input-nonaffine reactive dynamics of the evaders to steer them to desired regions. We develop a novel control technique, called Implicit Control, that is able to stabilize general continuous-time input-nonaffine dynamics and impose desired transient behaviors. We formally characterize the main properties of this new control method and show its beneficial properties for multi-robot herding. The original Implicit Control formulation is suited for centralized systems. Therefore, to enable decentralization, we design novel fast distributed optimization, consensus and dissensus techniques. They achieve, respectively, the fastest existing first-order distributed optimization method, effective robot reconstruction of control quantities of interest, and perception-based effective coordination and collision avoidance.

The second part of the thesis targets distributed stochastic estimation for fast and accurate global awareness acquisition. As departing point, we focus on recovering the optimality properties of the centralized standard Kalman filter in distributed settings, preserving the scalability properties of the existing algorithms. We resolve the problem by proposing the first certifiable optimal distributed Kalman filter under unknown correlations. We find that the consistent fusion of correlated estimates can be cast as a semi-definite program that computes the tightest outer ellipsoid that encloses the intersection of a set of ellipsoids. The local semi-definite program is seamlessly integrated in an event-triggered distributed Kalman filter algorithm to develop the most accurate and communication-efficient distributed stochastic estimator for linear Gaussian systems. The success of the solution motivates us to further work on reconstructing the global solution of the outer Löwner-John ellipsoidal method, a semi-definite program with applications in robust control, computer vision and stochastic estimation. The thesis derives, for the first time, continuous- and discrete-time solutions for static and time-varying cases.

The third part of the thesis copes with the problem of learning distributed multi-robot control policies that are scalable and effective in both cooperative and competitive tasks. We detect that there is a variety of multi-robot tasks that are hard to model

mathematically, motivating the use of learning techniques. We propose a novel physics-informed policy parameterization tailored to simultaneously address scalability, sample efficiency and accuracy in performance. The formulation builds upon port-Hamiltonian systems, a general yet simple description of physical systems to encode the distributed nature of multi-robot systems. We complement the design with a self-attention-based parameterization to handle time-varying topologies. The success of this novel physics-informed policy parameterization is proved to be effective in multi-agent reinforcement learning and imitation learning settings.

The fourth and last part of the thesis revolves around how to transfer some of these multi-robot control techniques to energy systems, an essential ingredient for robot autonomy. The problems examined in this part are considered through the lens of control theory: identification, estimation and control of power devices. We first propose a solution for the problem of automatic thermal model identification of power devices from demonstrations, with widespread applications in optimal thermal design of power devices. Next, we consider the fault-tolerant stochastic estimation and prediction of photovoltaic battery voltage, where Gaussian Process Regression allows effective online prediction of low-energy scenarios in critical environmental infrastructures. The last application consists in the nonlinear implementable control of dual active bridge series resonant converters, a fundamental component in many applications from robotic platforms to energy stations.

Resumen

Los sistemas multi-robot se perfilan como una solución prometedora para abordar tareas complejas, superando las capacidades de un solo robot. Su inherente paralelismo, su robustez ante fallos individuales y su capacidad para operar en entornos a gran escala los hacen especialmente atractivos para aplicaciones como búsqueda y rescate, vigilancia medioambiental, pastoreo y agricultura o automatización de almacenes. Sin embargo, la coordinación y el control de múltiples robots que operan mediante una infraestructura distribuida plantean retos fundamentales. En concreto, esta tesis aborda tres aspectos clave de los sistemas multi-robot distribuidos: (i) la eficacia de la coordinación en entornos altamente no lineales y volátiles; (ii) la reconstrucción rápida y precisa de información global; (iii) la escalabilidad de las políticas de control en el número de robots; y (iv) la gestión de la alimentación y energía. Para abordar los tres puntos, la tesis explota tres herramientas: (i) las propiedades físicas de los sistemas distribuidos, (ii) técnicas de control geométrico y (iii) métodos de optimización distribuida y aprendizaje automático.

La primera parte de la tesis aborda el problema de control de sistemas dinámicos no lineales con respecto a la entrada mediante un equipo de robots. El principal ejemplo motivador es el del pastoreo multi-robot, en el que los robots aprovechan la dinámica reactiva altamente no lineal de los evasores para dirigirlos hacia regiones deseadas. Desarrollamos una novedosa técnica de control, denominada Control Implícito, capaz de estabilizar dinámicas no lineales en entrada y salida en tiempo continuo e imponer comportamientos transitorios deseados. Caracterizamos formalmente las principales propiedades de este nuevo método de control y mostramos sus beneficios para pastoreo multi-robot. La formulación original de Control Implícito es adecuada para sistemas centralizados. Por lo tanto, para lograr su descentralización, diseñamos nuevas técnicas de optimización distribuida, consenso y disenso. Con ellas se consigue, respectivamente, el método de optimización distribuida de primer orden más rápido que existe, la reconstrucción efectiva por parte de los robots de las magnitudes de control de interés, y la coordinación y evitación de colisiones basadas en percepción.

La segunda parte de la tesis se centra en la estimación estocástica distribuida para la adquisición rápida y precisa de contexto global. Como punto de partida, nos centramos en recuperar las propiedades de optimalidad del filtro de Kalman estándar centralizado en entornos distribuidos, preservando las propiedades de escalabilidad de los algoritmos existentes. Resolvemos el problema proponiendo el primer filtro de Kalman distribuido óptimo certificable bajo correlaciones desconocidas. Descubrimos que la fusión consistente de estimaciones correladas puede plantearse como un programa semidefinido que calcula el elipsoide exterior más pequeño que encierra la intersección de un conjunto de elipsoides. El programa local semidefinido se integra de forma transparente en un algoritmo de filtro de Kalman distribuido activado por eventos, desarrollando el estimador estocástico distribuido más rápido y eficiente, desde el punto de vista de la comunicación, para sistemas lineales gaussianos. El éxito de la solución nos motiva a seguir trabajando en la reconstrucción de la solución global del método elipsoidal de Löwner-John, un programa semidefinido con aplicaciones en control robusto, visión por computador y estimación estocástica. La tesis

deriva, por primera vez, soluciones en tiempo continuo y discreto para casos estáticos y variables en el tiempo.

La tercera parte de la tesis aborda el problema del aprendizaje de políticas de control multi-robot distribuidas que sean escalables y efectivas tanto en tareas cooperativas como competitivas. Detectamos que existe una variedad de tareas multi-robot que son difíciles de modelar matemáticamente, lo que motiva el uso de técnicas de aprendizaje. Proponemos una novedosa parametrización de políticas informada por la física y adaptada para abordar simultáneamente la escalabilidad, la eficiencia muestral en entrenamiento y la precisión en el comportamiento. La formulación se basa en los sistemas hamiltonianos, una descripción general pero sencilla de los sistemas físicos para codificar la naturaleza distribuida de los sistemas multi-robot. Complementamos el diseño con una parametrización basada en atención para manejar topologías variables en el tiempo. El éxito de esta novedosa parametrización se demuestra eficaz en entornos de aprendizaje por refuerzo y aprendizaje por imitación.

La cuarta y última parte de la tesis gira en torno a cómo trasladar algunas de estas técnicas de control multi-robot a sistemas energéticos. Los problemas examinados en esta parte se estudian desde la perspectiva de la teoría de control: identificación, estimación y control de dispositivos de potencia. En primer lugar, proponemos una solución para el problema de identificación automática de modelos térmicos de dispositivos de potencia a partir de demostraciones, con amplias aplicaciones en el diseño térmico óptimo de dispositivos de potencia. A continuación, consideramos la estimación y predicción estocástica de la tensión de baterías en instalaciones fotovoltaicas, donde la regresión de procesos gaussianos permite una predicción eficaz en infraestructuras medioambientales críticas. La última aplicación consiste en el control implementable no lineal de convertidores resonantes en serie de doble puente activo, un componente fundamental en muchos dispositivos, que van desde los propios robots autónomos a las estaciones de energía.

Contents

Chapter 1

Introduction

In the contemporary technological landscape, where innovation and automation converge, a compelling paradigm is emerging: the confluence of collective intelligence, networked systems, and robotics. This synthesis foresees a transformative era in decision-making and problem-solving, wherein the aggregate capacity surpasses that of individual constituents. Fundamentally, this phenomenon explores the dynamics of interaction and collaboration among intelligent agents, encompassing both software algorithms and physical robots, within complex networks to achieve common objectives. From the swarm robotics to sensor networks, these systems exemplify the strength of collective intelligence in transcending the limitations of individual entities. Beyond the spectacle, however, lies significant technical challenges comprising sophisticated algorithms, communication protocols, and control mechanisms. This intricate apparatus empowers these systems to perceive their environment, negotiate consensus, and adapt to dynamic circumstances. It is through this synergy of intelligence, connectivity, and autonomy that we unlock the potential to revolutionize societal-scale domains [Christensen et al., 2021, Annaswamy et al., 2023], from search and rescue [Heintzman et al., 2021] to autonomous manufacturing [Walke et al., 2023].

A primary instance of a collective system is that of multi-robot systems. Multi-robot systems (see Fig. 1.1) are a class of complex systems comprising multiple autonomous robots that interact and collaborate to achieve common objectives. These systems leverage the collective capabilities of their constituent robots to accomplish tasks that might be beyond the reach of a single robot, or that can be performed more efficiently or robustly through cooperation. Multi-robot systems present different characteristics depending on the targeted application. In this thesis we consider teams of robots with the following key features:

- **Multiple robots**: The system involves two or more robots, which may be homogeneous (identical) or heterogeneous (with varying capabilities in terms of perception, kinodynamics or goals).

- **Autonomy**: Each robot possesses a degree of autonomy, enabling it to sense its environment, make decisions, and act independently to some extent.

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

Figure 1.1: Applications of multi-robot systems. Teams of robots offer promising capabilities in a broad range of applications, such as (a) agriculture (source: ALLERIN), (b) wildlife monitoring (source: [Shah et al., 2020]), (c) search and rescue (source: [Queralta et al., 2020]), and (d) warehouse operation (source: Amazon), tasks that are commonly challenging for single robot platforms.

- **Cooperation and communication**: Robots can communicate and coordinate their actions to achieve shared goals. This may involve exchanging information about their surroundings, tasks, or internal states. In this sense, the multi-robot infrastructure can be centralized (a server gathers all the information for the robots, processes the information and sends the result back to the robots) or distributed (robots exchange information only with neighboring mates, and all computations are carried on locally at each robot). In this thesis we will focus on distributed infrastructures.

Despite its promising capabilities, effective control of multi-robot systems is an open question, because existing approaches struggle with scalability, resilience and modularity. **The future lies in distributed algorithms** that leverage local multi-robot interactions to achieve global objectives. Sadly, the benefits of distributed systems come at a high cost. The mathematical formulation of task interactions between the robots and the environment, the control of those interactions, and the acquisition of global awareness pose challenges yet to be addressed. **This thesis aims to develop solutions to these challenges by harnessing the power of physics, geometry and learning**, leading to scalable yet accurate distributed multi-robot control policies.

More specifically, **the main contribution of the thesis** is a set of new control techniques to solve distributed multi-robot problems in an effective yet scalable manner. The proposed methods enable cooperative coordination of robotic teams independently on the number of robots, solving problems that encompass nonlinear control in complex settings, stochastic state estimation of dynamic processes, unknown task descriptions and

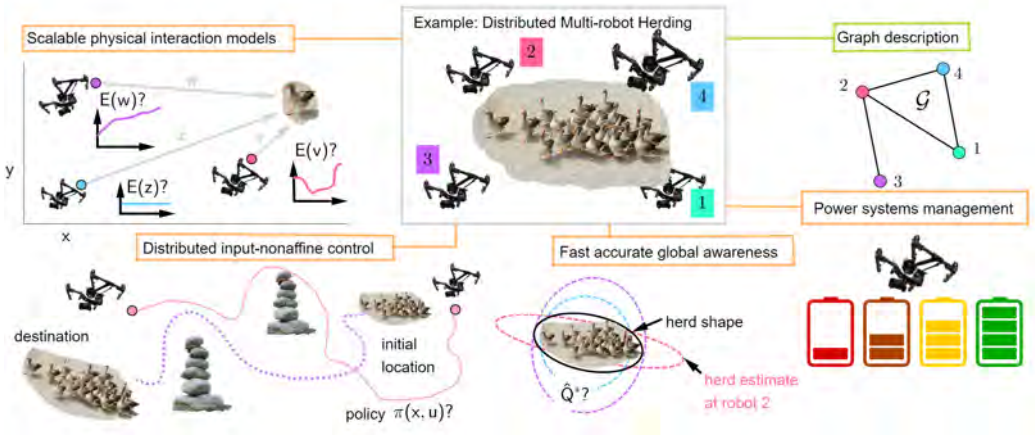


Figure 1.2: Overview of the challenges addressed in the thesis, illustrated by a multi-robot herding problem. Departing from a graph theoretical description of the multi-robot system, we propose to leverage physics, geometry and learning to (i) design distributed input-nonaffine control policies that exploit physical restrictions on the robots and interactions, (ii) obtain global awareness by exploiting geometry in distributed estimation, (iii) model physical interactions between the robots and their environment that scale with the dimensionality of the problem, and (iv) manage the power systems that supply the energy to the multi-robot team.

constraints, and long-term operation through effective power management. The range of applications spans across multiple domains, including cooperative active exploration, multi-robot herding, search and rescue, cooperative transportation or navigation in cluttered environments, always performed by a team of robots in the absence of a central coordination facility. In short, the contributed methods are of general applicability in all multi-robot domains that are restricted to distributed interactions, since they require estimation of observed external processes, demand coordinated control over nonlinear systems, involve unknown phenomena to be learned, and necessitate of a correct power management.

As a guiding example, this introduction will talk about the problem of multi-robot herding (Fig. 1.2). A team of robots is in charge of steering a group of non-cooperative evaders to a desired region. Multi-robot herding is a thrilling problem that highlights the four main challenges encountered in the thesis, leading to the four parts that form the main body of the thesis.

The **first part** of the thesis is focused on **how to address the control of networked systems composed by highly nonlinear complex entities**. In the case of herding problems, the interactions between herders (human, animal, robot) and evaders (e.g., ducks, sheep) involve complex nonlinear physical models that describe how evaders feel attraction to other evaders, whereas they can either feel repulsion or attraction to the herders. Moreover, the local interactions exerted by one of the evaders unchains a sequence of interactions that scale to the whole herd. In general, there exist multiple multi-robot problems that involve nonlinear interactions that, despite having a local origin, affect the whole system. In the absence of a central coordination unit that collects all the information, it is of key importance to develop novel solutions that are able to exploit these complex nonlinear interactions to enforce a desired collective behavior.

Figure 1.3: Multi-robot herding. The goal of multi-robot herding is to help human and animal herders using robotic herders. The robotic herders leverage the nonlinear attractive and repulsive interactions with the evaders (e.g., ducks or sheep) to steer them towards desired regions or motion patterns. Herding encompasses unique challenges from a multi-robot perspective: the interactions herder-evaders are highly nonlinear; the local interactions lead to collective effects that affect the whole herd; the environment is volatile, uncertain and dynamic; and the task is very demanding from a power systems perspective due to the long-term execution time.

The **second part** of the thesis studies **how to exploit collaboration among robots to reconstruct global information of the system**. Particularizing to multi-robot herding, the effective control of the herd requires the herding team to agree on important cues such that the shape and location of the herd, the configuration of the herding team, or what is the prospective direction taken by the evaders. From a control systems perspective, the distributed stochastic estimation of global quantities of interest is a fundamental ingredient in sensor networks, camera tracking systems or multi-robot exploration. The term stochastic refers to the fact that estimates on these quantities usually come from fusing sensor information, known behavior models and other priors, all of which are subject to uncertainty, noise, unexpected disturbances and other error sources. Therefore, one expects the robots to collaborate on reconstructing global information of the system by also accounting for how certain the robot is on the correctness of the estimated value.

The **third part** of the thesis copes with the problem of **how to learn the distributed interactions that emerge in collective tasks**. In herding problems, it is extremely hard to mathematically model all the possible behaviors of the evaders, the particularities of the terrain, and all the environmental conditions that may affect the herd. In contrast, some of these aspects are usually easier to model by means of a reward signal that recompenses the robot team whenever a good action is exerted, or by expert demonstrations from actual herders and evaders. Beyond herding, oftentimes, some of the goals, constraints or features of the task are impossible to describe analytically. This motivates the use of machine learning techniques that learn how to manage the complexities arising in complex tasks and scenarios.

The **fourth part** of the thesis concentrates on **how to exploit all the techniques typically used in multi-robot systems to manage energy system**. Not only in herding, but also in other multi-robot domains, one aspect that is usually overlooked is the power system that thrusts the robots, guaranteeing an energy supply. For instance, in a multi-robot herding problem, the power system is subject to variations in the temperature

Figure 1.4: Distributed Nonlinear Control. From nanobots (left, source: [Soto et al., 2020]) to microgrids (right, source: The Climate Center), there exists an increasing demand on distributed nonlinear control systems. Distributed systems allow to operate in the absence of a central computer, whereas nonlinear control accounts for complex dynamics and interactions.

and radiation conditions of the environment, which affect the power supply system in terms of energy efficiency and state of the batteries. Also, the actions imposed by the control policies of the robots might demand abrupt changes in the energy supply (e.g., the robot needs to rush towards a particular configuration), which must be handled by the power devices of the robot. All these situations require the application of system identification, estimation and control of power devices.

The remainder of this chapter dives into each of the challenges faced at each part of the thesis.

1.1 Distributed nonlinear control

Distributed nonlinear control is a domain of control theory devoted to study techniques to govern complex systems composed of multiple interconnected subsystems, where the dynamics of each subsystem, as well as the interactions between them, may exhibit nonlinear behavior (see Fig. 1.4). In contrast to centralized control schemes, which rely on a single controller to manage the entire system, distributed control decentralizes the control task among multiple local controllers, each responsible for a specific subsystem. In this sense, the local computations carried on at each controller only depends on the information from its adjacent subsystems, such that the information is eventually propagated across the whole network. This approach offers enhanced scalability (the controller is invariant to the size of the network), robustness (each subsystem only handles local information), and adaptability (it is flexible to additions of new subsystems), particularly for large-scale and geographically dispersed systems.

One of the open challenges in this domain is how to handle the interaction between elements of a networked system when they are input-nonaffine, i.e, the system dynamics is not only nonlinear with respect to the state of the system, but also with respect to the action exerted by the controller. In complex settings such as multi-robot herding, system dynamics is given by a general expression of the form $\dot{\mathbf{x}} = \mathbf{f}_\theta(\mathbf{x}, \mathbf{u})$. Function $\mathbf{f}_\theta(\bullet)$ describes input-nonaffine dynamics that depends on the configuration of all the entities $\mathbf{x}$, the robot

actions $\mathbf{u}$, and a collection of parameters θ that might be learned from data. Standard nonlinear control techniques like feedback linearization [Westenbroek et al., 2020] fail because they assume dynamics $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}$. Other solutions [Pierson and Schwager, 2017, Lin et al., 2020] propose control policies that only fit a certain election of the input-nonaffine dynamics. **This thesis proposes novel (distributed) nonlinear control techniques that handle general input-nonaffine dynamics by exploiting numerical methods popular in the applied physics domain, combined with novel distributed optimization, consensus and dissensus methods.**

To tackle general input-nonaffine dynamics, we formalize the first nonlinear control technique that stabilizes general input-nonaffine dynamics, called Implicit Control [Sebastián and Montijano, 2021]. The solution analyzes the flow of the Newton-Raphson method over $\mathcal{C}^1$ functions $\mathbf{f}$ to obtain an explicit expression and design rules for the dynamics of the input $\dot{\mathbf{u}}$ such that convergence to a desired behavior $\mathbf{f}^* \in \mathcal{C}^1$ is guaranteed. Implicit Control is successfully applied to the adaptive multi-robot herding problem [Sebastián et al., 2022a], validated in different herding dynamics and herd scales [Sebastián et al., 2022b]. The thesis also presents novel accelerated dynamic consensus algorithms [Sebastián et al., 2023c], distributed optimization methods [Sebastián et al., 2024] and dissensus methods based on opinion dynamics [Martinez-Baselga et al., 2024] to enable the extension to distributed settings.

1.2 Fast accurate distributed estimation

For collective coordination under control policies, robots need global awareness of the task in the form of approximations to the robots' local and global team configurations [Lasserre, 2015, Saravanos et al., 2023], or the goals to achieve [Deits and Tedrake, 2015]. Robots estimate and represent global quantities by exploiting perception cues or communication. Stochastic distributed estimation emerges as a promising solution, enabling each robot to leverage its own local observations, while collaborating with others to construct a globally consistent "picture" of the system. This approach becomes particularly crucial in scenarios where communication constraints or computational limitations hinder the feasibility of centralized estimation schemes (see Fig. 1.5). Various techniques, such as consensus-based algorithms or distributed Kalman filtering, can be employed to facilitate this process [Olfati-Saber et al., 2007, Talebi and Werner, 2019]. These algorithms exploit the inherent connectivity of the multi-robot network, allowing robots to exchange information with their neighbors and iteratively refine their individual estimates. The convergence properties of these algorithms ensure that, under suitable conditions, the collective estimate approaches the true system state, even in the presence of stochastic disturbances or communication delays.

One significant challenge is how to equip robots with methods that allow them to reconstruct the state of the system and build high-level representations (e.g., the shape of the herd) that accommodate their control objectives. Methods for acquiring such global awareness must be fast and accurate since they are intertwined with the control policies.

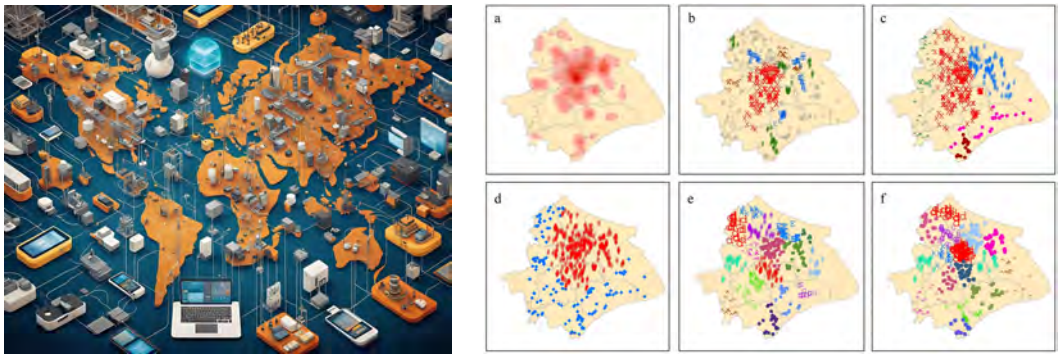


Figure 1.5: Fast Accurate Distributed Estimation. Multi-robot coordination in distributed settings requires the reconstruction of global quantities of interest, where ellipsoidal descriptions play a fundamental role. The left pannel represents an IoT network, where each node acquires local information that shares with neighboring nodes. The ellipsoidal description of the information is widespread across domains, such as in industrial clustering, depicted in the right pannel (source: [Zhao et al., 2023]).

One fundamental representation related to robotics is ellipsoids, positive definite symmetric matrices that are present everywhere from the local robot level (e.g., uncertainty of state estimates) to the system level (e.g., the herd's shape). Existing distributed approaches [Olfati-Saber, 2007a, Talebi and Werner, 2019] rely on the exchange of estimates with the neighboring robots. This exchange leads to cross-correlation of the estimates of one robot with those from all the other robots. However, since robots cannot track all the cross-correlations of the network for scalability, existing approaches assume that cross-correlations are zero, leading to protocols that trade accuracy, scalability and speed.

This thesis develops distributed ellipsoidal methods that are optimal in terms of accuracy and communication efficiency for distributed stochastic estimation settings. We find that the distributed estimation problem can be reformulated as finding the smallest outer ellipsoid that approximates the intersection of a set of ellipsoids spanned across the nodes of a network. This leads to an optimization problem, called the outer Löwner-John method [Henrion et al., 2001], that we have exploited to develop two classes of algorithms. First, a novel certifiable distributed Kalman filter that recovers the optimality guarantees of the centralized setting and has linear scalability with the number of nodes [Sebastián et al., 2021b], along with its event-triggered version [Sebastián et al., 2023a]. Second, a novel distributed version of the outer Löwner-John method [Aldana-López et al., 2023, Sebastián et al., 2023b] that robots can use to reconstruct the global optimal outer ellipsoid by exchanging their local ellipsoidal representations, reconstructing, e.g., the shape of a herd in a distributed manner.

1.3 Learning distributed collective interactions

Learning distributed collective interactions involves understanding how individual agents within a system interact to produce complex group behaviors. These interactions can be found in various domains, such as flocks of birds, ant colonies, and even human social networks [Nolfi, 2002, Strömbom et al., 2014a, Mavrogiannis et al., 2023] (see Fig. 1.6).

Figure 1.6: Learning Distributed Collective Interactions. The local interactions among agents in a network lead to global emergent behaviors. For instance, on the left pannel, birds in a flock only rely on their sensors (eyes) to account for collision avoidance and velocity alignment, such that the flock navigates in the sky. This idea also applies to competitive scenarios, like in StarCraft II (pannel on the right), where units arrange in groups with different goals such that the whole team conquers a territory.

For instance, in a flock of birds, each bird might follow simple rules, such as maintaining a certain distance from its neighbors and aligning its direction with the flock. These local interactions can then lead to the emergence of beautiful and coordinated flocking patterns, such as V-formations or swirling movements. Understanding these principles can help us design better algorithms for swarm robotics and distributed control systems, where many simple agents need to cooperate to achieve complex tasks.

Unfortunately, it is often the case that the complexity of such interactions impede the derivation of an analytical description of such behaviors. Instead, it is possible to provide a high-level description of the behavior that expresses the main goals and restrictions. One option to this description is the use of reward functions, typical in reinforcement learning settings [Sutton, 2018], where the reward signal is just a function that incentives the fulfilment of task objectives (e.g., reaching a target) and penalizes forbidden behaviors (e.g., colliding with another robots). Another alternative is to leverage demonstrations provided by experts (e.g., videos of bird flocks flying) to learn to mimic such behavior through imitation learning [Ho and Ermon, 2016]. In all cases, data is required to capture the underlying behaviors for a successful control. Meanwhile, models must respect the physical and dynamical properties of the environment (cf. evader-herder interactions) and the robots (e.g., the kinodynamic hardware constraints). In recent years, the area of physics-informed neural networks [Cuomo et al., 2022, Nghiem et al., 2023] has emerged as an excellent alternative to impose physical priors. Driven by Neural Ordinary Differential Equations [Chen et al., 2018], we can learn neural-network-based models of the form $\dot{\mathbf{x}} = \mathbf{f}_\theta(\mathbf{x}, \mathbf{u})$, bridging the gap between learned models and control. However, can we impose scalability priors? Therefore, **this thesis pursues data-driven methods than learn distributed interactions that scale with the dimension of the system, and respect its network and physical constraints.**

We bring together the benefits of physics-informed neural networks to the problem of learning local interactions by exploiting port-Hamiltonian mechanics and self-attention networks, embedded in a Neural Ordinary Differential Equations model [Sebastián et al., 2023b]. Port-Hamiltonian dynamics [Van Der Schaft et al., 2014] describes a system in terms of input-output energy ports. This compositional property [Furieri et al., 2022a]

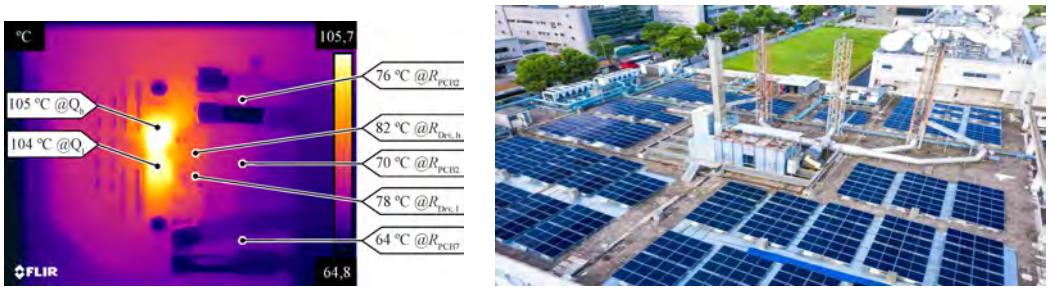


Figure 1.7: Management of power systems. Power systems are fundamental for the operation and energy supply of networked systems. The correct operation of power devices is subject to physical restrictions and environmental conditions. For instance, on the left, a infrared frame from a thermal camera highlights the significant differences in temperature of the different semiconductor devices within a few centimeters. On the right, a photovoltaic standalone that supplies the batteries of a factory, where the radiation and temperature impact the sustainability of the station (source: ARKA 306).

naturally imposes the network scalability constraints while the energy conservation laws of the robots are respected. The self-attention networks [Vaswani et al., 2017] parameterize the port-Hamiltonian model because they can handle the time-varying information available at each robot from communication or perception. This approach exhibits promising results in multi-robot learning-from-demonstrations [Sebastián et al., 2023b, Roche et al., 2025], multi-robot reinforcement learning [Sebastian et al., 2023], dynamic multi-robot games [Sebastián et al., 2025] and topology learning from multi-robot trajectories [Sebastián et al., 2023a]. These works stand out in the ability to equal or surpass state-of-the-art centralized control policies in performance while generalizing to unseen sizes without increasing the computation and communication cost.

1.4 Management of power systems

Modern robots are complex systems demanding precise and efficient power management for optimal performance and safety. Control techniques play a fundamental role in regulating the power supply within these systems, ensuring reliable operation and extending operational life. From industrial manipulators to mobile service robots, sophisticated control algorithms address challenges such as varying workloads, energy efficiency, thermal management, and fault tolerance (see Fig. 1.7). By intelligently adapting to dynamic conditions and optimizing power utilization, these techniques enhance the capabilities and robustness of robotic systems across diverse domains.

The effective supply of energy in robotic systems is subject to the correct thermal design [Wattenberg et al., 2023], fault-tolerant estimation [Sechilariu et al., 2012] and control [Tavakoli and Pantic, 2018] of power devices. In regard of thermal design, the power-temperature dynamics of power converters are critical to characterize the behavior of the energy supply to the robot under non-standard weather and environmental conditions [Chang et al., 2023], which is not trivial due to the advances in semiconductors and substrate technologies [Buffolo et al., 2024, Guacci et al., 2020]. Typical solutions rely on model-based approaches [Azurza Anderson et al., 2021] or finite-element analysis [Van Der

Broeck et al., 2018], which overlook the unmodeled effects that appear in the real devices once fabricated. The limitations associated to the external conditions not only affect the hardware, but also the source of energy supply itself [Vazquez et al., 2010], as with photovoltaic supply, where a medium- or long-term lack of radiation can lead to a deterioration of the batteries and the inoperativeness of the robot. Existing solutions either obtain temperature-voltage models from simulated data [Liu et al., 2019a, Liu et al., 2020, Liu et al., 2019b] or classical model-based estimators [El Mejdoubi et al., 2015] that do not consider complex meteorological phenomena captured in real data. In addition, although the power devices and battery system were correctly supplied, then there is the question on how to control the power supply to the other components of the robots. In particular, bi-directional DC/DC resonant converters enable fast power conversion ratios [Zhao et al., 2017], but their model complexities hindered the design of advanced control techniques that fully exploit the capabilities of the power converter [Krismer and Kolar, 2009, Li and Bhat, 2009]. Henceforth, **this thesis solves problems on identification, estimation and control of power devices, enabling optimal power converter design, fault-tolerant prediction, and fast and stable energy supply of autonomous systems.**

First, we develop an automatic data-driven method that is able to learn the optimal power-temperature linear discrete-time dynamics of any power converter from just temperature-power profiles [Sanz-Alcaine et al., 2024]. The method enables the addition of other prior information to constrain the learning, surpassing existing model-based methods in accuracy and flexibility. Second, we study Gaussian Process Regression to learn to predict the current/voltage battery profiles of autonomous systems supplied by photovoltaic devices. We find that the use of inducing points enables fast yet accurate long-term predictions from temperature information, a fundamental feature to predict potential failures in the energy supply. Third, we develop a novel nonlinear implementable control of a dual active bridge series resonant converter [Sebastián et al., 2021a]. Departing from a Generalized Averaging Method model [Sanders et al., 1991, Mahdavi et al., 1997], we design a control policy for switching frequency and phase shift of the converter, both of them affecting in a nonlinear manner to the dynamics of the plant, which makes it particularly suitable for the systems addressed in the thesis. The nonlinear technique, based on control Lyapunov functions and hybrid control, is simple enough to be implementable in a microcontroller, yet proved to be globally asymptotically stable with fast transient.

1.5 Summary of results

The outcomes of all the work developed during the thesis have resulted in 11 journal papers, 8 international conference papers and 2 workshop papers (* means equal contribution):

Journals

1. D. Martinez-Baselga*, E. Sebastián*, E. Montijano, L. Riazuelo, C. Sagüés, L. Montano. AVOCADO: Adaptive Optimal Collision Avoidance driven by Opinion. Submitted to IEEE Transactions on Robotics. 2024 Quartile: Q1. Impact Factor:

9.4. [Martinez-Baselga et al., 2024]

2. E. Sebastián, T. Duong, N. Atanasov, E. Montijano, C. Sagüés . Physics-Informed Multi-Agent Reinforcement Learning for Distributed Multi-Robot Problems. Submitted to IEEE Transactions on Robotics. 2024 Quartile: Q1. Impact Factor: 9.4. [Sebastian et al., 2023]

3. E. Sebastián*, R. Aldana-López*, R. Aragüés , E. Montijano, C. Sagüés. Distributed Discrete-time Dynamic Outer Approximation of the Intersection of Ellipsoids. Submitted to IEEE Transactions on Automatic Control. 2024 Quartile: Q1. Impact Factor: 6.2. [Sebastián et al., 2023b]

4. J. M. Sanz-Alcaine, E. Sebastián, F. J. Perez-Cebolla, A. Arruti, C. Bernal-Ruiz, I. Aizpuru. Estimation of Semiconductor Power Losses Through Automatic Thermal Modeling. IEEE Transactions on Power Electronics. 2024 Quartile: Q1. Impact Factor: 6.6. [Sanz-Alcaine et al., 2024]

5. E. Sebastián, M. Franceschelli, A. Gasparri, E. Montijano, C. Sagüés. Accelerated Alternating Direction Method of Multipliers Gradient Tracking for Distributed Optimization. IEEE Control Systems Letters. 2024 Quartile: Q2. Impact Factor: 2.4. [Sebastián et al., 2024]

6. E. Sebastián, E. Montijano, C. Sagüés. ECO-DKF: Event-Triggered and Certifiable Optimal Distributed Kalman Filter. IEEE Transactions on Automatic Control. 2024 Quartile: Q1. Impact Factor: 6.2. [Sebastián et al., 2023a]

7. E. Sebastián, E. Montijano, C. Sagüés, M. Franceschelli, A. Gasparri. Accelerated Multi-stage Discrete Time Dynamic Average Consensus. IEEE Control Systems Letters. 2023 Quartile: Q2. Impact Factor: 2.4. [Sebastián et al., 2023c]

8. R. Aldana-López, E. Sebastián, R. Aragüés , E. Montijano, C. Sagüés. Distributed Outer Approximation of the Intersection of Ellipsoids. IEEE Control Systems Letters. 2023 Quartile: Q2. Impact Factor: 2.4. [Aldana-López et al., 2023]

9. E. Sebastián, E. Montijano, C. Sagüés. Adaptive Multi-robot Implicit Control of Heterogeneous Herds. IEEE Transactions on Robotics. 2022 Quartile: Q1. Impact Factor: 7.8. [Sebastián et al., 2022a]

10. J. M. Sanz-Alcaine, E. Sebastián, I. Sanz-Gorrachategui, C. Bernal-Ruiz, A. Bono-Nuez, M. Pajovic, V. Orlik. Online Voltage Prediction using Gaussian Process Regression for Fault-Tolerant Photovoltaic Standalone Applications. Neural Computing and Applications. 2021 Quartile: Q2. Impact Factor: 5.1. [Sanz-Alcaine et al., 2021]

11. E. Sebastián, E. Montijano, E. Oyarbide, C. Bernal, R. Gálvez. Nonlinear Implementable Control of a Dual Active Bridge Series Resonant Converter. IEEE Transactions on Industrial Electronics. 2021 Quartile: Q1. Impact Factor: 8.2. [Sebastián et al., 2021a]

International Conferences

1. W. Zhan*, C. Dong*, E. Sebastián, N. Atanasov. Image-to-Automata Learning for Robot Task Planning. In preparation. 2025. CORE: TBR. [Zhan et al., 2025]

2. E. Sebastián, N. Atanasov, E. Montijano, C. Sagüés. Policy Gradient learning of Distributed Regulators for Nonlinear Multi-Agent Dynamic Games. In preparation. 2025. CORE: TBR. [Sebastián et al., 2025]

3. J. Roche, E. Sebastián, E. Montijano. Curriculum Imitation Learning for Distributed Multi-robot Policies. submitted to IEEE International Conference on Robotics and Automation (ICRA). 2025. CORE: A+. [Roche et al., 2025]

4. E. Sebastián, T. Duong, N. Atanasov, E. Montijano, C. Sagüés. Learning to Identify Graphs from Node Trajectories in Multi-robot Networks. IEEE International Symposium on Multi-robot & Multi-agent Systems (MRS). 2023. CORE: TBR. [Sebastián et al., 2023a]

5. E. Sebastián, T. Duong, N. Atanasov, E. Montijano, C. Sagüés. LEMURS: Learning Distributed Multi-Robot Interactions. IEEE International Conference on Robotics and Automation (ICRA). 2023. CORE: A+. [Sebastián et al., 2023b]

6. E. Sebastián, E. Montijano, C. Sagüés . Multi-robot Implicit Control of Massive Herds. Iberian Robotics Conference (ROBOT). 2022. CORE: TBR. [Sebastián et al., 2022b]

7. E. Sebastián, E. Montijano, C. Sagüés . All-in-one: Certifiable Optimal Distributed Kalman Filter under Unknown Correlations. IEEE Conference on Decision and Control (CDC). 2021. CORE: A. [Sebastián et al., 2021b]

8. E. Sebastián, E. Montijano. Multi-robot Implicit Control of Herds. IEEE International Conference on Robotics and Automation (ICRA). 2021. CORE: B. [Sebastián and Montijano, 2021]

Workshops:

1. E. Sebastián, E. Montijano, C. Sagüés . On the Distributed Multi-robot Herding. IEEE ICRA 2023 Workshop on Distributed Graph Algorithms for Robotics. 2023.

2. E. Sebastián, E. Montijano, C. Sagüés . Adaptive Multi-robot Implicit Control of Herds. RSS 2021 Workshop on Behavioral Inference of Remotely Sensed Multi-Agent Systems. 2021.

These publication results involve international collaborations with the University of California San Diego, the University of Cagliari, the University of Roma Tre, Mitsubishi Electric Research Laboratory, Mondragon Unibertsitatea, and Epic Power. As part of dissemination activities, I have given talks at the Sibley School of Mechanical and Aerospace Engineering (Cornell University, October 2024), the Collaborative Autonomy Group (Boston

University, December 2023), the ProrokLab (University of Cambridge, March 2023 and October 2024) and Infineon Technologies (Austria, October 2022).

The research associated to this thesis has been supported by the following agencies/projects:

- Spanish grants FPU19-05700 and EST22/00253.
- Fulbright grant for predoctoral visits 2023-2024.
- ONRG-NICOP-grant N62909-19-1-2027 and N62909-24-1-2081.
- NSF CCF-2112665 (TILOS).
- Spanish projects PID2021-125514NB-I00, PID2021-124137OBI00 and TED2021-130224B-I00 funded by MCIN/AEI/10.13039/501100011033, by ERDF A way of making Europe and by the European Union NextGenerationEU/PRT.
- Spanish Project PGC2018-098719-B-I00 and Spanish Project PGC2018-098817-A-I00 (MCIU/AEI/FEDER, UE).
- Gobierno de Aragón projects DGA T45-17R, DGA T45-20R and DGA T45-23R.
- Sociedad Ibérica de Construcciones Eléctricas S.A.
- Confederación Hidrográfica del Ebro.

During this thesis, I also had the chance to undertake **two research visits at the University of California San Diego**. I visited the Existential Robotics Laboratory (ERL) under the supervision of Prof. Nikolay Atanasov from April to July of 2022, funded by the EST22/00253 Spanish grant and resulting in the following publications: [Sebastián et al., 2023b, Sebastián et al., 2023a]. The second visit was also to the Existential Robotics Laboratory (ERL) under the supervision of Prof. Nikolay Atanasov, from May to November of 2024, funded by a Fulbright grant. The visit resulted in the following publications: [Sebastian et al., 2023, Zhan et al., 2025, Sebastián et al., 2025].

During the thesis, I supervised the **Bachelor Thesis** *Machine learning for emergent behaviours in multi-robot systems* of Jesus Roche Soguero, student at the Universidad de Zaragoza in Computer Science Engineering. Besides, I have collaborated as **Teaching Assistant** in *Control Engineering* courses at the B.Eng. in Industrial Technologies Engineering (2024) and B.Eng. in Electronic and Automatic Engineering (2021 and 2022), both at the Universidad de Zaragoza. I have also collaborated as **Teaching Assistant** in *Electronic Design and Advanced Control* courses at the M.Eng. in Industrial Engineering (2021, 2022 and 2024) of the Universidad de Zaragoza. Finally, I have collaborated as **Teaching Assistant** in the *Distributed Estimation using Consensus methods* course at the DISC Summer School in Multi-Robot Systems at TU Delft (2021).

Along with the presented publications, I have participated in the reviewing process of 30 papers from **different journals and international conferences**: IEEE Transactions on Robotics (3), IEEE Robotics and Automation Letters (4), IEEE Transactions on Automatic Control (1), Automatica (1), IEEE Transactions on Signal Processing (1), Robotics:

Science and Systems (1), IEEE Control Systems Letters (1), Autonomous Robots (2), IEEE Transactions on Automation Science and Engineering (1), IEEE International Conference on Robotics and Automation (5), IEEE European Control Conference (1), Iberian Robotics Conference (1), IEEE International Symposium on Multi-Robot and Multi-Agent Systems (2), IEEE/RSJ IEEE International Conference on Intelligent Robots and Systems (2), IFAC World Congress (1), American Control Conference (2), International Conference on System Theory, Control and Computing (1).

Importantly, standing on the shoulders of giants, the results of this thesis derive from the efforts and brilliant ideas from many other researchers from the present and the past. Among the thousands of pages read in these 5 years, I have studied in-depth around 1600 papers. From these articles, around 500 papers have directly influenced the works that encompass this thesis document.

Finally, in these five years I have participated in different representative and volunteering initiatives. I have been the representative of the graduate students of the Engineering and Architecture School in the Graduate Commission of the Universidad de Zaragoza, from 2022 to 2025. I have been an organizational member of PIF (union in the defense of the rights of graduate students) from 2020 to 2024, being part of the movement that achieved labor rights for the Spanish graduate students, e.g., end-of-contract compensations. I participated as a volunteer in the 5th Iberian Robotics Conference held in Zaragoza in 2022.

1.6 Manuscript organization

The thesis is divided in **four parts**:

- The **first part** deals with the problem of controlling input-nonaffine systems by means of distributed control techniques. Specifically, Chapter 2 presents Implicit Control, a novel control technique for general input-nonaffine dynamic systems, and presents solutions for multi-robot herding applications. Chapter 3 proposes distributed optimization, consensus and dissensus methods to enable distributed multi-robot control.
- The **second part** of the thesis describes novel ellipsoidal methods for distributed stochastic estimation. Chapter 4 presents the first event-triggered certifiable optimal distributed Kalman filter, the optimal and communication-efficient stochastic estimator for distributed linear Gaussian systems. Chapter 5 extends the ellipsoidal notions of Chapter 4 to general outer approximations of the intersection of ellipsoids, with relevant applications in robust control, computer vision and estimation.
- The **third part** of the thesis develops novel methods for the learning of distributed multi-robot policies. Chapter 6 presents a novel physics-informed policy parameterization that achieves scalability in the number of robots and sample efficiency in training in both multi-agent reinforcement learning and imitation learning settings.
- The **fourth part** of the thesis studies how to transfer the control theoretical notions developed in the previous chapters to power systems. This is done in Chapter 7, where

the thesis solves problems of thermal-power system identification, state estimation in fault-tolerant standalone installations and nonlinear control of power converters.

Each of these chapters contains a dedicated section discussing related work and conclusions. Besides, due to the diversity of topics covered in the thesis, each chapter uses its own symbols and notation.

Finally, Chapter 8 reflects on the main contributions of the thesis, discusses potential opportunities of research and introduces future research directions related to the effective, scalable control of distributed multi-robot systems.

Part I

Distributed Input-Nonaffine Control

Chapter 2

Implicit Control for Multi-robot Herding

Recent advances in multi-robot systems have favored the development of successful control strategies in real-life problems such as entrapment [Antonelli et al., 2008], hunting [Zhu et al., 2015] or escorting [Gao et al., 2018]. Despite the different nature of scenarios, these problems can all be seen as instances of herding [Pierson and Schwager, 2018], where the objective is to drive a group of targets or evaders to specific locations using a team of robots or herders. Common to all of them is the non-cooperative nature of the evaders with respect to the control objective, typically entangled in complex nonlinear and heterogeneous behaviors. In fact, the difficulties hidden in the herding problem have motivated broad interdisciplinary research assembling physiologists, mathematicians and neurologists with engineers [Nolfi, 2002, Strömbom et al., 2014b, Long et al., 2020].

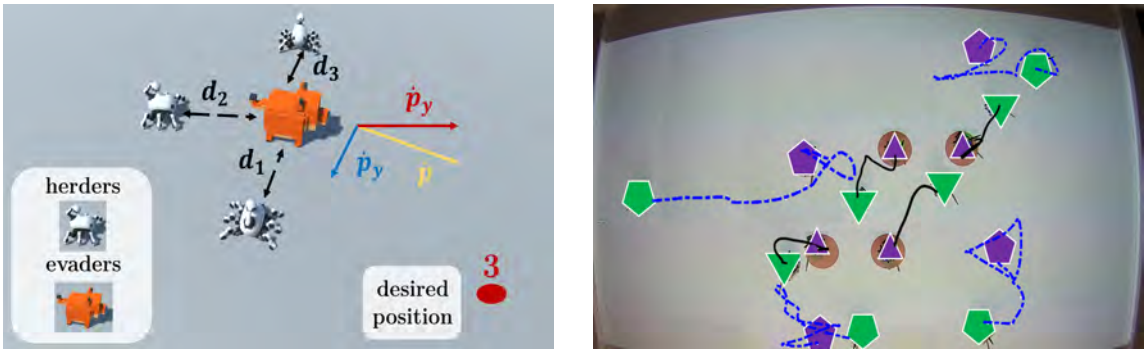


Figure 2.1: In herding, the robotic herders drive the evaders towards specific locations simultaneously. The control strategy exploits repulsive forces (left) to place the evaders in their desired positions. The experiment in the right depicts four robots herding four evaders. The symbols are explained in Table 2.1.

To cope with these difficulties, this chapter presents a novel control solution, called Implicit Control, that is capable of steering a group of evaders towards individually assigned goals simultaneously, irrespective of their heterogeneous and non-cooperative nonlinear dynamics. The proposal leverages numerical analysis theory to derive proofs of existence and stability of the control. Furthermore, these fundamentals are used to develop a general

adaptation law that tackles the herding in the presence of uncertainties in the evaders' models. With the inclusion of a distributed estimator and a caging strategy we present a complete, flexible, general herding solution. Finally, we present a new dynamic assignment strategy that chooses the worthiest evaders to focus in terms of herd coalition. The flexibility and generality of Implicit Control also allows to steer large herds to a common region or to control different herds individually. Overall, the multi-robot Implicit Control of herds stands as a representative solution for the control of input-nonaffine systems. The contributions of this chapter have been published in [Sebastián and Montijano, 2021, Sebastián et al., 2022a, Sebastián et al., 2022b].

2.1 Related work

Most of the existing works deal with the herding of a single entity. An early example is [Antonelli et al., 2008], which employs a Null space-based behavioral control to escort/entrap the target. The authors of [Zhu et al., 2015] apply a Bio-inspired Neural Network to hunt in an underwater environment. To produce an escorting behavior, [Du et al., 2017] develops a distributed switching strategy, where the escorting task is assigned from one herder to another when a Voronoi boundary is crossed. Few works solve the herding of more than one evader. In [Jahn et al., 2017], a group of robots navigates around a certain area to prevent evaders to cross its boundaries. The work in [Pierson and Schwager, 2018] drives groups of entities by an active encirclement but does not consider specific final positions for each evader. Following a similar approach, the authors in [Chipade and Panagou, 2019] go a step further and propose an active encirclement which avoids obstacles. With the same spirit, [Stilwell et al., 2005] and [Antonelli and Chiaverini, 2006] develop solutions to drive a whole herd towards a certain region, where it is required that some global features of the herd (e.g., mean position) converge to a desired equilibrium. Recently, [Auletta et al., 2022] exploits simple local rules to push all herders towards the same containment area. The authors of [Song et al., 2021] present an approach based on algebraic topology called "herding by caging", where repulsive forces are leveraged to steer the evaders, again, towards a certain general region. In contrast to all these works, we seek a method for the herding of all the evaders to precise individual locations simultaneously, such that the method can also be used to steer large herds towards a global region or to split and steer each sub-herd towards individually assigned regions simultaneously.

Another feature of herding problems is the behavior of the evaders, where two assumptions are often considered: linear and homogeneous dynamics. The first assumption is considered in, e.g., [Jiang et al., 2018], where the containment of linear heterogeneous agents is performed by a time-varying formation. The second assumption is used in [Pierson and Schwager, 2018], where agents with nonlinear homogeneous repulsive dynamics are controlled using a team of robots. A different instance is [Alexopoulos et al., 2017], where a complete control structure is presented, from the hex-rotor motion to the pursuit layer. In this chapter we pursue a method that is able to handle general nonlinear and heterogeneous evaders.

Most of the aforementioned solutions assume perfect knowledge of the herd dynamics. To deal with uncertainty, a common approach is the design of model-free strategies. The work in [Franchi et al., 2010] localizes the target with relative-position sensors and then applies probabilistic tools for the encirclement. Similarly, [Zhu et al., 2018] includes information from a camera to learn the behavior of the target. These instances deal with the herding of a single target. An extension to n robots detecting n targets can be found in [Desouky and Schwartz, 2010], using a mixture of learning, optimization and fuzzy techniques. The use of Gene Regulatory Networks is studied in [Peng et al., 2016] to entrap groups of targets. In both cases, no accuracy in the final position of the evaders is required, and in [Peng et al., 2016] the targets do not react against the robots' efforts. This lack of accuracy and flexibility in the number of evaders under uncertainty is another challenge addressed in this chapter.

2.2 Problem statement

We consider the problem of herding a group of m evaders using a team of n robotic herders. We denote the evaders by $j \in \{1, ..., m\}$, and the herders by $i \in \{1, ..., n\}$. Since we aim at using the position of the herders to control the position of the evaders, we define the state $\mathbf{x} \in X \subseteq \mathbb{R}^{2m}$ and the input $\mathbf{u} \in U \subseteq \mathbb{R}^{2n}$ as $\mathbf{x} = [\mathbf{x}_1^\top, \ldots, \mathbf{x}_m^\top]^\top$ and $\mathbf{u} = [\mathbf{u}_1^\top, \ldots, \mathbf{u}_n^\top]^\top$, where $\mathbf{x}_j = [x_j^e, y_j^e] \in X_j \subseteq \mathbb{R}^2$ is the position of evader j and $\mathbf{u}_i = [x_i^h, y_i^h] \in U_i \subseteq \mathbb{R}^2$ is the position of herder i, operating in a 2D space[1]. The movement of each evader is described by general dynamics

$$\dot{\mathbf{x}}_j = f_j(\mathbf{x}, \mathbf{u}), \tag{2.1}$$

allowing for any nonlinear behavior encoded in $f_j(\mathbf{x}, \mathbf{u})$. Notice that we account for the possibility of nonlinear dependencies on the input. The only assumption regarding f_j is that it is of class C^1 for all j. We exemplify Eq. (2.1) using two dynamic models from the literature. Their choice is motivated by the strongly nonlinear behavior in the position of evaders and herders, described by input-nonaffine dynamics.

The *Inverse Model* (adapted from [Pierson and Schwager, 2018]) is

$$\dot{\mathbf{x}}_j = f_j^{inv}(\mathbf{x}, \mathbf{u}) = \theta_j \sum_{i=1}^{n} \frac{\mathbf{d}_{ij}}{||\mathbf{d}_{ij}||^3}, \tag{2.2}$$

where $\mathbf{d}_{ij} = \mathbf{x}_j - \mathbf{u}_i$ is the relative position between evader j and herder i, and θ_j is a positive constant which expresses the aggressiveness in the repulsion provoked by the herders. Note that despite the model has a singularity in $\mathbf{d}_{ij} = \mathbf{0}$, in practice the speed remains bounded. The repulsion grows with $\frac{1}{||\mathbf{d}_{ij}||^3}$ so the closer the herders, the larger the repulsion. The only way of achieving $\mathbf{d}_{ij} = \mathbf{0}$ is that a perfectly evenly distributed number of herders approaches the evader. This does not happen in practice because herders need to impose non-zero repulsive forces to steer the evaders.

[1]The choice of a 2D space is to fit the real experiments, but the solution can be generalized to higher dimensions without changes in the formulation.

The *Exponential Model* (adapted from [Licitra et al., 2018]) is

$$\dot{\mathbf{x}}_j = f_j^{exp}(\mathbf{x}, \mathbf{u}) = \theta_j \sum_{i=1}^{n} \mathbf{d}_{ij} e^{-\chi_{ij}} (1 - \beta_j \text{sigm}(-||\mathbf{d}_{ij}|| + d_{\min})), \tag{2.3}$$

where $\chi_{ij} = \frac{1}{\sigma_j^2} \mathbf{d}_{ij}^\top \mathbf{d}_{ij}$ and $\sigma_j > 1$. In this model there is a switching condition when $||\mathbf{d}_{ij}|| \leq d_{\min}$, where the evader j becomes "scared" when the distance with some herder i is smaller than $d_{\min}$ and the intensity of the repulsive interaction increases, due to $0 < \beta_j < 1$. In practice, the sigmoid function sigm($\bullet$) is included in (2.3) to avoid the violation of the C^1 assumption. In a similar fashion, a sigmoid function or an hyperbolic tangent can be used to model saturations in the speed of the evaders.

Given the individual dynamics of the evaders in Eq. (2.1), the joint system dynamics can be defined as

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}), \tag{2.4}$$

where $f(\mathbf{x}, \mathbf{u})$ simply comes from stacking all $f_j(\mathbf{x}, \mathbf{u})$. This formulation allows to consider heterogeneous herds, with different number of evaders and motion models.

Our goal is to herd the evaders towards the desired positions $\mathbf{x}^* \in X$ simultaneously. To do this, we define the position error of the evaders as $\widetilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}^*$, and we set the control objective to be to drive $\widetilde{\mathbf{x}}$ to zero. This is what we call precise herding. A particular instance of precise herding is the herding of a herd towards the same region, where some of the elements of $\mathbf{x}^*$ are specified by the practitioner and the rest of them are free or directly dependent on the others. Nevertheless, in general, we consider the case where each evader $\mathbf{x}_j$ has an individually assigned $\mathbf{x}_j^*$.

It is noteworthy that the reactive behavior of the evaders is with respect to the position of the herders. Therefore, a control strategy which determines inputs in terms of herders' positions is adequate to generalize the solution to different robotic platforms. This high-level approach can be combined with any robot-specific low-level controller.

To keep the generality of the solution, in this work we assume that the maximum velocity of both herders and evaders is $v_{\max}$. This requires an initial caging phase, surrounding the evaders to avoid their escape before the precision herding begins. It is not mandatory to achieve a compact and closed encirclement to succeed but just the distribution of the herders near the evaders. This is the reason of developing a caging stage to complete the herding, described in Subsection 2.5.2.

2.3 Implicit Control

Herding seeks an expression for the input $\mathbf{u}$ such that the evaders go to their desired positions $\mathbf{x}^*$. Besides, the herding may need to accomplish other requirements, such as a desired transient response. This can be translated into designing $\mathbf{u}$ such that the evaders follow a desired dynamics f^*,

$$\dot{\mathbf{x}} = f^*(\mathbf{x}). \tag{2.5}$$

To ease the analysis we consider, without loss of generality, $\mathbf{x}^* = \mathbf{0}$ as the desired equilibrium point of the system. The function f^* can adopt any desired structure[2] with the constraint of being stable at the equilibrium point.

In order to control the evaders, the typical practice consists in finding a closed expression for $\mathbf{u}$ that depends only on the state and maps (2.4) into (2.5). Unfortunately, due to the complexity of the system, sometimes it is extremely challenging and time consuming to find such explicit mapping. Indeed, trying to find it generally yields to an implicit expression for the input. The analytical solution of the latter is not guaranteed, and the mechanisms to calculate such input do not certify that the desired properties prevail anymore. In contrast, we propose Implicit Control as a general control solution to achieve desired closed-loop dynamics despite the complexity of (2.4), overcoming the necessity of looking for a controller to perform the mapping from (2.4) to (2.5).

We study the conditions that allow to find $\mathbf{u}$ such that the evaders evolve according to f^*. Then, we propose a design procedure to solve the control input. The description is kept in general control terms since we believe that this procedure can be of interest in other control problems.

2.3.1 Control existence

Firstly, it is necessary to know if there exists a smooth input that makes the actual dynamics equal to the desired ones. For convenience, let define

$$h(\mathbf{x}, \mathbf{u}) = f(\mathbf{x}, \mathbf{u}) - f^*(\mathbf{x}) \tag{2.6}$$

as the working equation, shifting the analysis to that of looking for the existence of $\mathbf{u}$ such that $h(\mathbf{x}, \mathbf{u}) = \mathbf{0}$, i.e., looking for the existence of the roots of $h(\mathbf{x}, \mathbf{u})$.

At this point, a straightforward solution can be to compute the input using a numerical method to find the roots of h. This is a simple and powerful approach, but lacks of formal guarantees, depends on the numerical method and is computationally expensive since we must ensure that the numerical method finds the roots of h at all instants. Therefore, we will use this approach as a baseline to compare the Implicit Control solution.

In any case, the first step is to determine the existence of the roots of h, which is, in general, not trivial. We review some fundamentals of numerical analysis theory to show sufficient conditions to ensure existence and smoothness[3] of $\mathbf{u}$. The Implicit Function Theorem (Theorem 9.28 of [Rudin, 1976]), applied to h, offers a set of sufficient conditions to guarantee the existence and smoothness of $\mathbf{u}$ in a local region.

Theorem 1 (Adapted from Theorem 9.28 of [Rudin, 1976])**.** *Let*

$$h : I = X \times U \subset \mathbb{R}^M \times \mathbb{R}^N \longmapsto \mathbb{R}^N \tag{2.7}$$

[2]The choice of f^* as a state-dependent function is to follow the typical desired dynamics in regulation problems. However, the results can be extended to desired dynamics that depend on the input as well.

[3]Smoothness is not mandatory, but it is a desirable property.

a C^1-mapping, such that $h(\mathbf{x}^*, \mathbf{u}_0^*) = \mathbf{0}$ for some point $\mathbf{u}_0^* \in U$. Additionally, consider the Jacobian

$$\mathbf{J} = (\mathbf{J}_\mathbf{x} | \mathbf{J}_\mathbf{u}) = \begin{pmatrix} \frac{\partial h_1}{\partial \mathbf{x}_1} & \cdots & \frac{\partial h_1}{\partial \mathbf{x}_M} & \frac{\partial h_1}{\partial \mathbf{u}_1} & \cdots & \frac{\partial h_1}{\partial \mathbf{u}_N} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_M}{\partial \mathbf{x}_1} & \cdots & \frac{\partial h_M}{\partial \mathbf{x}_M} & \frac{\partial h_M}{\partial \mathbf{u}_1} & \cdots & \frac{\partial h_M}{\partial \mathbf{u}_N} \end{pmatrix} \tag{2.8}$$

such that $\mathbf{J}_\mathbf{u}$ is non-singular in the point $(\mathbf{x}^*, \mathbf{u}_0^*)$. Then, there exist $I^* \subset \mathbb{R}^M \times \mathbb{R}^N$ and $X^* \subset \mathbb{R}^M$, with $(\mathbf{x}, \mathbf{u}^*) \in I^*$ and $\mathbf{x} \in X^*$, having the following property: to every possible $\mathbf{x} \in X^*$ corresponds a unique $\mathbf{u}^*$ such that $(\mathbf{x}, \mathbf{u}^*) \in I^*$ and $h(\mathbf{x}, \mathbf{u}^*) = \mathbf{0}$.

The Theorem imposes three conditions to be fulfilled. First, there must exist an input, $\mathbf{u}_0^*$, which solves the control in $\mathbf{x}^*$. In the herding context, there must exist a stable configuration of the herders when the evaders are in their desired positions. A general condition to ensure this, for all $\mathbf{x} \in X$, is to have at least the same number of inputs than states to control. Otherwise, it is not always possible to control the system because there are less degrees of freedom than states to control. In the case of the precision herding problem, this means $n \geq m$, since both evaders and herders are first order entities in the space and we want to control all the evaders simultaneously; i.e., at all instants, all the herders contribute to the motion of all the evaders. Besides, a team of herders is needed so that, at equilibrium, the total repulsive force in each evader is zero. Nevertheless, we demonstrate in Section 2.6 that Implicit Control can also be used to control large-scale herds with only a few robotic herders if the control objective is the position of the centroid of the herd.

Second, h must be of class C^1 in $(\mathbf{x}^*, \mathbf{u}_0^*)$. If f^* is chosen of class C^1 in $\mathbf{x}^*$, then the condition is accomplished because f_j in (2.1) is of class C^1 for all j, so f in (2.4) is of class C^1.

The last condition requires the Jacobian of h with respect to $\mathbf{u}$, $\mathbf{J}_\mathbf{u}$, to be non-singular in the desired location. Since for $m \neq n$ the matrix is not square, the right Moore-Penrose inverse matrix $\mathbf{J}_\mathbf{u}^+ = \mathbf{J}_\mathbf{u}^\top (\mathbf{J}_\mathbf{u} \mathbf{J}_\mathbf{u}^\top)^{-1}$ is generally considered as the one to be non-singular. An alternative is to resort to the Constant Rank Theorem (Theorem 5.22 in [Lee, 2003]) and find a projection of lower dimension which locally preserves the properties of the Jacobians but achieving a square matrix. The use of the pseudoinverse can be seen as this projection. Given the aforementioned features of h and $\mathbf{J}_\mathbf{u}$, the last condition is accomplished in $\mathbf{x}^*$. Moreover, by restricting I to the subspace without collisions it is ensured that the two last conditions of Theorem 1 hold for all $\mathbf{x}$ and, therefore, the Theorem holds for all $\mathbf{x}$ in this subspace. Considering that each herder provokes a repulsive reaction in every evader, collision among herders and evaders will not happen. Similarly, since herders are the controlled entities, it is easy to prevent collisions among them.

Theorem 1 is an analytical tool to guarantee the necessary conditions to afford the Implicit Control proposal. Further analysis on the existence of input will depend on the particularities of the controlled system. Therefore, we consider the following assumption, so that a suitable input can be computed.

Assumption 1. *There exists $\mathbf{u}_0$ such that $h(\mathbf{x}, \mathbf{u}_0) = \mathbf{0}$, the functions f and f^* are of class C^1 and $\mathbf{J}_\mathbf{u}$ has rank m for all $(\mathbf{x}, \mathbf{u}) \in I$.*

The problems of uniqueness of solution or existence of local minima are not a concern since no cost function is being optimized. We underline this to emphasize that Implicit Control does not search for any local or global optimum, but for the roots of h.

2.3.2 Control design

The proposed design method consists in expanding the initial system in (2.4) with an input dynamics that converges to the roots of h. This transforms the problem to that of computing the input $\mathbf{u}$ as part of an expanded explicit system, described in continuous time and with analytical solution. To ease the presentation, we first assume $n = m$, generalizing to $n \neq m$ later.

The function h is determined by the desired closed-loop behavior f^*. In an ideal scenario, $h = 0$ always holds. However, achieving $h = 0$ for all time is generally not an easy task. Our solution consists in considering h as a dynamic system. The evolution of h over time is not determined a priori; therefore, the input dynamics can be designed such that the evolution of h follows a desired dynamics h^*,

$$\frac{dh(\mathbf{x}, \mathbf{u})}{dt} = h^*(\mathbf{x}, \mathbf{u}), \tag{2.9}$$

so that h converges to zero. From now on, we assume that h^* is chosen to make h stable. With this in mind, we propose the following expression for the input dynamics,

$$\dot{\mathbf{u}} = \mathbf{J}_{\mathbf{u}}^{-1}(h^*(\mathbf{x}, \mathbf{u}) - \mathbf{J}_{\mathbf{x}} f(\mathbf{x}, \mathbf{u}))\,, \tag{2.10}$$

where $\mathbf{J}_{\mathbf{x}}$ is the Jacobian of h with respect to $\mathbf{x}$ and $\mathbf{J}_{\mathbf{u}}$ is the Jacobian of h with respect to $\mathbf{u}$. This allows to build an explicit expanded system of the form

$$\begin{cases} \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \\ \dot{\mathbf{u}} = \mathbf{J}_{\mathbf{u}}^{-1}(h^*(\mathbf{x}, \mathbf{u}) - \mathbf{J}_{\mathbf{x}} f(\mathbf{x}, \mathbf{u})) \end{cases}. \tag{2.11}$$

The control problem is then reduced to analyzing the stability of the system in (2.11). With the proposed expansion the complete system becomes autonomous, i.e., the input becomes part of the state of the expanded system. As the structure in (2.11) is very general, we provide stability results for typical dynamics. The first one considers that the original system is Input-to-State Stable (ISS).

Theorem 2. *If the system defined in* (2.4) *is ISS and Assumption 1 holds, then the input dynamics in* (2.10) *ensures the convergence of h towards zero and achieves the desired closed-loop dynamics in* (2.5).

Proof. Applying the chain rule over dh/dt yields to

$$\frac{dh(\mathbf{x}, \mathbf{u})}{dt} = \mathbf{J}_{\mathbf{x}}(\mathbf{x}, \mathbf{u})\frac{d\mathbf{x}}{dt} + \mathbf{J}_{\mathbf{u}}(\mathbf{x}, \mathbf{u})\frac{d\mathbf{u}}{dt}, \tag{2.12}$$

where $\mathbf{J}_{\mathbf{x}}$ and $\mathbf{J}_{\mathbf{u}}$ are the Jacobians of h with respect to $\mathbf{x}$ and $\mathbf{u}$. By Assumption 1, $\mathbf{J}_{\mathbf{u}}$ is full rank. Then, the substitution of the input dynamics in (2.10) gives (2.9), meaning that $\dot{\mathbf{u}}$ imposes the desired dynamics over h.

This ensures that, at some point, $h = \mathbf{0}$. However, in the transient, $h \neq \mathbf{0}$ so the input $\mathbf{u}$ is not the one that achieves $f = f^*$. This non-zero difference can be seen as a perturbation in the input $\mathbf{u}$ of system (2.4). Nevertheless, since the system is ISS, the stability remains despite the perturbation [Khalil, 2014]. Due to the stability of h, the perturbation vanishes with time, and the expanded system in (2.11) makes (2.4) to converge to (2.5). □

The ISS property is common but particular to some systems. The next case of study aims at using Implicit Control for more general dynamics. First, we restrict the possible desired closed-loop behaviors to

$$f^*(\mathbf{x}) = -\mathbf{K}_f(\mathbf{x})\mathbf{x}, \tag{2.13}$$

where $\mathbf{K}_f(\mathbf{x})$ is a function which returns a positive definite matrix. In particular, $\mathbf{K}_f(\mathbf{x})$ can be a polynomial whose terms are all of even order. For instance, if $\mathbf{K}_f(\mathbf{x}) = \mathbf{K}_f$, then $f^*(\mathbf{x}) = -\mathbf{K}_f\mathbf{x}$ has a linear behavior with its corresponding exponential-like transient, desired settling time and stability properties. Similarly, let

$$h^*(\mathbf{x}, \mathbf{u}) = -\mathbf{K}_h(h(\mathbf{x}, \mathbf{u}))h(\mathbf{x}, \mathbf{u}), \tag{2.14}$$

with $\mathbf{K}_h(h(\mathbf{x}, \mathbf{u}))$ a function which returns a positive definite matrix as well.

Theorem 3. *Given Assumption 1, the definition of f^* in* (2.13) *and the definition of h^* in* (2.14)*, if matrices $\mathbf{K}_f(\mathbf{x})$ and $\mathbf{K}_h(h(\mathbf{x}, \mathbf{u}))$ are chosen in such a way that*

$$\mathbf{K} = \begin{pmatrix} -\mathbf{K}_f(\mathbf{x}) & 0.5\mathbf{I} \\ 0.5\mathbf{I} & -\mathbf{K}_h(h(\mathbf{x}, \mathbf{u})) \end{pmatrix} \tag{2.15}$$

is negative definite, then the system in Eq. (2.11) *is Globally Asymptotically Stable (GAS).*

Proof. Consider $V = \frac{1}{2}\mathbf{x}^\top\mathbf{x} + \frac{1}{2}h^\top h$ as a Lyapunov candidate function, whose derivative is

$$\dot{V} = \mathbf{x}^\top\dot{\mathbf{x}} + h^\top\dot{h} = \mathbf{x}^\top f + h^\top h^* \tag{2.16}$$

and where $\dot{h}$ is substituted by (2.9) following the same steps of the proof of Theorem 2. Using the definition of h in (2.6) yields to

$$\dot{V} = \mathbf{x}^\top h + \mathbf{x}^\top f^* + h^\top h^*. \tag{2.17}$$

Considering f^* and h^* from Eqs. (2.13) and (2.14), $\dot{V}$ is

$$\dot{V} = \begin{pmatrix} \mathbf{x}^\top & h^\top \end{pmatrix} \mathbf{K} \begin{pmatrix} \mathbf{x}^\top & h^\top \end{pmatrix}^\top. \tag{2.18}$$

Then, designing $\mathbf{K}_f(\mathbf{x})$ and $\mathbf{K}_h(h(\mathbf{x}, \mathbf{u}))$ such that $\mathbf{K}$ is negative definite guarantees that (2.11) is GAS, and both h and $\mathbf{x}$ go to zero. □

In practice, $||\mathbf{K}_h|| \gg ||\mathbf{K}_f||$ imposes the convergence of h to be much faster than the desired closed-loop dynamics, so the evaders will behave following f^*.

2.3.3 Control extensions

Assuming $n = m$ is a restrictive condition. We deal with the generalization to $n \neq m$ by replacing $\mathbf{J}_{\mathbf{u}}^{-1}$ with the pseudoinverse $\mathbf{J}_{\mathbf{u}}^{+}$. In the case that there are some states whose dynamics are not explicitly dependent on $\mathbf{u}$ (for instance, the velocity of the evaders might not depend on herders' position, but their acceleration), then a linear transformation $\mathbf{T}$ over h can be applied

$$\bar{h}(\bar{\mathbf{x}}, \mathbf{u}) = \mathbf{T}h(\mathbf{x}, \mathbf{u}). \tag{2.19}$$

The purpose is to create a new group of states $\bar{\mathbf{x}}$ combining all the states which do not depend on the input with some of the states which do depend on the input. Then, Implicit Control can be applied, over the new function $\bar{h}$, to control states which do not explicitly depend on the input.

Furthermore, all the previous results can be extended to dynamic references. Let consider a desired state reference $\mathbf{x}^*$ with desired dynamics $\dot{\mathbf{x}}^*$.

Corollary 1. *Assuming that $\dot{\mathbf{x}}^*$ is smooth and bounded, the results in Theorems 2-3 hold if the function h is redefined as*

$$h(\mathbf{x}, \mathbf{u}) = f(\mathbf{x}, \mathbf{u}) - f^*(\mathbf{x}) - \dot{\mathbf{x}}^*. \tag{2.20}$$

Proof. By assuming that $\dot{\mathbf{x}}^*$ is smooth and bounded we maintain the existence and smoothness of the input. The original system is still described by (2.4) but the control objective changes to

$$f^*(\mathbf{x}, \mathbf{u}) = f(\mathbf{x}, \mathbf{u}) - \dot{\mathbf{x}}^*. \tag{2.21}$$

The expression comes from imposing that the error dynamics follow f^*. This information can be incorporated in the design of the function h by adding the known time-varying term in (2.21) to compensate the dynamic reference. The time derivative of $\dot{\mathbf{x}}^*$ is implicitly included in the Jacobians of the new h, therefore the structure in system (2.11) remains. Moreover, since $\dot{\mathbf{x}}^*$ does not depend on $\mathbf{u}$, the Jacobians in Theorems 2-3 do not change and the statements in Theorems 2-3 hold. □

The importance of Corollary 1 is that our herding solution can be seen as a generalization of previous herding works [Pierson and Schwager, 2018] [Song et al., 2021] where the purpose is to drive the evaders to a certain region. By imposing a time-varying reference, herders not only drive evaders towards particular positions but can impose a specific formation. Besides, note that no particular restrictions regarding the initial configuration of the entities is specified. Nevertheless, it is convenient that the herders surround the evaders to avoid escapes before the beginning of the herding, so in Subsection 2.5.2 we develop a secure caging stage.

From an algorithmic point of view, the calculation of the control input is very simple. At each instant, the controller receives $\mathbf{x}$ and $\mathbf{u}$ from an observer and/or from measurements. Then, $f^*(\mathbf{x})$ is computed, which, together with the dynamic model of the evaders $f(\mathbf{x}, \mathbf{u})$ (e.g., (2.2) or (2.3)), allows to compute $h^*(\mathbf{x}, \mathbf{u})$ with Eq. (2.14). Besides, the Jacobians $\mathbf{J}_{\mathbf{x}}$

and $\mathbf{J_u}$ can be calculated either analytically or numerically, depending on their complexity. Finally, $\dot{\mathbf{u}}$ is computed from Eq. (2.11). Regarding the estimation of $\mathbf{x}$ and $\mathbf{u}$, another advantage of Implicit Control is that $\mathbf{u}$ now belongs to the state of an expanded system. Therefore, it is possible to apply the same estimation techniques as with $\mathbf{x}$, robustifying the herding solution.

2.4 Adding adaptation

In the previous section perfect knowledge of the evaders' dynamics is assumed. However, in real problems there exist sources of uncertainty. We derive a general adaptation law which overcomes this issue and preserves the control properties.

Consider the parameters θ_j in Eqs. (2.2) and (2.3). Let $\theta = \text{diag}(\theta_1, \ldots, \theta_m) \in \mathbb{R}^{m \times m}$, then the original system (2.4) is rewritten to the following, with a little abuse of notation for the sake of readability,

$$\dot{\mathbf{x}} = f_\theta(\mathbf{x}, \mathbf{u}, \theta) = \theta f(\mathbf{x}, \mathbf{u}). \tag{2.22}$$

We assume a linear dependency with respect to the uncertain parameters, and that the parameters are constant. This assumption is typical in adaptive control theory and, in the case of Implicit Control, implies that the gradients with respect to the parameters do not depend on the parameters, which is the key that yields to the proposed adaptation law. Nevertheless, despite simple, linear parameters can still model the aggressiveness in the repulsion provoked by the herders. Besides, notice that $\theta_j > 0$ because otherwise the evaders do not move or are not repelled by the herders.

We represent the uncertainty in θ by the vector of estimates $\hat{\theta}$, defining the error between the real and estimated parameters as $\tilde{\theta} = \theta - \hat{\theta}$. From now on, the symbols $\hat{\bullet}$ and $\tilde{\bullet}$ denote the quantities that depend on $\hat{\theta}$ and $\tilde{\theta}$. In general, $\tilde{\theta} \neq \mathbf{0}$ so Implicit Control no longer guarantees stability. The difference between actual and estimated dynamics can be defined by a new function

$$\tilde{h}(\mathbf{x}, \mathbf{u}, \tilde{\theta}) = h(\mathbf{x}, \mathbf{u}, \theta) - \hat{h}(\mathbf{x}, \mathbf{u}, \hat{\theta}) = \tilde{\theta} f(\mathbf{x}, \mathbf{u}). \tag{2.23}$$

This is a standard adaptive control problem, for which there exist different solutions for both linear and nonlinear systems [Khalil, 2014]. Nevertheless, we propose a different adaptation law, following the principles of Implicit Control, that simplifies the stability analysis.

The objective of the adaptation law is to design $\dot{\hat{\theta}}$ such that $\tilde{h}$ converges to zero while the closed-loop behavior of the system is preserved. The structure of the problem, and, in particular, of $\tilde{h}$, is similar to that of Implicit Control. We can follow the same input design procedure of Implicit Control, in this case over $\tilde{h}$, to design an adaptation law of the form

$$\dot{\hat{\theta}} = -\tilde{\mathbf{J}}_\theta^{-1}(\tilde{h}^* + \hat{\mathbf{J}}_\mathbf{x}\dot{\hat{\mathbf{x}}} + \hat{\mathbf{J}}_\mathbf{u}\dot{\mathbf{u}}). \tag{2.24}$$

Here,

$$\hat{\mathbf{J}}_{\mathbf{x}}=\begin{pmatrix} \frac{\partial \hat{h}_1}{\partial \mathbf{x}_1} \cdots \frac{\partial \hat{h}_1}{\partial \mathbf{x}_m} \\ \vdots \ \ddots \ \vdots \\ \frac{\partial \hat{h}_m}{\partial \mathbf{x}_1} \cdots \frac{\partial \hat{h}_m}{\partial \mathbf{x}_m} \end{pmatrix}, \hat{\mathbf{J}}_{\mathbf{u}}=\begin{pmatrix} \frac{\partial \hat{h}_1}{\partial \mathbf{u}_1} \cdots \frac{\partial \hat{h}_1}{\partial \mathbf{u}_n} \\ \vdots \ \ddots \ \vdots \\ \frac{\partial \hat{h}_m}{\partial \mathbf{u}_1} \cdots \frac{\partial \hat{h}_m}{\partial \mathbf{u}_n} \end{pmatrix}, \tag{2.25}$$

$\tilde{\mathbf{J}}_\theta = \text{diag}(f)$, and $\tilde{h}^*$ is a free-design function.

As in the design of the input dynamics, $\tilde{h}^*$ determines the desired dynamic behavior for the derivative of (2.23). Concatenating the new set of equations yields a new expanded system

$$\begin{cases} \dot{\mathbf{x}} = \theta f \\ \dot{\mathbf{u}} = \hat{\mathbf{J}}_{\mathbf{u}}^{+}\left(\hat{h}^* - \hat{\mathbf{J}}_{\mathbf{x}}\dot{\hat{\mathbf{x}}}\right) \\ \dot{\hat{\theta}} = -\tilde{\mathbf{J}}_\theta^{-1}(\tilde{h}^* + \hat{\mathbf{J}}_{\mathbf{x}}\dot{\hat{\mathbf{x}}} + \hat{\mathbf{J}}_{\mathbf{u}}\dot{\mathbf{u}}) \end{cases}. \tag{2.26}$$

Acknowledging the similarities between (2.26) and (2.11), we extend the results in Theorems 2 and 3.

Proposition 1. *If the system defined in* (2.4) *is ISS and Assumption 1 holds, then the adaptive law in* (2.24) *preserves the properties of the Implicit Control if*

$$\tilde{h}^* = h_\theta^* - \dot{h} + \hat{\mathbf{J}}_{\mathbf{x}}(\dot{\mathbf{x}} - \dot{\hat{\mathbf{x}}}), \tag{2.27}$$

where h_θ^* is a free-design function chosen to make $\tilde{h}$ stable.

Proof. The proof follows some of the steps in Theorem 2. Let consider the chain rule expansion of $d\tilde{h}/dt$,

$$\frac{d\tilde{h}}{dt} = \frac{\partial \tilde{h}}{\partial \mathbf{x}}\dot{\mathbf{x}} + \frac{\partial \tilde{h}}{\partial \mathbf{u}}\dot{\mathbf{u}} + \frac{\partial \tilde{h}}{\partial \tilde{\theta}}\dot{\tilde{\theta}}. \tag{2.28}$$

Since θ is constant, $\dot{\tilde{\theta}} = -\dot{\hat{\theta}}$ and the substitution of Eq. (2.24) gives

$$\frac{d\tilde{h}}{dt} = \mathbf{J}_{\mathbf{x}}\dot{\mathbf{x}} - \hat{\mathbf{J}}_{\mathbf{x}}\dot{\mathbf{x}} + \mathbf{J}_{\mathbf{u}}\dot{\mathbf{u}} - \hat{\mathbf{J}}_{\mathbf{u}}\dot{\mathbf{u}} + \tilde{h}^* + \hat{\mathbf{J}}_{\mathbf{x}}\dot{\hat{\mathbf{x}}} + \hat{\mathbf{J}}_{\mathbf{u}}\dot{\mathbf{u}}. \tag{2.29}$$

Note that $\mathbf{J}_{\mathbf{x}}\dot{\mathbf{x}} + \mathbf{J}_{\mathbf{u}}\dot{\mathbf{u}}$ is equal to $\dot{h}$, and the $\hat{\mathbf{J}}_{\mathbf{u}}\dot{\mathbf{u}}$ terms cancel out. Therefore, Eq. (2.29) can be written as

$$\frac{d\tilde{h}}{dt} = \dot{h} + \tilde{h}^* - \hat{\mathbf{J}}_{\mathbf{x}}(\dot{\mathbf{x}} - \dot{\hat{\mathbf{x}}}) = h_\theta^*, \tag{2.30}$$

where the later follows from the definition of $\tilde{h}^*$ in Eq. (2.27). Now, let consider the chain rule expansion of $d\hat{h}/dt$,

$$\frac{d\hat{h}}{dt} = \frac{\partial \hat{h}}{\partial \mathbf{x}}\dot{\mathbf{x}} + \frac{\partial \hat{h}}{\partial \mathbf{u}}\dot{\mathbf{u}} + \frac{\partial \hat{h}}{\partial \hat{\theta}}\dot{\hat{\theta}}. \tag{2.31}$$

If we use the definitions of $\dot{\mathbf{u}}$ and $\dot{\hat{\theta}}$ in Eq. (2.26), Eq. (2.31) reads

$$\frac{d\hat{h}}{dt} = \hat{\mathbf{J}}_{\mathbf{x}}\dot{\mathbf{x}} + \hat{h}^* - \hat{\mathbf{J}}_{\mathbf{x}}\dot{\hat{\mathbf{x}}} - \tilde{h}^* - \hat{\mathbf{J}}_{\mathbf{x}}\dot{\hat{\mathbf{x}}} - \hat{h}^* + \hat{\mathbf{J}}_{\mathbf{x}}\dot{\hat{\mathbf{x}}}. \tag{2.32}$$

By exploiting $\tilde{h}^*$ in Eq. (2.27), we get

$$\frac{d\hat{h}}{dt} = \dot{h} - h_\theta^*. \tag{2.33}$$

Finally, let consider the chain rule expansion of dh/dt

$$\frac{dh}{dt} = \frac{\partial h}{\partial \mathbf{x}}\dot{\mathbf{x}} + \frac{\partial h}{\partial \mathbf{u}}\dot{\mathbf{u}} + \frac{\partial h}{\partial \theta}\dot{\theta} = \mathbf{J}_{\mathbf{x}}\dot{\mathbf{x}} + \mathbf{J}_{\mathbf{u}}\dot{\mathbf{u}}. \tag{2.34}$$

The substitution of $\dot{\mathbf{u}}$ in Eq. (2.26) gives

$$\frac{dh}{dt} = \mathbf{J}_{\mathbf{x}}\dot{\mathbf{x}} + \mathbf{J}_{\mathbf{u}}\hat{\mathbf{J}}_{\mathbf{u}}^{+}\left(\hat{h}^* - \hat{\mathbf{J}}_{\mathbf{x}}\dot{\hat{\mathbf{x}}}\right) = \mathbf{J}_{\mathbf{x}}(\dot{\mathbf{x}} - \dot{\hat{\mathbf{x}}}) + h^* = \mathbf{J}_{\mathbf{x}}\dot{\tilde{h}} + h^*. \tag{2.35}$$

With Eqs. (2.30), (2.33) and (2.35) we use the same arguments in the proof of Theorem 2. Let $\tilde{h} \neq \mathbf{0}$ be considered as a perturbation. Due to Eq. (2.30), $\tilde{h}$ converges to zero at some point. In the meantime, since the system is ISS, the stability remains. Besides, $\tilde{h}$ is stable, so the perturbation vanishes with time. Then, according to Eq. (2.33), $\dot{\hat{h}} = \dot{h}$ and, according to Eq. (2.35), $\dot{h} = h^*$. Therefore, we recover the expanded system in Eq. (2.11) and the results of Theorem 2 hold. □

Proposition 2. *Given Assumption 1,*

$$h_\theta^*(\mathbf{x}, \mathbf{u}) = -\mathbf{K}_\theta(\mathbf{x}, \mathbf{u})\tilde{h} \tag{2.36}$$

with $\mathbf{K}_\theta(\mathbf{x}, \mathbf{u})$ *a function which returns a positive definite matrix, and* $\mathbf{K}(h(\mathbf{x}, \mathbf{u}))$ *defined as in* (2.15)*, if*

$$\bar{\mathbf{K}} = \begin{pmatrix} -\mathbf{K}_f(\mathbf{x}) & 0.5\mathbf{I} & \mathbf{0} \\ 0.5\mathbf{I} & -\mathbf{K}_h(h(\mathbf{x}, \mathbf{u})) & -0.5\mathbf{J}_{\mathbf{x}}\mathbf{K}_\theta(\mathbf{x}, \mathbf{u}) \\ \mathbf{0} & -0.5\mathbf{J}_{\mathbf{x}}\mathbf{K}_\theta(\mathbf{x}, \mathbf{u}) & -\mathbf{K}_\theta(\mathbf{x}, \mathbf{u}) \end{pmatrix} \tag{2.37}$$

is negative definite, then the adaptive law preserves the stability properties of Theorem 3.

Proof. The proof follows from Theorem 3 by considering

$$V = \frac{1}{2}\mathbf{x}^\top\mathbf{x} + \frac{1}{2}h^\top h + \frac{1}{2}\tilde{h}^\top\tilde{h} \tag{2.38}$$

as a Lyapunov candidate function. Developing its derivative and using Eqs. (2.30) and (2.35) from Proposition 1, we obtain

$$\begin{aligned} \dot{V} =& \mathbf{x}^\top h + \mathbf{x}^\top f^* + h^\top\dot{h} + \tilde{h}^\top\dot{\tilde{h}} = \\ & \mathbf{x}^\top h - \mathbf{x}^\top\mathbf{K}_f\mathbf{x} - h^\top\mathbf{J}_{\mathbf{x}}\mathbf{K}_\theta\tilde{h} - h^\top\mathbf{K}_h h - \tilde{h}^\top\mathbf{K}_\theta\tilde{h}. \end{aligned} \tag{2.39}$$

As in Theorem 3, the latter reads

$$\dot{V} = \begin{pmatrix} \mathbf{x}^\top & h^\top & \tilde{h}^\top \end{pmatrix} \bar{\mathbf{K}} \begin{pmatrix} \mathbf{x}^\top & h^\top & \tilde{h}^\top \end{pmatrix}^\top. \tag{2.40}$$

Then, if $\mathbf{K}, \mathbf{K}_\theta$ are designed such that $\bar{\mathbf{K}}$ is negative definite, then the adaptation law preserves the stability properties of Theorem 3. □

The adaptation law, then, evolves the estimates towards the values which ensure the correct behavior of the control. Remarkably, the convergence of $\tilde{h}$ to zero implies that $\hat{h}$ converges to h. Therefore, the convergence of $\hat{h}$ to zero implies the convergence of h

to zero, and the evaders behave as desired. In practice, to implement the controller, a slight change in Eq. (2.26) is required to compute $\tilde{h}^*$. Since $\dot{\mathbf{x}} = \theta f$ is not available, an approximation (e.g., by finite differences) must be used to obtain $\dot{\mathbf{x}}$. Note that this only affects the implementation, as long as the stability analysis involves the real dynamics of the evaders.

Example. We provide a brief description of how to design the control and adaptation gains.

The first step is to set the desired behavior of the herd. Theorem 3 and Proposition 2 allow for several possible choices, but to keep it simple, consider a desired settling time of $1/\tau = 12$s and an exponential transient according to m first order independent systems. Thus,

$$f^*(\mathbf{x}) = -\mathbf{K}_f(\mathbf{x})\mathbf{x} = -\mathbf{K}_f\mathbf{x} = -(3\tau\mathbf{I})\mathbf{x} = -0.25\mathbf{I}\mathbf{x}. \tag{2.41}$$

Then, according to Theorem 3, $\mathbf{K}_h(h(\mathbf{x},\mathbf{u}))$ in Eq. (2.15) must be such that $\mathbf{K}$ is negative definite. To preserve simplicity, let $\mathbf{K}_h(h(\mathbf{x},\mathbf{u})) = \mathbf{K}_h$ be a constant matrix. Under these conditions, for example $\mathbf{K}_h = 50\mathbf{I}$ works.

In the same way, if an adaptation law is desired, then $\mathbf{K}_\theta(\mathbf{x},\mathbf{u})$ in Eq. (2.37) must be such that $\bar{\mathbf{K}}$ is negative definite. In the herding case, the speed of evaders and herders is bounded, so $\mathbf{J}_\mathbf{x}$ is bounded and there always exist a matrix $\mathbf{K}_\theta(\mathbf{x},\mathbf{u})$ that makes $\bar{\mathbf{K}}$ negative definite. To preserve simplicity again, let $\mathbf{K}_\theta(\mathbf{x},\mathbf{u}) = \mathbf{K}_\theta$ be a constant matrix. For instance, $\mathbf{K}_\theta = 200\mathbf{I}$ works in the examples of Section 2.6. In general, starting from $\mathbf{K}_f$, choosing the other gains such that $||\mathbf{K}_f|| < ||\mathbf{K}_h|| < ||\mathbf{K}_\theta||$ will make the system stable. Nevertheless, there are no particular restrictions on the gains, providing a lot of design possibilities. If needed/desired, $\mathbf{K}_f$ might not be a constant, nor f^* a linear function of the state. Similarly, $\mathbf{K}_h$ can change with time to preserve the negative definiteness of $\mathbf{K}$, and the same applies to $\mathbf{K}_\theta$.

2.5 Herding solution

Implicit Control, along with the adaptation law, constitutes the core of our proposal. Nonetheless, to have a complete herding solution, additional issues are considered.

2.5.1 Distributed estimator

Up to now, the solution assumes that herders know the state and input of the system perfectly, i.e., the position of all the herders and evaders. This subsection describes a distributed estimator that fulfills this assumption. Particularly, we develop an Extended version of the Distributed Kalman Filter [Olfati-Saber, 2007a] (E-DKF), tailored to work with the Implicit Control expanded dynamics. The objective is, for all the herders, to have an accurate estimation of evaders' and herders' position, aggregated in the vector $\xi = [\mathbf{x}_1^\top, \ldots, \mathbf{x}_m^\top, \mathbf{u}_1^\top, \ldots, \mathbf{u}_n^\top]^\top$. Notice that, in contrast with typical DKFs, the control

inputs must be included as quantities to be estimated. In typical control solutions this is not possible because the input has no dynamics because it is just an algebraic expression.

Implicit Control addresses this issue by providing the input dynamics. In particular, for the prediction step of the filter we consider

$$\dot{\xi} = \left.\frac{\partial \Sigma(\xi)}{\partial \xi}\right|_{\xi=\xi(t)} + \mathbf{w}, \tag{2.42}$$

where $\Sigma(\xi)$ denotes the expanded system in (2.11) and $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ is a zero mean Gaussian noise with covariance $\mathbf{Q}$. Note that the linearization in Eq. (2.42) is necessary because the overall expanded dynamics are strongly nonlinear, so we use $\frac{\partial \Sigma(\xi)}{\partial \xi}$ evaluated at $\xi = \xi(t)$. Since the evaders' dynamics are deterministic, $\mathbf{Q}$ will have small values, useful to mitigate the effects of the uncertainty during the adaptation law's transient.

For the filter update, it is important to note that the herders' measurements will depend on its position, i.e., a herder will only be able to measure the entities that are close to it. Therefore, the measurement equation is of the form

$$\mathbf{z}_i = \mathbf{H}_i(\xi)\xi + \mathbf{v}_i, \tag{2.43}$$

where $\mathbf{z}_i$ is the measurement taken by herder i, $\mathbf{v}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_i)$ is the measurement noise with covariance $\mathbf{R}_i$, and $\mathbf{H}_i$ depends on ξ. This dependency makes the dimension of $\mathbf{z}_i$ to change at each filter iteration, complicating the implementation.

The solution lies in using an information form of the E-DKF so, at each instant, each herder has a vector of measurements and a vector of booleans. The latter says whether an entity has been sensed or not, whereas the former contains the actual measurements and zeroes for the rest of entities, i.e., $\mathbf{z}_i \in \mathbb{R}^{2m+2n}$. Given the two vectors, the information form of the E-DKF permits to add up all the measurements in common state coordinates, only adding those measurements where the boolean vector is equal to one. Besides, communication among herders happens only if the distance between two herders is less than d_c. Thus, the estimator is input-dependent in the topology as well.

It is noteworthy that the stochastic nature of the estimator can absorb discrepancies from the adaptation and the unmodeled perturbations. This, together with the information form of the filter, permits to run the control and the estimation at different frequencies. The control loop can be the fastest while the estimation runs slower to account for communication bandwidth issues. Regarding the influence of the estimator in the overall stability of the system, if the estimator is designed to converge sufficiently fast, from a practical point of view its influence in the controller can be overlooked, and the controller works with accurate states and inputs. Regarding global awareness, the estimator still assumes a common reference frame for all the agents (herders and evaders) as well as knowledge of their number. Nevertheless, there are works available that deal with the distributed computation of these quantities [Franceschelli and Gasparri, 2013, Montijano et al., 2014a].

2.5.2 Caging the evaders

To avoid escapes, herders must be close to the evaders before beginning the precision herding using Implicit Control. This is not a complete solution because robots might not be deployed in the proximity of the evaders. Therefore, we propose a caging stage to prevent these escapes.

We consider a caging strategy based on the mean position and covariance of the herd. We denote them by $\mathbf{x}_{avg} := \frac{1}{m}\sum_j \mathbf{x}_j$ and $\mathbf{P}_{avg} := \frac{1}{m}\sum_j (\mathbf{x}_j - \mathbf{x}_{avg})(\mathbf{x}_j - \mathbf{x}_{avg})^\top$. The combination of both gives a description of the herd in terms of an ellipsoid centered in $\mathbf{x}_{avg}$. The geometry of the setup for a 2D herding is depicted in Fig. 2.2. It can be observed that the ellipse (in red) delimits a region around the evaders (in green) which suggests a uniform distribution of the herders along the perimeter. However, these locations might be too close to the evaders. Therefore, it is convenient to use an augmented version of the ellipse, $\mu_1\mathbf{P}_{avg}$ with $\mu_1 > 1$, to place the herders sufficiently far from the evaders. We do not make any assumptions about the initial distribution of the evaders. For instance, a very spread distribution of evaders with a large $\mathbf{P}_{avg}$ will only imply that it will take more time for the herders to surround the evaders. Moreover, μ_1 can be tuned so that herders provoke weak repulsive forces in the evaders. During the caging, evaders will not escape, and afterwards, the herders will approach the evaders to provoke large enough repulsive forces to steer the herd. Anyway, this is transparent for both the caging and the precise herding stage.

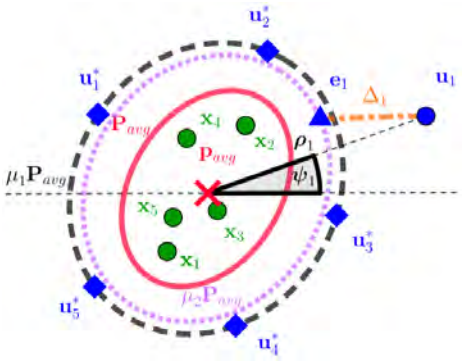


Figure 2.2: Geometrical view of the caging stage. The evaders (green dots) are in the region determined by their mean position and covariance (in red). To avoid escapes and high repulsive forces, the herder (blue dot) must be placed in the perimeter (blue diamonds) of an augmented ellipse (black-dashed). Besides, since there is uncertainty, CBFs prevent the herders to enter in the magenta ellipse, computing the closest point to its perimeter (blue triangle).

To solve the caging, we propose a controller, adapted from [Montijano et al., 2013], that steers the herders towards an even distribution along the perimeter of $\mu_1\mathbf{P}_{avg}$. The position of herder i, in polar coordinates with respect to the ellipse, is driven by

$$\dot{\psi}_i = (\psi_{i+1} - \psi_i) + (\psi_{i-1} - \psi_i) \tag{2.44}$$

$$\dot{\rho}_i = \rho_i - \rho_i^* \tag{2.45}$$

where ψ_i and ρ_i are the angular and radial positions, and $i+1$, $i-1$ denote the left and right closest herders. The policy is inspired in the well known N-bugs problem, that evenly distributes N agents over a closed curve. An even distribution of herders along the perimeter of an ellipse requires the herders to be distributed with a uniform angle. The agreement in the angle is given by Eq. (2.44). Meanwhile, to each angle corresponds a certain radial position to place herders over the desired ellipse, given by ρ_i^*. Therefore, the controller of the radial position is just the linear feedback controller in Eq. (2.45). For formal guarantees of convergence, we refer to [Montijano et al., 2013].

Herder i only needs to measure the closest left and right neighbor. These commands are then translated to Cartesian coordinates. Under this control policy the herders arrive to equally-distributed positions along the perimeter of the ellipse, denoted by $\mathbf{u}_i^*$ and depicted in blue in Fig. 2.2.

Due to perturbations or unmodeled phenomena, the caging might not prevent herders to move too close to the evaders. To solve this issue, we propose the use of CBFs [Wang et al., 2017]. The controller in Eqs. (2.44)-(2.45) computes the velocity $\dot{\mathbf{u}}_i^{nom}$, which is then translated into Cartesian coordinates. This is called the nominal action. The nominal action might move the herder towards a region where the interaction with the evaders is not convenient. We define this region by $\mu_2\mathbf{P}_{avg}$, where $\mu_1 > \mu_2 > 1$, and it is depicted in purple in Fig. 2.2.

To correct the nominal action, we use a quadratic program based on CBFs. In particular, at each instant, the closest point in the perimeter of the ellipse $\mu_2\mathbf{P}_{avg}$ to herder i is computed, denoted by $\mathbf{e}_i$ [Eberly, 2011]. Then, an optimization process [Wang et al., 2017] is carried out,

$$\dot{\mathbf{u}}_i^{cbf} = \underset{\dot{\mathbf{u}}_i^{cbf}}{\arg\min} \; ||\dot{\mathbf{u}}_i^{cbf} - \dot{\mathbf{u}}_i^{nom}||^2 \tag{2.46a}$$

$$s.t. \quad L_f h(\Delta_i) + L_g h(\Delta_i)\dot{\mathbf{u}}_i^{cbf} \geq -\kappa(h(\Delta_i)) \tag{2.46b}$$

Here, $\Delta_i = \mathbf{u}_i - \mathbf{e}_i$, depicted in orange in Fig. 2.2. Moreover, $\dot{\mathbf{u}}_i^{cbf}$ is the action to be optimized, $h(\Delta_i)$ is the control barrier function, $\kappa(h(\Delta_i))$ is a class $\mathcal{K}$ function and L_f, L_g denote the Lie derivatives. It is important to remark that the abuse of notation with function h is to fit the typical notation of CBFs [Wang et al., 2017].

Finally, a high-level algorithm of the complete herding sequence is described in Algorithm 1.

Algorithm 1 Complete herding for herder i

```
Receive x_avg and P_avg
while True do
    Compute estimate ξ̄_i with E-DKF
    if herders are far from evaders then
        Do caging: compute ψ̇_i, ρ̇_i, calculate u̇_i^cbf
    else
        Do precise herding: compute u̇_i, θ̂̇_i with Adaptive Implicit Control, and update θ̂_i
    end if
    Update u_i
end while
```

2.6 Simulation results

The purpose of these simulations is to validate and demonstrate the success of the proposal against challenging situations using the models in Section 2.2. The herding solution is formed by different parts, so the simulations will be conducted progressively to analyze each individual component.

2.6.1 Implicit Control and adaptation law

The first case of study consists in the herding of 5 evaders by 5 herders. The herders receive perfect measurements and they are already deployed near the evaders. The details are in Table 2.2. We set $\mathbf{K}_f = 0.25\mathbf{I}_{2m}$, yielding a settling time of 12s with an exponential transient according to $2m$ first order independent systems. To ensure that the conditions of Theorem 3 and Proposition 2 hold, we set $\mathbf{K}_h = 50\mathbf{I}_{2m}$ and $\mathbf{K}_\theta = 200\mathbf{I}_{2m}$. For the sake of comparison, we use the Levenberg-Marquardt (LM) numerical method [Marquardt, 1963] as a baseline to compute the control input.

Table 2.1: List of symbols in the figures.

Symbol	Meaning
	Desired positions

Symbol	Meaning	Symbol	Meaning
	Initial position herders		Initial position evaders
	Final position herders		Final position evaders
	Trajectories herders		Trajectories evaders

Table 2.2: Parameters of the simulations.

θ_j (Inv.)	θ_j (Exp.)	β_j	σ_j	d_{min}	T	v_{max}	d_m	d_c
1.0	0.5	0.5	2.0	1.0m	10ms	0.4m/s	6.5m	6.5m

The first row of Fig. 2.3 shows the trajectories followed by herders and evaders for the different test cases. Implicit Control (IC) is able to herd the evaders successfully, achieving almost the same trajectories obtained by the numerical baseline in all the experiments. Conversely, the trajectories of the herders present some differences depending on the motion model, which highlights the complexity of the control problem at hand, greater for the Exponential Model due to the switching dynamics. The second row of Fig. 2.3 represents the evolution over time of the evaders' position error. Implicit Control achieves the desired settling time of 12s and the state evolves as an exponential function according to the imposed closed-loop dynamics. Interestingly, it is corroborated that the Implicit Control solution does not assign any particular evader to any robotic herder, so herders just move

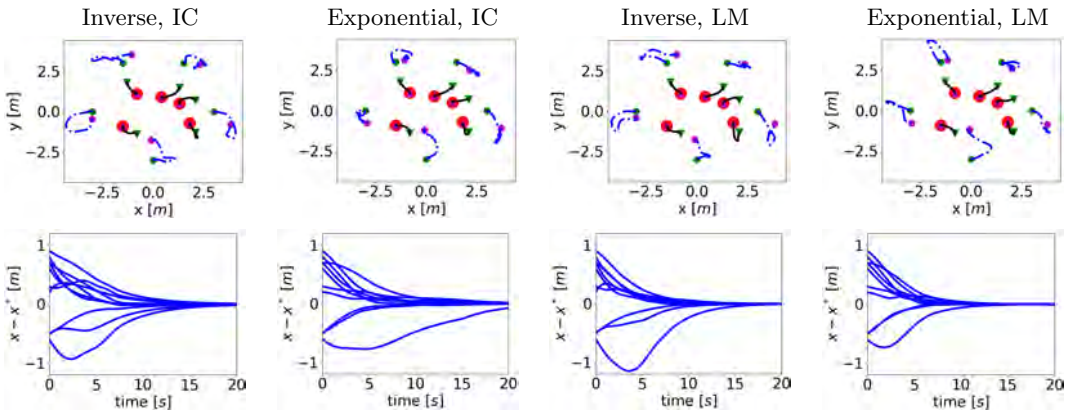


Figure 2.3: Simulation results of the herding of 5 evaders by 5 herders. The first row shows the trajectories followed by the herders and the evaders. The symbols are explained in Table 2.1. The second row presents the evolution of the error between desired and current position of the evaders.

to steer all the evaders towards their particular assigned positions *simultaneously*. All herders are contributing at all instants to the motion of all evaders. But, to achieve a stable equilibrium, a team of herders is needed because otherwise the herding can not be simultaneous, as it is seen in the second row of Fig. 2.3. At equilibrium, each herder is compensating the repulsive forces provoked by the other herders, reason why the evaders end up in the desired positions with zero velocity.

The only discrepancy between Implicit Control and the numerical baseline is in the first instants of the transient. The numerical baseline applies the "correct input" from the beginning, because at each instant it iterates until it solves $f(\mathbf{x}, \mathbf{u}) - f^*(\mathbf{x}) = \mathbf{0}$. Meanwhile, Implicit Control requires some time to converge to $f(\mathbf{x}, \mathbf{u}) - f^*(\mathbf{x}) = \mathbf{0}$, which is when the "correct input" is applied. Therefore, what only differs between methods is the shape of the transient in the first instants. Then, the total settling time with the Implicit Control is at most the settling time of $f^*(\mathbf{x})$ plus the time $h(\mathbf{x}, \mathbf{u})$ takes to converge to zero. If $\mathbf{K}_h$ is large enough, then the latter is negligible. This is why the differences in the transient are minimal in the second row of Fig. 2.3. Meanwhile, the differences in the input are much greater, as it is seen in Fig. 2.4. The controlled system is strongly input-nonaffine, so there are many roots of h that solve $h(\mathbf{x}, \mathbf{u}) = \mathbf{0}$. After Implicit Control converges to $h(\mathbf{x}, \mathbf{u}) = \mathbf{0}$, herders are in a different configuration with respect to the herders driven by the numerical baseline,

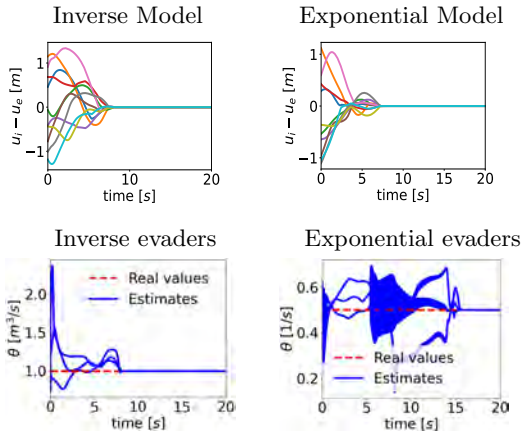


Figure 2.4: The top row shows the difference in the control input calculated by Implicit Control and the numerical baseline ($\mathbf{u}_i$ and $\mathbf{u}_e$ respectively). An arbitrary color has been assigned to each input for the sake of visibility. The bottom row shows the adaptation results of the herding of 5 evaders by 5 herders, with the evolution of the estimated parameters.

and this difference implies different roots for $h(\mathbf{x}, \mathbf{u})$ until the evaders are close to the desired configuration.

Fig. 2.4 also shows how the adaptation law adjusts the parameters to ensure stability. Initially, all the parameters are equal; however, they evolve differently because each evader is in a specific situation with respect to its desired position and the herders. At steady-state, the estimated parameters are equal to the real ones.

2.6.2 Effects of the E-DKF in the herding

The first series of simulations has validated the success of Implicit Control and the adaptation law. Next, we show how the herding works when perfect state and input knowledge is replaced by the distributed estimator. The communication and sensing thresholds are in Table 2.2. We set $\mathbf{Q} = 0.02\mathbf{I}_{2m+2n}$ and $\mathbf{R}_i = 0.07\mathbf{I}_{2m+2n}\ \forall i$ to simulate a scenario with bad sensing capabilities, i.e., $\mathbf{R}_i \succ \mathbf{Q}$ to illustrate how incorporating Implicit Control in the prediction stage is useful. Finally, to be realistic in the use of the communication bandwidth, messages are exchanged every 100ms, i.e., the communication process runs 10 times slower than the control. Fig. 2.5 depicts the performance of the herders against the same herding scenario of Fig. 2.3. The results are almost identical, with herders and evaders following similar trajectories as if there was perfect feedback. The reason for that is answered in the second row of Fig. 2.5. The Root Mean Square Error (RMSE) between the estimates and real positions of the entities rapidly decreases to the noise level. We remark that both axis of the panels are in logarithmic scale. Thus, before the first 400ms (4 communication rounds) the estimator has converged, which justifies the assumption of perfect knowledge in the control and adaptation. Despite the different frequencies between communication and control, the herders are successful because they can predict the movement of the evaders and the other herders.

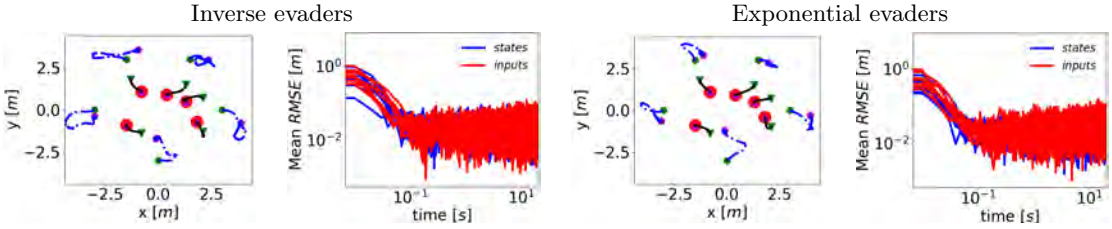


Figure 2.5: Simulation results of the herding of 5 evaders by 5 herders using Implicit Control with adaptation law and the distributed estimator. The first row shows the trajectories followed by the herders (same symbols of Table 2.1). The second row depicts the RMSE in the estimation of evaders' (blue) and herders' (red) position, averaged over the five herders.

In this vein, it is interesting to analyze how increasing the noise covariance affects the performance. To do so, we have run the same simulations in Fig. 2.5 but changing the measurement covariance matrix, checking to what extent the noise can increase without affecting the performance. The results are in Table 2.3. Implicit Control is robust against measurement noise, achieving good performance with noises up to 0.3m^2. We underline that this implies measurement errors in the order of 0.5m, which is significant considering

that the herding takes place in a 5×5m square (i.e., a 10% of the side size of the arena). It is also observed that the tolerance to noise depends on the evaders' dynamics. On the other hand, we do not consider changes in $\mathbf{Q}$ because the movement of herders and evaders is deterministic, so $\mathbf{Q}$ just models the noise associated with the difference between real and estimated parameters during the adaptation.

Table 2.3: Impact of the measurement noise in the herding.

$\mathbf{R}$ [m^2]	Steady-state error $\|\|\mathbf{x} - \mathbf{x}^*\|\|$ [m]	
	Inverse Model, 5vs5	Exponential Model, 5vs5
$0.07\mathbf{I}_{20}$	0.02	0.01
$0.15\mathbf{I}_{20}$	0.06	0.06
$0.30\mathbf{I}_{20}$	1.34	0.12

2.6.3 Complete precise herding

The flexibility and generality of the solution can be extended to heterogeneous groups and time-varying references, resulting in a more realistic herding. In this example we evaluate the inclusion of the caging stage as considered in Algorithm 1. Fig. 2.6 shows how three herders herd a group of three evaders using the complete herding solution in Algorithm 1. The red evader is Exponential while the purple evaders are Inverse. The CBFs have been tuned as follows, choosing

$$h(\Delta_i) = ||\Delta_i|| + T\frac{\Delta_i \delta \mathbf{u}_i^{nom}}{||\Delta_i||} - \varphi \tag{2.47}$$

and

$$\kappa(h(\Delta_i)) = k_{cbf} h(\Delta_i)^3, \tag{2.48}$$

where T is the sampling time, $\varphi = 3$ is a desired distance with the perimeter of $\mu_2 \mathbf{P}_{avg}$ ($\mu_1 = 7$ and $\mu_2 = 4$) and $k_{cbf} = 50$ is a gain which modulates how strong is the repulsion provoked by the proximity with $\mathbf{e}_i$. We recall that in the caging stage the dynamics of the herders are

$$\mathbf{u}_i = \mathbf{u}_i + T\delta \mathbf{u}_i^{cbf}. \tag{2.49}$$

Meanwhile, the desired herding configuration evolves with

$$\dot{x}_j^* = v_j^*, \qquad \dot{y}_j^* = 0.5 w_j^* \cos(w_j^* t + 2\pi/j),$$

where $\mathbf{w}^*$=[0.05, 0.1, 0.02]rad/s and $\mathbf{v}^*$=[0.05, 0.05, 0.05]m/s. Due to the distances involved in this herding instance, we set $d_m = d_c = 15$m.

Initially, the herders move to uniformly surround the evaders, positioning themselves on the perimeter of the desired ellipse. Once the herders are there, the precise herding begins, driving the evaders to their sinusoidal references. This yields to herders' trajectories surrounding and modulating the interaction forces with the evaders. With the evaders in their desired trajectories, the system reaches a steady-state behavior where the periodic movement of the evaders is shared by the herders.

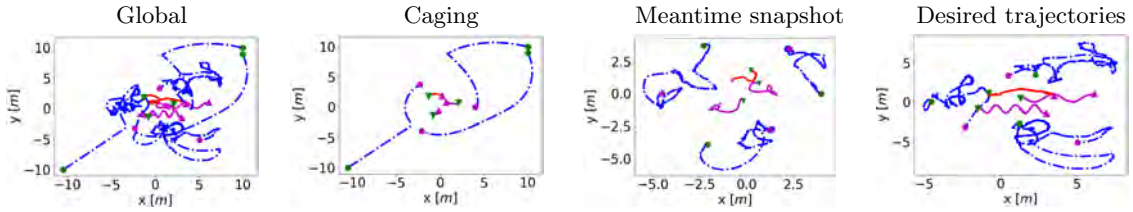


Figure 2.6: Three robotic herders herding three heterogeneous evaders. The magenta trajectories correspond to Inverse evaders while the red trajectory is an Exponential evader. The other symbols follow the convention in Table 2.1.

2.6.4 Controlling large herds with a few herders

To demonstrate the generality and flexibility of Implicit Control, we conduct some experiments where Implicit Control is used to control a large number of evaders with a few herders. To do this, the control goal is to steer the centroid of the herd towards a particular location. At each instant, the control algorithm first calculates the centroid of the herd and, then, it computes the input with Implicit Control, using the centroid as if it was a virtual evader with its corresponding evader's dynamics. The computation of the centroid induces a little noise in the dynamics of the virtual evader, but it is negligible because, point-wise, the centroid's dynamics is still that of the virtual evader. Besides, this discrepancy can be absorbed by the adaptation law.

In Fig. 2.7 it is shown how 5 herders can control 50 evaders. There are 10 inputs and 2 states to control (the x and y position of the centroid). To maintain cohesion of the herd, but also to avoid collisions among the evaders, we have added very weak repulsive and coalition forces among evaders

$$\dot{\mathbf{x}}_j = f_j^{inv/exp}(\mathbf{x}, \mathbf{u}) + \vartheta \sum_{j'=1}^{m} \mathbf{d}_{jj'} \left(\frac{1}{||\mathbf{d}_{jj'}||^3} - ||\mathbf{d}_{jj'}||^2 \right) \tag{2.50}$$

with $\mathbf{d}_{jj'} = \mathbf{x}_j - \mathbf{x}_{j'}$ and $\vartheta = 2 \times 10^{-4}$, i.e., 4 orders of magnitude smaller than the dynamics in Eqs. (2.2) and (2.3).

The first row of Fig. 2.7 shows how the five herders can steer the whole herd towards a certain region. As a curiosity, since the number of available control inputs is much greater than the number of states to control, there is a "leftover" herder in steady-state (right pannel): there are two herders which share a similar position. More interestingly, the second row of Fig. 2.7 shows how the herders can split the herd in two sub-herds and steer each of them, simultaneously, towards individually assigned locations. This confirms that, with our herding, it is possible to consider each evader to virtually represent a herd (see, e.g., [Pierson and Schwager, 2018] for a similar consideration).

2.6.5 Dynamic assignment for massive herding

In previous examples, herders under Implicit Control can steer as many evaders as herders unless we use the centroid of the herd as a virtual evader. Nevertheless, it is not required,

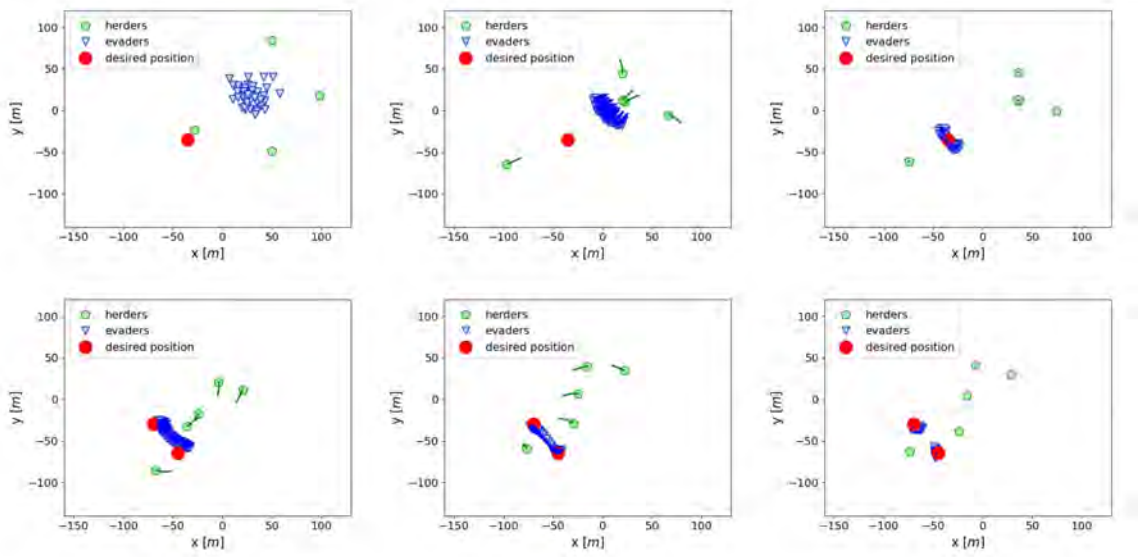


Figure 2.7: 5 herders herding 50 Inverse evaders. In the first 50 seconds, the herd is steered towards the desired location (first row). Afterwards, the herd is split in two sub-herds and driven towards two different desired locations simultaneously (second row).

a priori, that the evaders targeted by each robot are the same for all time. In this section we design a dynamic assignment strategy to select, at each instant, the most convenient evaders to be directly controlled. Therefore, the constraint in the number of controlled evaders is always preserved, whereas the other evaders are steered indirectly.

The first step is to compute the convex hull of the herd (Fig. 2.8a). Let $\mathcal{X} = \{\mathbf{x}_j\}_0^m$ be the set with all the evaders' positions. The convex hull of $\mathcal{X}$ is

$$\mathcal{CH} = \left\{ \sum_{j=0}^{m} \lambda_j \mathbf{x}_j,\ \lambda_j \geq 0\ \forall j \text{ and } \sum_{j=0}^{m} \lambda_j = 1 \right\} \tag{2.51}$$

We denote $\mathbf{c}_j \in \mathcal{CH}$ the evaders that form the convex hull, i.e., $\lambda_j \neq 0$, and $\mathcal{C} = \{\mathbf{c}_j | \lambda_j \neq 0\}_0^m$. Fig. 2.8b shows the convex hull of an illustrative herd configuration. The convex hull is important because it delimits the herd, so the smaller the area of $\mathcal{C}$, the more compact is the herd formation. It also characterizes whether an evader has escaped from the herders or not. In particular, we use the Quickhull Algorithm [Barber et al., 1996] for a fast computation of $\mathcal{CH}$ and $\mathcal{C}$. With $\mathcal{C}$, we are interested in finding a uniform partition of these evaders. A uniform partition allows to evenly cover all the herd and helps in avoiding escapes. Inspired by Voronoi diagrams [Aurenhammer, 1991], we propose a K-Means clustering [Likas et al., 2003] over the $\mathcal{C}$. Formally, the objective is to find clusters $\mathcal{K} = \{\mathcal{K}_1, \ldots, \mathcal{K}_p | \mathcal{K}_1 \cup \ldots \cup \mathcal{K}_p = \mathcal{C}\}$ such that

$$\mathcal{K} = \arg\min_{\hat{\mathcal{K}}} \sum_{k=0}^{p} \sum_{\mathbf{k}_l \in \hat{\mathcal{K}}_k} ||\mathbf{k}_l - \mu_k||^2. \tag{2.52}$$

Here, $\mathcal{K}_k = \{\mathbf{k}_l\}_0^{p_k}$ is the set of convex-hull evaders that belong to cluster k, μ_k is the centroid of cluster $\mathcal{K}_k$, $p_k > 0$ is the number of evaders in cluster k and $p = \min(|\mathcal{C}|, n)$. The latter means that the strategy selects as many evaders as herders, unless $|\mathcal{C}| < n$, where we select all the evaders. Note that the K-Means algorithm [Dempster et al., 1977] gives the Voronoi centers μ_k $\forall k$. Fig. 2.8c shows the resulting clusters.

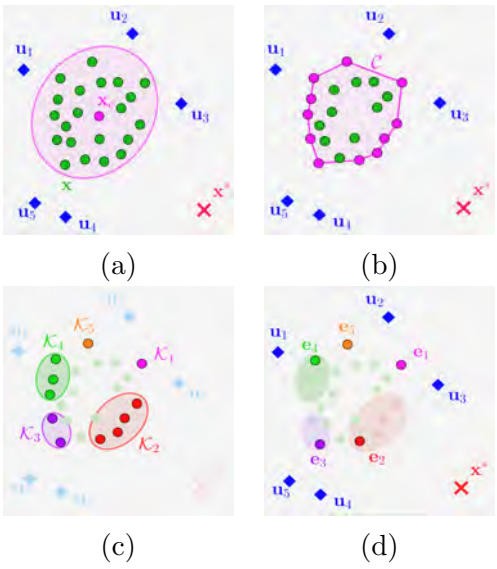


Figure 2.8: Visualization of the dynamic assignment strategy: (a) configuration at instant t, (b) convex hull of the herd, (c) computed clusters, (d) final assignment.

The final step is to select the evaders to be directly herd. The centers of the clusters can not be chosen because their dynamics are not the ones modeled in Eq. (2.1). For each cluster, the furthest evader to the cluster's center is computed. This promotes the compactness of the herd. Formally, we solve

$$\mathbf{e}_k = \underset{\mathbf{k}_l \in \mathcal{K}_k}{\arg\max} \; ||\mathbf{k}_l - \mu_k||, \; \forall k. \tag{2.53}$$

After solving (2.53), new set of evaders $\{\mathbf{e}_1, \ldots, \mathbf{e}_p\}$ is obtained. Fig. 2.8d shows the assigned evaders in an illustrative herd configuration. These evaders are the ones that are directly controlled by the herders. In other words, the controlled state employed in Eq. (2.11) is $\mathbf{x} = \begin{bmatrix} \mathbf{e}_1^\top & \ldots & \mathbf{e}_p^\top \end{bmatrix}^\top$.

The overall dynamic assignment strategy along with Implicit Control is summarized in Algorithm 2. Roughly speaking, at each instant the algorithm ensures that the convex hull of the herd converges towards the desired reference. Therefore, since Implicit Control ensures global asymptotic stability, the convex hull converges to a region near the desired reference. In particular, the convex hull is steered to be as close as possible to $\mathbf{x}^*$. However, the repulsive forces among evaders prevents to obtain $\mathbf{e}_k = \mathbf{x}^*$ for all k because there are evaders within the herd that repel them. Since at each instant the furthest evaders are selected, evaders do not escape and the center of the herd converges asymptotically to $\mathbf{x}^*$.

Algorithm 2 Implicit Control and Dynamic Assignment Strategy at time t

$\mathcal{X}(t), \mathbf{u}(t) \leftarrow$ obtain herders' and evaders' positions at instant t
$\mathcal{C}(t) \leftarrow \text{ConvexHull}(\mathcal{X}(t))$
if $|\mathcal{C}(t)| < n$ **then**
  $\mathcal{K}(t) \leftarrow \text{KMeans}(|\mathcal{C}(t)|, \mathcal{C}(t))$
else
  $\mathcal{K}(t) \leftarrow \text{KMeans}(n, \mathcal{C}(t))$
end if
for $\mathcal{K}_k(t)$ in $\mathcal{K}(t)$ **do**
  $\mathbf{e}_k(t) \leftarrow \arg\max_{\mathbf{k}_l(t) \in \mathcal{K}_k(t)} ||\mathbf{k}_l(t) - \mu_k(t)||$
end for
$\mathbf{u}(t) \leftarrow \text{ImplicitControl}(\mathbf{e}(t), \mathbf{u}(t))$

To validate the proposal, we conduct different simulated experiments. The purpose is to verify that the herding is achieved for (i) different numbers of evaders and herders, where $m \gg n$; (ii) heterogeneous groups of evaders, where each evader may have different dynamics; (iii) both fixed and time-varying references.

The values of the parameters are: $\theta_j = 1.2$, $\gamma_j = 2 \times 10^{-4}$, $\beta_j = 0.5$, $\sigma_j = 2.0$, $d_{min} = 10$m, the sample time for the simulation is $T = 40$ms, and $v_{max} = 4$m/s for both evaders and herders. The sample time is chosen for the robots to have enough time for all the calculations, while the maximum speed and the other aforementioned parameters are chosen to obtain realistic results in terms of distances, velocities and times. Nonetheless, any combination of parameters is possible. We set $\mathbf{F} = 0.25\mathbf{I}_{2p}$ and $\mathbf{H} = 50\mathbf{I}_{2p}$ in (2.13) to ensure the stability of Implicit Control. We test three scenarios: (i) all evaders follow Inverse model dynamics, (ii) all evaders follow Exponential model dynamics and (iii) evaders follow either Inverse or Exponential model dynamics. When the experiments involve both models, we initialize the model of each evader randomly from a Bernoulli distribution. The evaders' positions are initialized from a normal distribution of mean $[20.0, 0.0]$m and covariance $\mathbf{I}_2$. Meanwhile, robots are deployed in an even distribution around the evaders, with center $[20.0, 0.0]$m and radius 70m. The desired tracking reference is $\mathbf{x}^* = [-35.0, -35.0]$m and $\dot{\mathbf{x}}^* = [0.0, 0.0]$m/s for $t \in [0.0, 160.0)$s and $\dot{\mathbf{x}}^* = [\nu, \omega A \cos(\omega t T)]$m/s for $t \in [160.0, 400.0]$s, with $\nu = -0.2$m/s, $A = 5$m/rad and $\omega = 0.1$rad/s.

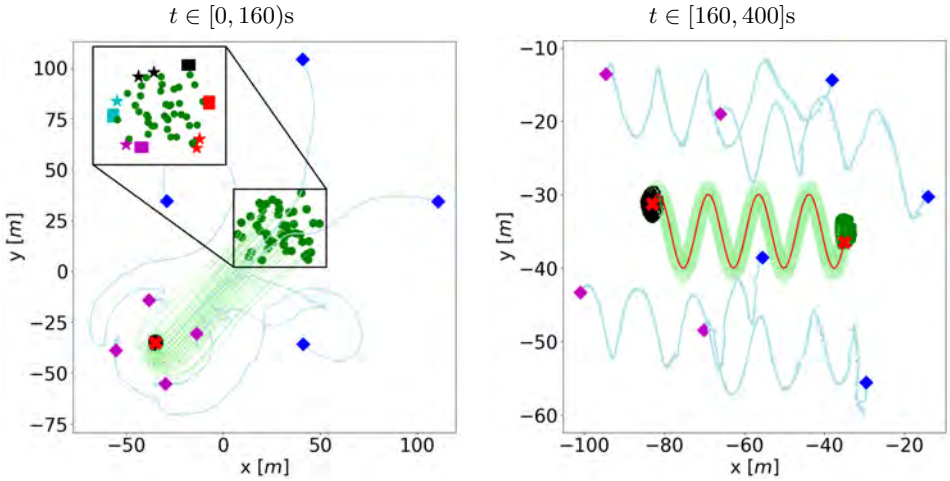


Figure 2.9: Illustrative example. Herding of 50 Inverse evaders by 4 robotic herders. The symbols are explained in Table 2.1. The left pannel also shows the results of the dynamic assignment strategy for the first instant: each color represents a cluster, the square marks represent the assigned evaders while the star marks represent the other evaders that belong to the convex hull.

Fig. 2.9 shows 4 robotic herders herding 50 Inverse model evaders. In Fig. 2.9a it is depicted the first part of the herding, where the four herders are successful in steering all the evaders towards the desired static reference. From the trajectories it is verified that none of the evaders escape and they all remain within the influence of the herders. The convex hull is pushed towards the desired reference. Once the evaders are close to the desired location, the herders move to make the herd more compact. The final configuration of the herd is the compact black circle in Fig. 2.9a, with the centroid in $\mathbf{x}^*$ and with the herders in a symmetric position to preserve the convex hull compactness around $\mathbf{x}^*$. After 160s the time-varying herding begins, and in Fig. 2.9b it is demonstrated that the herders are capable of steering the herd throughout a complex tracking reference. The compactness is preserved, and the centroid of the herd accurately follows $\mathbf{x}^*$. To complement Fig. 2.9, Fig. 2.10 depicts the same experiment but with either Inverse, Exponential or a mix of

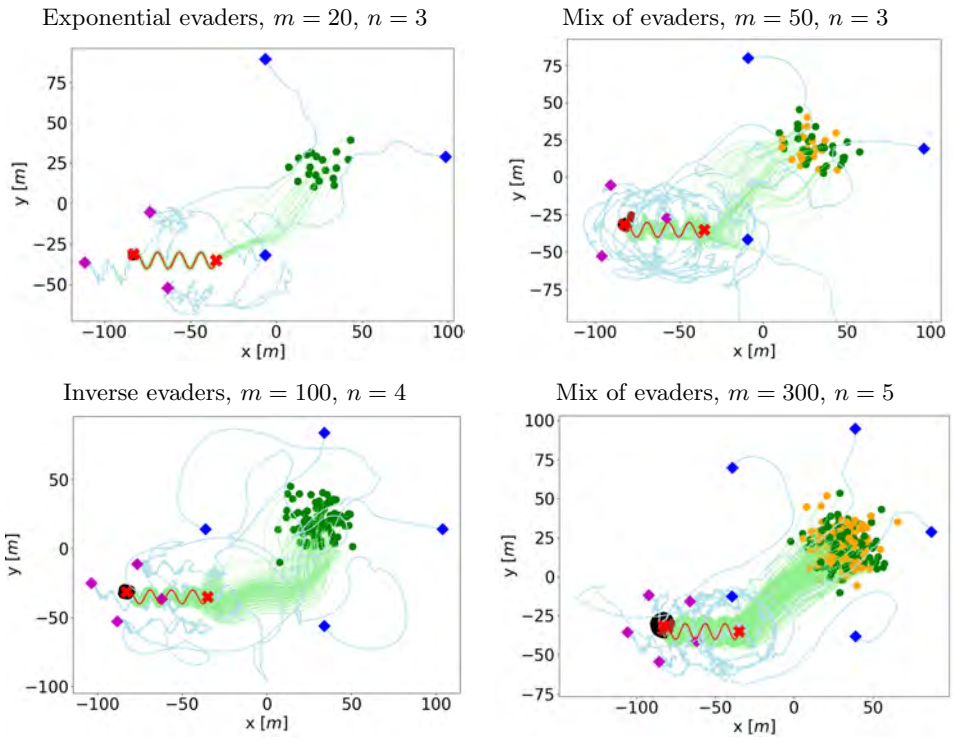


Figure 2.10: Illustrative examples. The pannels depict the overall trajectories from $t = 0$s to $t = 400$s. In the mixed configurations, we use green and black for the initial and final configuration of the Inverse evaders whereas orange and brown for the initial and final configuration of the Exponential evaders. The other symbols follow Table 2.1.

Inverse and Exponential evaders in different configurations. The results are as accurate as in the previous example.

As a last remark, the computational time of the proposal does not vary significantly when the number of evaders increases. The only part that depends on the size of the herd is the computation of the convex hull. However, the Quickhull Algorithm [Barber et al., 1996] has an average complexity $O(n \log n)$, so it can compute the convex hull of up to thousands of evaders in a few milliseconds. After that, the number of evaders involved in the computations is bounded by the number of herders, and since we only use a few herders, the proposal is cheap to compute. This is why $T = 40$ms is enough for computing Algorithm 2.

2.7 Real experiments

In this section we extend the simulations to the real framework provided by the Robotarium [Wilson et al., 2020]. To do so, some robots play the role of herders while the others act as evaders, following the dynamics in Section 2.2. The robots are GRITSBot X moving on a 3.2m x 2m area, coordinated by a central server which receives odometry and sends velocity commands at an approximately delay of 0.033s. Thus, a low level controller is used to translate the control output into velocity commands, with $v_{max} = 0.2$m/s. We use the

experiments to validate Implicit Control, the adaptation law and the caging stage. Finally, we adjust some parameters to fit the conditions of the experiment: $T = 0.033$s, $\theta = 0.02$, $\hat{\theta}(0) = 0.015$ (Inverse), $\theta = 0.05$, $\hat{\theta}(0) = 0.04$ (Exponential), $\sigma = 1.2$.

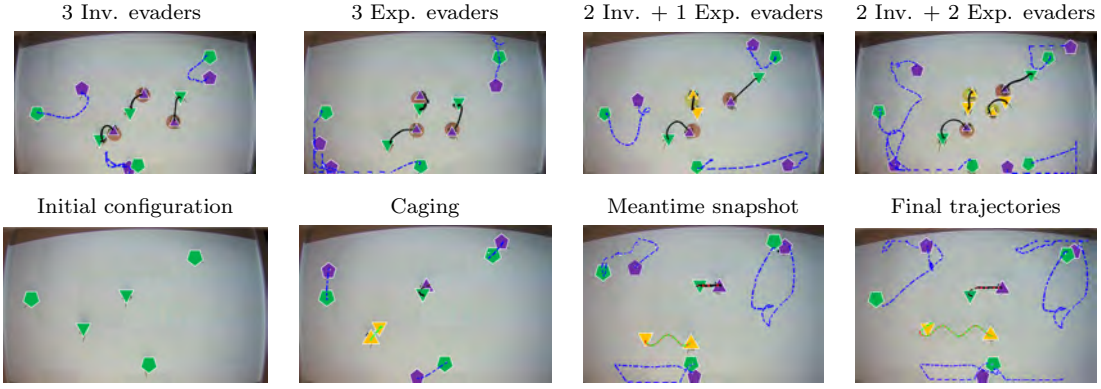


Figure 2.11: Results of the herding using Robotarium [Wilson et al., 2020]. The symbols are explained in Table 2.1. When evaders' dynamics are mixed, initial, final and desired position of Exponential evaders are yellow. The second row shows snapshots of three herders herding two heterogeneous evaders in a time-varying path.

The experiment in Fig. 2.1 exhibits a similar behavior to the simulations in Section 2.6. The evaders try to evade the herders, going in the direction of lower density of herders. To tackle this, the closest herders surround the evaders to align with the other herders, which move away to modulate the interaction forces. These conclusions are reaffirmed by the experiments in Fig. 2.11, where different combinations of evaders, in their number and dynamics, are tested. Additionally, the herders successfully herd heterogeneous groups of evaders. The last experiment consists in the herding of an Inverse evader and an Exponential evader by three herders, where the caging stage has been included. Initially, the herders approach the evaders, surrounding them to avoid escapes. Then they steer the evaders towards the desired trajectories (in red). Despite the space limitations and the complex nonlinear repulsive dynamics, the evaders successfully follow the references.

2.8 Conclusions

This chapter has presented a novel control strategy for input-nonaffine dynamical systems in multi-robot control problems. This strategy, based on numerical analysis theory and coined as Implicit Control, finds suitable inputs even when, due to the complex nonlinearities, the control law is given by a set of implicit equations. Implicit Control develops a continuous-time expansion of the system, and comes with formal proofs of convergence. In addition, it is flexible in the number of evaders and general with respect to their motion model. The theoretical concepts of Implicit Control allow to derive an adaptation law. It preserves the stability properties of the control while naturally integrates in the control architecture. As motivating example, we have exemplified the design on an Implicit Control law in a multi-robot herding problem. To complete the proposal in terms of multi-robot herding, we have developed a novel solution for the caging problem. The herders can depart from any

arbitrarily far region because they are able to approach and surround the evaders. This is done in conjunction with a novel version of an Extended Distributed Kalman Filter, which has solved the problem of complete perfect measurements. Furthermore, we have developed a new dynamic assignment strategy that selects the evaders to be targeted by the robotic herders, such that herd coalition is achieved. The flexibility, robustness, and generality of the solution has been demonstrated in numerous simulations and experiments with real robots.

Implicit Control paves the way for the distributed control of general input-nonaffine systems, which is one of the challenges targeted in this thesis. Nonetheless, the original Implicit Control formulation is restricted to centralized systems. Despite the fact that this chapter already proposes a distributed estimator to overcome this limitation, the results of the chapter inspire novel research lines of potential interest, including the full distribution of the control computation, e.g., using distributed optimization and consensus techniques. These ideas are explored in the next chapter.

Chapter 3

Distributed Optimization, Consensus and Dissensus

Nonlinear control techniques such as Implicit Control are tailored for centralized systems, where robots have access to all the information of the environment. This limits its applicability to distributed multi-robot systems, where information such as the location of all the robots is not readily available at the local robot level. In this chapter, we elaborate on distributed optimization, consensus and dissensus tools that allow to recover the global quantities of interest necessary to apply nonlinear control techniques. More specifically, we cover the following three topics:

- **Distributed optimization** refers to the problem of finding the global optimum of an optimization problem where the cost function, the constraints or the available information is distributed over a network [Nedić et al., 2018]. Of particular interest is the case of unconstrained consensus optimization [Nedić and Liu, 2018], where the global cost function is the sum of local cost functions. Practical instances of such a setting are federated learning [Nedic, 2020], networked games [Parise and Ozdaglar, 2019] or multi-robot control [Shorinwa et al., 2024]. In all these applications, it is of key importance to reconstruct the global optimum as fast as possible to reduce the number of gradient computations, the communication efforts and adapt to evolving environments. To cope with this, we propose a novel accelerated distributed optimization algorithm that exploits momentum to outperform existing first order approaches in convergence speed, while leveraging the robustness of the alternating direction method of multipliers (ADMM) and the computational simplicity of gradient tracking (GT).
- **The problem of consensus** in control theory [Garin and Schenato, 2010] consists of finding a protocol such that a set of nodes in a network agree on the value of a certain quantity of interest. In particular, the discrete time dynamic average consensus [Kia et al., 2019] is interesting because the tracked signals usually evolve with time, and protocols are implemented in computing units that work in discrete steps. Despite the existing literature, current solutions still suffer from some of the following issues:

(i) integral and difference input terms are not robust against input and initialization noise, or changes of network size; (ii) trade-off between convergence speed and steady-state accuracy; (iii) absence of robustness against packet losses and communication delays. These aspects affect the applicability of consensus in real-world scenarios such as smart grids [Franceschelli et al., 2020] or sensor networks [Sebastián et al., 2021b]. To overcome these issues, we propose a novel algorithm that combines a multi-stage consensus protocol and a second order diffusion method, along with its asynchronous and randomized version.

- **Distributed dissensus** is a less explored problem compared to consensus, since it studies how to enforce disagreement among agents in a network. A particularly relevant example in robotic systems is that of robot navigation in dynamic environments, where robots move from one location to another while avoiding collisions with obstacles that are in motion around the environment [Choset et al., 2005]. Existing solutions assume the degree of cooperation between the robot and an agent [Van Den Berg et al., 2011,Alonso-Mora et al., 2018,Han et al., 2020,Boldrer et al., 2023], assume the robot can communicate with other robots [Bajcsy et al., 2019, Li et al., 2020b, Patwardhan et al., 2022,Zhang et al., 2023], or assume the computational capabilities of the robot are sufficient to do inference on learning-based policies or long-term horizon planning [Long et al., 2017, Everett et al., 2021, Brito et al., 2019, Poddar et al., 2023].To cope with all these issues, we propose a novel adaptation law based on nonlinear opinion dynamics [Bizyaeva et al., 2022], a sophisticated dissensus technique that guarantees fast and flexible decision-making.

In the following three sections we dive into each of these problems separately, including a description of the related work, the proposed method, and simulations and experiments that validate them.

3.1 Distributed optimization: accelerated alternating direction method of multipliers gradient tracking

3.1.1 Related work

The evolution of distributed optimization algorithms can be divided in three stages. The first approaches departed from subgradient [Sundhar Ram et al., 2010], gradient [Jakovetić et al., 2014] and proximal methods [Bertsekas, 2011] to derive distributed versions that achieve convergence to the optimum if and only if the step size diminishes with time. This property poses a fundamental trade-off between speed and accuracy because fixed step sizes lead to convergence within a given distance from the optimum. To overcome the speed-accuracy trade-off, several works capitalized from different points of view the properties of dynamic average consensus [Franceschelli and Gasparri, 2019,Sebastián et al., 2023c] and splitting methods to develop GT [Shi et al., 2015,Qu and Li, 2017,Nedic et al., 2017, Xin and Khan, 2019], and Alternating Direction Method of Multipliers (ADMM)

or primal/dual methods for distributed optimization [Wei and Ozdaglar, 2013, Bastianello et al., 2020, Notarnicola et al., 2019]. In both cases, linear convergence is achieved for constant step sizes; however, these approaches suffer from their respective caveats. GT is simple to compute but is inherently non-robust against non-ideal affections such as initialization errors, gradient noise or communication disturbances [Bin et al., 2022]. ADMM is robust against those undesirable effects but, as a proximal method, it has a non-negligible computational cost at each iteration [Bastianello et al., 2020]. In both cases, due to the importance of convergence speed, different works have exploited acceleration methods to improve convergence rate such as momentum [Nguyen et al., 2023, Huang et al., 2024], Nesterov acceleration [Qu and Li, 2019], Anderson acceleration and adaptive preconditioning for proximal-based optimization such as ADMM and Douglas-Rachford Splitting [Fu et al., 2020, Adil et al., 2021] or smooth/non-smooth regularizers [Li et al., 2019b]. Many of the aforementioned methods work for time-varying and/or directed graphs, whereas here we focus on the static undirected case.

Recently, a distributed optimization protocol that combines ADMM and GT [Carnevale et al., 2023a, Carnevale et al., 2023b] was proposed as an alternative to combine the benefits of both approaches. The key idea is that GT can be reformulated in ADMM form, where the ADMM cost function is quadratic in the optimization variables. Therefore, it is possible to obtain a closed analytical solution that is cheap to compute and robust. Nevertheless, despite proving linear convergence for strongly convex cost functions, the algorithm needs time-scale decoupling between the optimum estimate and auxiliary dynamics, limiting the achievable convergence speed.

3.1.2 Preliminaries

A network of $\mathsf{N} > 0$ nodes cooperates to find the global optimum $\mathbf{x}^* \in \mathbb{R}^n$ of the consensus optimization problem

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{R}^n} \sum_{i=1}^{\mathsf{N}} f_i(\mathbf{x}), \tag{3.1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the $n-$dimensional decision variable. Each node $i \in \mathcal{V} = \{1, \dots, \mathsf{N}\}$ only has access to its local objective function $f_i(\bullet)$ and the local estimate of global optimum $\mathbf{x}_i^t$, with $t \in \mathbb{N}$ denoting the discrete time steps. Problem (3.1) is reformulated in distributed form by including the consensus constraint as follows

$$\mathbf{x}^* = \arg\min_{\mathbf{x}_i\in\mathbb{R}^n} \sum_{i=1}^{\mathsf{N}} f_i(\mathbf{x}_i) \quad \text{s.t. } \mathbf{x}_i = \mathbf{x}_j \quad \forall i, j \in \mathcal{V}. \tag{3.2}$$

We assume that $f_i(\bullet)$ is $\mathbf{c}$-strongly convex and the gradients $\nabla f_i(\mathbf{x})$ are L-Lipschitz continuous $\forall i \in \mathcal{V}$. This assumption guarantees that $\mathbf{x}^*$ is unique.

The network of nodes is described by an undirected connected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges of the graph, such that $(i, j) \in \mathcal{E}$ means that nodes i and j can communicate with each other. The set of neighbors of node i is $\mathcal{N}_i = \{j | (i, j) \in \mathcal{E}\}$. Note that $i \notin \mathcal{N}_i$. The degree of node i is $d_i = \mathsf{car}(\mathcal{N}_i)$ and $d = \sum_{i=1}^{\mathsf{N}} d_i$, where $\mathsf{car}(\bullet)$ denotes the cardinality of a set.

The goal of this section is to develop a distributed algorithm such that each local estimate $\mathbf{x}_i^t$ linearly converges to the global optimum $\mathbf{x}^*$, respecting the topology structure of $\mathcal{G}$.

3.1.3 Singularly perturbed systems

Some of the mathematical derivations exposed in this section are based on singularly perturbed systems theory [Carnevale et al., 2024]. Singularly perturbed systems are dynamical systems composed by two interconnected dynamics, each of them evolving at different timescales. Typically, the slow dynamics corresponds to the desired states to be controlled (e.g., the estimates $\mathbf{x}_i^t$) while the fast dynamics corresponds to auxiliary variables. To ensure stability, the speed of the slow system is constrained by the speed of the fast dynamics. For clarity, we reproduce a theorem that establishes that, for a sufficiently slow dynamics, the slow system (3.3a) is decoupled from the fast system (3.3b) and they are both globally exponentially stable. We will use this fact to prove convergence properties of our algorithm.

Theorem 4 (Theorem II.3, [Carnevale et al., 2023b])**.** *Let*

$$\mathbf{p}^{t+1} = \mathbf{p}^t + \gamma f(\mathbf{p}^t, \mathbf{q}^t, t) \tag{3.3a}$$
$$\mathbf{q}^{t+1} = g(\mathbf{q}^t, \mathbf{p}^t, t), \tag{3.3b}$$

with $\mathbf{p} \in \mathbb{R}^n$, $\mathbf{q} \in \mathbb{R}^m$, $\gamma > 0$ *and* $f(\bullet), g(\bullet)$ *Lipschitz continuous uniformly over* t*. Let assume that there exists a function* $\mathbf{q}_{\mathsf{eq}}(\mathbf{p})$ *that is Lipschitz continuous such that*

$$\mathbf{0}_n = \gamma f(\mathbf{0}_n, \mathbf{q}_{\mathsf{eq}}(\mathbf{0}_n), t) \tag{3.4a}$$
$$\mathbf{q}_{\mathsf{eq}}(\mathbf{p}) = g(\mathbf{q}_{\mathsf{eq}}(\mathbf{p}), \mathbf{p}, t). \tag{3.4b}$$

We call reduced system the system given by

$$\mathbf{p}^{t+1} = \mathbf{p}^t + \gamma f(\mathbf{p}^t, \mathbf{q}_{\mathsf{eq}}(\mathbf{p}^t), t) \tag{3.5}$$

and boundary system layer with $\tilde{\mathbf{q}} \in \mathbb{R}^m$ *the system*

$$\tilde{\mathbf{q}}^{t+1} = g(\tilde{\mathbf{q}}^t + \mathbf{q}_{\mathsf{eq}}(\mathbf{p}^t), \mathbf{p}^t, t) - \mathbf{q}_{\mathsf{eq}}(\mathbf{p}^t). \tag{3.6}$$

Assume that there exist continuous function $u(\bullet)$ *and gain* $\bar{\gamma}_1 > 0$ *such that, for any* $\gamma \in (0, \bar{\gamma}_1)$*, there exist* $b_1, b_2, b_3, b_4 > 0$ *and values* $\tilde{\mathbf{q}}, \tilde{\mathbf{q}}_1, \tilde{\mathbf{q}}_2 \in \mathbb{R}^m$ *such that*

$$b_1||\tilde{\mathbf{q}}||^2 \leq u(\tilde{\mathbf{q}}, t) \leq b_2||\tilde{\mathbf{q}}||^2 \tag{3.7a}$$
$$u(\tilde{\mathbf{q}}^{t+1}, t+1) - u(\tilde{\mathbf{q}}^t, t) \leq -b_3||\tilde{\mathbf{q}}||^2 \tag{3.7b}$$
$$|u(\tilde{\mathbf{q}}_1, t) - u(\tilde{\mathbf{q}}_2, t)| \leq b_4||\tilde{\mathbf{q}}_1 - \tilde{\mathbf{q}}_2||(||\tilde{\mathbf{q}}_1|| + ||\tilde{\mathbf{q}}_2||). \tag{3.7c}$$

Symbol $||\bullet||$ *denotes the L2-norm. Also, assume that there exists a continuous function* $w(\bullet)$*, and gain* $\bar{\gamma}_2 > 0$ *such that, for any* $\gamma \in (0, \bar{\gamma}_2)$*, there exist* $c_1, c_2, c_3, c_4 > 0$ *and values* $\mathbf{p}, \mathbf{p}_1, \mathbf{p}_2 \in \mathbb{R}^n$ *such that*

$$c_1||\mathbf{p}||^2 \leq w(\mathbf{p}, t) \leq c_2||\mathbf{p}||^2 \tag{3.8a}$$
$$w(\mathbf{p}^{t+1}, t+1) - w(\mathbf{p}^t, t) \leq -c_3||\mathbf{p}||^2 \tag{3.8b}$$
$$|w(\mathbf{p}_1, t) - w(\mathbf{p}_2, t)| \leq c_4||\mathbf{p}_1 - \mathbf{p}_2||(||\mathbf{p}_1|| + ||\mathbf{p}_2||). \tag{3.8c}$$

Then, there exists $\bar{\gamma} \in (0, \min(\bar{\gamma}_1, \bar{\gamma}_2))$ *and gains* $\kappa_1, \kappa_2 > 0$ *such that, for any* $\gamma \in (0, \bar{\gamma})$ *and* $(\mathbf{p}^0, \mathbf{q}^0)$*, it holds that*

$$\left\| \begin{pmatrix} \mathbf{p}^t - \mathbf{p}^* \\ \mathbf{q}^t - \mathbf{q}_{\mathsf{eq}}(\mathbf{p}^t) \end{pmatrix} \right\| \leq \kappa_1 \left\| \begin{pmatrix} \mathbf{p}^0 \\ \mathbf{q}^0 - \mathbf{q}_{\mathsf{eq}}(\mathbf{p}^0) \end{pmatrix} \right\| e^{-\kappa_2 t}, \tag{3.9}$$

where $\mathbf{p}^* \in \mathbb{R}^n$ *is the desired equilibrium of* $\mathbf{p}$*.*

Conditions (3.7a), (3.8a), (3.7b), (3.8b) come from standard Lyapunov stability arguments, whereas (3.7c) and (3.8c) bound the changes in the Lyapunov functions $w(\mathbf{p}, t)$ and $u(\tilde{\mathbf{q}}, t)$ to apply Lyapunov exponential stability arguments.

3.1.4 ADMM gradient tracking

The authors of [Carnevale et al., 2023b] propose a first order distributed optimization algorithm that combines ADMM and GT to obtain a robust linearly convergent algorithm. Algorithm 3 presents such scheme. Each node i updates four variables. First, $\mathbf{x}_i^t$ is the

Algorithm 3 Original [Carnevale et al., 2023b] ADMM-GT at node i

State of agent: $\mathbf{x}_i^0 \in \mathbb{R}^n$, $z_{ij}^0 \in \mathbb{R}^{2n} \quad \forall j \in \mathcal{N}_i$
Parameters: $\gamma > 0$, $\rho > 0$, $\alpha \in (0, 1)$
for $t = 1, \ldots$ **do**
  $\mathbf{y}_i^{t+1} = \frac{1}{1+\rho d_i}(\mathbf{x}_i^t + \begin{pmatrix} \mathbf{I}_n & \mathbf{0}_{n\times n} \end{pmatrix} \sum_{j\in\mathcal{N}_i} \mathbf{z}_{ij}^t)$
  $\mathbf{s}_i^{t+1} = \frac{1}{1+\rho d_i}(\nabla f_i(\mathbf{x}_i^t) + \begin{pmatrix} \mathbf{0}_{n\times n} & \mathbf{I}_n \end{pmatrix} \sum_{j\in\mathcal{N}_i} \mathbf{z}_{ij}^t)$
  $\mathbf{z}_{ij}^{t+1} = (1-\alpha)\mathbf{z}_{ij}^t - \alpha \left(\mathbf{z}_{ji}^t - 2\rho \begin{pmatrix} \mathbf{y}_i^{t+1} \\ \mathbf{s}_i^{t+1} \end{pmatrix} \right) \quad \forall j \in \mathcal{N}_i$
  $\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \gamma(\mathbf{y}_i^{t+1} - \mathbf{x}_i^t) - \gamma \mathbf{s}_i^{t+1}$
end for

variable that estimates the global optimum $\mathbf{x}^*$. The variable $\mathbf{y}_i^t$ reconstructs the average of the estimates over the nodes of the network $\frac{1}{\mathsf{N}} \sum_{i=1}^{\mathsf{N}} \mathbf{x}_i^t$ and aims at eliminating the consensus error. The variable $\mathbf{s}_i^t$ reconstructs the average of the gradients $\frac{1}{\mathsf{N}} \sum_{i=1}^{\mathsf{N}} \nabla f_i(\mathbf{x}_i^t)$ and aims at reducing the error $\mathbf{e}_i^t = ||\mathbf{x}_i^t - \mathbf{x}^*||$. Finally, $\mathbf{z}_{ij}^t \in \mathbb{R}^{2n}$ are auxiliary variables that enforce consensus over $\mathbf{y}_i^t$ and $\mathbf{s}_i^t$ [Bastianello et al., 2020].

One fundamental aspect of Algorithm 3 is that, by conveniently rewriting the dynamics of $\mathbf{x}^t = [(\mathbf{x}_1^t)^\top, \ldots, (\mathbf{x}_{\mathsf{N}}^t)^\top]^\top$ and $\mathbf{z}^t = [(\mathbf{z}_1^t)^\top, \ldots, (\mathbf{z}_{\mathsf{N}}^t)^\top]^\top$ (with $\mathbf{z}_i^t = [(\mathbf{z}_{i1}^t)^\top, \ldots, (\mathbf{z}_{id_i}^t)^\top]^\top$), it can be shown that the overall dynamics of the scheme have the structure of a time-varying singularly perturbed system [Carnevale et al., 2023a]. Therefore, relying on Theorem 4, for a sufficiently small γ, Algorithm 3 converges linearly to the global optimum of (3.1).

In the following section, we present an algorithm that, by adding momentum to the dynamics of $\mathbf{y}_i^t$, $\mathbf{s}_i^t$ and $\mathbf{z}_{ij}^t$, speeds up the convergence while preserving the computation and communication properties of Algorithm 3.

3.1.5 Accelerated ADMM gradient tracking

The proposed accelerated distributed optimization scheme is presented in Algorithm 4. Al-

Algorithm 4 A2DMM-GT at node i

1: **State of agent**: $\mathbf{x}_i^0 \in \mathbb{R}^n$, $\mathbf{z}_{ij}^0 \in \mathbb{R}^{2n} \forall j \in \mathcal{N}_i$, $\mathbf{z}_{ij}^{-1} \in \mathbb{R}^{2n} \forall j \in \mathcal{N}_i$, $\mathbf{y}_i^0 \in \mathbb{R}^n$, $\mathbf{s}_i^0 \in \mathbb{R}^n$
2: **Parameters**: $\gamma > 0$, $\rho > 0$, $\alpha \in (0,1)$, $\lambda \in (1,2)$, $\mu \in (1,2)$, $\epsilon \in (0,1)$
3: **for** $t = 1, \dots$ **do**
4: $\quad \bar{\mathbf{y}}_i^{t+1} = \frac{1}{1+\rho d_i}(\mathbf{x}_i^t + \begin{pmatrix}\mathbf{I}_n & \mathbf{0}_{n\times n}\end{pmatrix} \sum_{j\in\mathcal{N}_i} \mathbf{z}_{ij}^t)$
5: $\quad \bar{\mathbf{s}}_i^{t+1} = \frac{1}{1+\rho d_i}(\nabla f_i(\mathbf{x}_i^t) + \begin{pmatrix}\mathbf{0}_{n\times n} & \mathbf{I}_n\end{pmatrix} \sum_{j\in\mathcal{N}_i} \mathbf{z}_{ij}^t)$
6: $\quad \mathbf{y}_i^{t+1} = \mathbf{y}_i^t + \lambda(\bar{\mathbf{y}}_i^{t+1} - \mathbf{y}_i^t)$
7: $\quad \mathbf{s}_i^{t+1} = \mathbf{s}_i^t + \lambda(\bar{\mathbf{s}}_i^{t+1} - \mathbf{s}_i^t)$
8: $\quad \bar{\mathbf{z}}_{ij}^{t+1} = (1-\alpha)\mathbf{z}_{ij}^t - \alpha\left(\mathbf{z}_{ji}^t - 2\rho\begin{pmatrix}\mathbf{y}_i^{t+1}\\ \mathbf{s}_i^{t+1}\end{pmatrix}\right) \quad \forall j \in \mathcal{N}_i$
9: $\quad \mathbf{z}_{ij}^{t+1} = \mu(\mathbf{z}_{ij}^t + \epsilon(\bar{\mathbf{z}}_{ij}^{t+1} - \mathbf{z}_{ij}^t)) + (1-\mu)\mathbf{z}_{ij}^{t-1} \quad \forall j \in \mathcal{N}_i$
10: $\quad \mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \gamma(\mathbf{y}_i^{t+1} - \mathbf{x}_i^t) - \gamma\mathbf{s}_i^{t+1}$
11: **end for**

gorithm 4 adds momentum to the dynamics of $\mathbf{y}_i^t$ and $\mathbf{s}_i^t$ (lines 6 and 7) and the dynamics of $\mathbf{z}_{ij}^t$ (line 9). The former allows to write Algorithm 4 as a chain of two singularly perturbed systems. The outer singularly perturbed system interconnects the dynamics given by $\mathbf{y}_i^t$ and $\mathbf{s}_i^t$ with the dynamics given by $\mathbf{x}_i^t$ and $\mathbf{z}_i^t$. The inner singularly perturbed system interconnects the dynamics given by $\mathbf{z}_i^t$ with the dynamics given by $\mathbf{x}_i^t$. In terms of convergence speed, there are three time-scales, from the fastest to the slowest: (1) $\mathbf{y}_i^t, \bar{\mathbf{y}}_i^t, \mathbf{s}_i^t, \bar{\mathbf{s}}_i^t$; (2) $\mathbf{z}_i^t, \bar{\mathbf{z}}_i^t$; and (3) $\mathbf{x}_i^t$. This time-scale separation leads to a novel convergence analysis of the ADMM-GT algorithm, discovering a limitation in the maximum achievable convergence rate of the dynamics of $\mathbf{z}_i^t$. By adding momentum to $\mathbf{z}_i^t$ this bound is overcome, enhancing the overall convergence rate when compared to Algorithm 3. Additionally, by respecting the ADMM gradient tracking structure of Algorithm 3, Algorithm 4 preserves the robustness from ADMM proved in [Carnevale et al., 2023b] and the computational simplicity of gradient tracking methods (the momentum operations are inexpensive to compute). Comparing Algorithms 3 and 4, their communication cost is the same since only the $\mathbf{z}_{ij}^t$ are exchanged. In terms of memory, each node has to additionally store $\mathbf{z}_{ij}^{t-1}$, $\mathbf{y}_i^t$, $\mathbf{s}_i^t$.

To analyze the convergence properties of Algorithm 4, we define the following matrices:

$$\mathbf{A}_x = \mathsf{diag}\left(\left\{\begin{pmatrix}\mathbf{1}_{d_i,n}\\ \mathbf{0}_{d_i,n}\end{pmatrix}\right\}_{i=1}^{\mathsf{N}}\right), \mathbf{A}_z = \mathsf{diag}\left(\left\{\begin{pmatrix}\mathbf{0}_{d_i,n}\\ \mathbf{1}_{d_i,n}\end{pmatrix}\right\}_{i=1}^{\mathsf{N}}\right)$$
$$\mathbf{A} = \mathsf{diag}\left(\{\mathbf{1}_{d_i,2n}\}_{i=1}^{\mathsf{N}}\right), \mathbf{H} = \mathsf{diag}\left(\left\{\frac{\mathbf{I}_n}{1+\rho d_i}\right\}_{i=1}^{\mathsf{N}}\right).$$

We use $\mathsf{diag}(\{\mathbf{E}_i\}_{i=1}^{\mathsf{N}})$ for the block diagonal matrix whose $i-$th block element matrix is $\mathbf{E}_i$. Symbol $\mathbf{1}_{\mathsf{N},p} = \mathbf{1}_{\mathsf{N}} \otimes \mathbf{I}_p$ where $\otimes$ denotes the Kronecker product. Besides, let $\mathbf{P} \in \{0,1\}^{2nd\times 2nd}$ be a permutation matrix that exchanges auxiliary variable $\mathbf{z}_{ij}^t$ with

auxiliary variable $\mathbf{z}_{ji}^t$ and implements the pairwise message passing found in line 8 of Algorithm 4, $\mathbf{x}^t = [(\mathbf{x}_1^t)^\top, \dots, (\mathbf{x}_{\mathsf{N}}^t)^\top]^\top$, $\mathbf{z}^t = [(\mathbf{z}_1^t)^\top, \dots, (\mathbf{z}_{\mathsf{N}}^t)^\top]^\top$, $\mathbf{z}_i^t = [(\mathbf{z}_{i1}^t)^\top, \dots, (\mathbf{z}_{id_i}^t)^\top]^\top$, $\mathbf{y}^t = [(\mathbf{y}_1^t)^\top, \dots, (\mathbf{y}_{\mathsf{N}}^t)^\top]^\top$ and $\mathbf{s}^t = [(\mathbf{s}_1^t)^\top, \dots, (\mathbf{s}_{\mathsf{N}}^t)^\top]^\top$. Finally,

$$\mathbf{g}(\mathbf{x}^t) = [(\nabla f_1(\mathbf{x}_1^t))^\top, \dots, (\nabla f_{\mathsf{N}}(\mathbf{x}_{\mathsf{N}}^t))^\top]^\top \text{ and } \mathbf{v}(\mathbf{y}^t, \mathbf{s}^t) = [(\mathbf{y}_1^t)^\top, (\mathbf{s}_1^t)^\top, \dots, (\mathbf{y}_{\mathsf{N}}^t)^\top, (\mathbf{s}_{\mathsf{N}}^t)^\top]^\top.$$

Exploiting these definitions, Algorithm 4 for the whole network can be written as

$$\mathbf{y}^{t+1} = (1-\lambda)\mathbf{y}^t + \lambda \mathbf{H}(\mathbf{x}^t + \mathbf{A}_x^\top \mathbf{z}^t), \tag{3.10a}$$
$$\mathbf{s}^{t+1} = (1-\lambda)\mathbf{s}^t + \lambda \mathbf{H}(\mathbf{g}(\mathbf{x}^t) + \mathbf{A}_z^\top \mathbf{z}^t), \tag{3.10b}$$
$$\mathbf{z}^{t+1} = \mu \mathbf{F}\mathbf{z}^t + (1-\mu)\mathbf{z}^{t-1} + 2\mu\epsilon\alpha\rho\mathbf{P}\mathbf{A}\mathbf{v}(\mathbf{y}^{t+1}, \mathbf{s}^{t+1}), \tag{3.10c}$$
$$\mathbf{x}^{t+1} = \mathbf{x}^t + \gamma(\mathbf{y}^{t+1} - \mathbf{x}^t) - \gamma \mathbf{s}^{t+1}, \tag{3.10d}$$

where $\mathbf{F} = (1-\epsilon\alpha)\mathbf{I}_{2nd} - \epsilon\alpha\mathbf{P}$. The compact expression in (3.10) can be reformulated as an interconnection of two singularly perturbed systems. In Fig. 3.1 we represent the interconnections between the dynamics of $\mathbf{x}^t$, $\mathbf{z}^t$, $\mathbf{y}^t$ and $\mathbf{s}^t$. The inner system (green) connects the dynamics of $\mathbf{x}^t$ (slow) and the dynamics of $\mathbf{z}^t$ (fast). Indeed, if $\mu = \epsilon = 1$, the inner system corresponds to the singularly perturbed system studied in [Carnevale et al., 2023b]. On the other hand, the outer system is also a singularly perturbed system given by $\mathbf{r}^t = [(\mathbf{y}^t)^\top, (\mathbf{s}^t)^\top]^\top$, $\mathbf{w}^t = [(\mathbf{x}^t)^\top, (\mathbf{z}^t)^\top, (\mathbf{z}^{t-1})^\top]^\top$, $\mathbf{v}(\mathbf{r}^t) = \mathbf{v}(\mathbf{y}^t, \mathbf{s}^t)$, and

$$\mathbf{r}^{t+1} = (1-\lambda)\mathbf{r}^t + \lambda\Big(\mathbf{H}(\mathbf{x}^t + \mathbf{A}_x\mathbf{z}^t)\mathbf{H}(\mathbf{g}(\mathbf{x}^t) + \mathbf{A}_z\mathbf{z}^t)\Big) = (1-\lambda)\mathbf{r}^t + \lambda\mathbf{h}(\mathbf{w}^t), \tag{3.11}$$

$$\mathbf{w}^{t+1} = \begin{pmatrix} (1-\gamma)\mathbf{I}_{\mathsf{N}n} & \mathbf{0}_{\mathsf{N}n\times 2nd} & \mathbf{0}_{\mathsf{N}n\times 2nd} \\ \mathbf{0}_{2nd\times 2nd} & \mu\mathbf{F} & (1-\mu)\mathbf{I}_{2nd} \\ \mathbf{0}_{2nd\times 2nd} & \mathbf{I}_{2nd} & \mathbf{0}_{2nd\times 2nd} \end{pmatrix} \mathbf{w}^t + \begin{pmatrix} \gamma\begin{pmatrix}\mathbf{I}_{\mathsf{N}n} & -\mathbf{I}_{\mathsf{N}n}\end{pmatrix}\mathbf{r}^t \\ 2\mu\epsilon\alpha\rho\mathbf{P}\mathbf{A}\mathbf{v}(\mathbf{r}^t) \\ \mathbf{0}_{2nd} \end{pmatrix} = \mathbf{Q}\mathbf{w}^t + \mathbf{B}(\mathbf{r}^t). \tag{3.12}$$

Given an arbitrary constant $\mathbf{w}^t = \mathbf{w}$ for all t, let function $\mathbf{r}_{\mathsf{eq}}(\mathbf{w})$ from (3.11) be

$$\mathbf{r}_{\mathsf{eq}}(\mathbf{w}) = (1-\lambda)\mathbf{r}_{\mathsf{eq}}(\mathbf{w}) + \lambda\mathbf{h}(\mathbf{w}) \Rightarrow \mathbf{r}_{\mathsf{eq}}(\mathbf{w}) = \mathbf{h}(\mathbf{w}). \tag{3.13}$$

Then, we define the error coordinates $\tilde{\mathbf{r}}^t = \mathbf{r}^t - \mathbf{r}_{\mathsf{eq}}(\mathbf{w})$. Regarding the slow system dynamics (3.12), we define the desired equilibrium $\mathbf{w}^*$ as

$$\mathbf{w}^* = \begin{pmatrix} \mathbf{x}^* \\ 2(\mathbf{I}_{2nd} + \mathbf{P})^{-1}\rho\mathbf{P}\mathbf{A}\mathbf{v}(\mathbf{r}_{\mathsf{eq}}(\mathbf{w}^t)) \\ 2(\mathbf{I}_{2nd} + \mathbf{P})^{-1}\rho\mathbf{P}\mathbf{A}\mathbf{v}(\mathbf{r}_{\mathsf{eq}}(\mathbf{w}^t)) \end{pmatrix}, \tag{3.14}$$

where we have exploited the fact that $\mathbf{z}^*$ can be arbitrarily chosen, making it coincide with the value of $\mathbf{z}^t$ when $\mathbf{z}^t = \mathbf{z}^{t-1}$, which is the steady-state value of $\mathbf{z}^t$ in Eq. (3.10c). Finally, we define the error dynamics $\tilde{\mathbf{w}}^t = \mathbf{w}^t - \mathbf{w}^*$.

We aim at exploiting Theorem 4 to prove the convergence properties of Algorithm 4. To do so, we first study the convergence properties of the boundary system layer error dynamics $\tilde{\mathbf{r}}^{t+1}$ and the reduced system error dynamics $\tilde{\mathbf{w}}^{t+1}$ by drawing their equivalence with the fast (3.3b) and slow (3.3a) dynamics respectively, proving that (3.7a)-(3.7c) and (3.8a)-(3.8c) hold.

In Lemmas 1 and 2 we prove that $\tilde{\mathbf{r}}^{t+1}$ can be written in the form of Eq. (3.3b) and $\tilde{\mathbf{w}}^{t+1}$ in the form of Eq. (3.3a). After that, Proposition 3 resorts to Theorem 4 to prove the global exponential stability of the dynamics of $\tilde{\mathbf{r}}^{t+1}$ and $\tilde{\mathbf{w}}^{t+1}$ and, consequently, the global exponential stability of Algorithm 4. Finally, we use Lemma 2 to demonstrate that Algorithm 4 leads to acceleration with respect to Algorithm 3, formalized through Theorem 5, where we substitute the slow state from $\mathbf{p}$ to $\tilde{\mathbf{w}}$ and the fast state from $\mathbf{q}$ to $\tilde{\mathbf{r}}$.

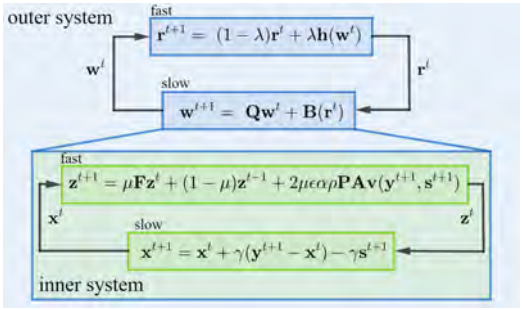


Figure 3.1: Representation of Algorithm 4 as an interconnection of two singularly perturbed systems. The outer singularly perturbed system connects the dynamics of the consensus variables $\mathbf{r}^t$ and the estimates/auxiliary variables $\mathbf{w}^t$. The inner singularly perturbed system connects the dynamics of the estimates $\mathbf{x}^t$ and the auxiliary variables $\mathbf{z}^t$.

Lemma 1. *Let* $\mathbf{w}^t = \mathbf{w}$. *There exists a continuous function* $u(\tilde{\mathbf{r}}^t, t)$ *such that, for* $\lambda \in (1, 2)$, (3.7a), (3.7b) *and* (3.7c) *hold.*

Proof. The boundary system layer dynamics of (3.11) is

$$\tilde{\mathbf{r}}^{t+1}=\mathbf{r}^{t+1} - \mathbf{r}_{\mathsf{eq}}(\mathbf{w})=(1-\lambda)(\tilde{\mathbf{r}}^t + \mathbf{r}_{\mathsf{eq}}(\mathbf{w})) + \lambda\mathbf{h}(\mathbf{w}^t) - \mathbf{r}_{\mathsf{eq}}(\mathbf{w}), \tag{3.15}$$

from which, using (3.13), it is derived that $\tilde{\mathbf{r}}^{t+1} = (1-\lambda)\tilde{\mathbf{r}}^t$. Thus, if $\lambda \in (1, 2)$, then the error dynamics of $\tilde{\mathbf{r}}^{t+1}$ is globally exponentially stable and the conditions (3.7a), (3.7b) and (3.7c) are satisfied by choosing as Lyapunov function $u(\tilde{\mathbf{r}}^t, t) = \frac{1}{2}(\tilde{\mathbf{r}}^t)^\top\tilde{\mathbf{r}}^t$. □

Lemma 2. *Let* $\mathbf{r}^t = \mathbf{r}_{\mathsf{eq}}(\mathbf{w}^t)$, $\lambda, \mu \in (1, 2)$, $\gamma > 0$, $\epsilon, \alpha \in (0, 1)$ *and* $\rho > 0$. *There exists a continuous function* $w(\tilde{\mathbf{w}}^t, t)$ *such that* (3.8a), (3.8b) *and* (3.8c) *hold.*

Proof. First, the error dynamics $\tilde{\mathbf{w}}^{t+1}$ are

$$\tilde{\mathbf{w}}^{t+1} = \mathbf{Q}(\tilde{\mathbf{w}}^t+\mathbf{w}^*)+\mathbf{B}(\mathbf{r}_{\mathsf{eq}}(\mathbf{w}^t))-\mathbf{w}^* \Rightarrow \tilde{\mathbf{w}}^{t+1}=\mathbf{Q}\tilde{\mathbf{w}}^t+\begin{pmatrix} -\gamma\mathbf{x}^* \\ \mu\mathbf{F}\mathbf{z}^* - \mu\mathbf{z}^* \\ \mathbf{z}^* - \mathbf{z}^* \end{pmatrix}+\mathbf{B}(\mathbf{r}_{\mathsf{eq}}(\mathbf{w}^t)). \tag{3.16}$$

By substituting the definition of $\mathbf{z}^*$ from Eq. (3.14) in matrix $\mathbf{B}(\mathbf{r}_{\mathsf{eq}}(\mathbf{w}^t))$ of Eq. (3.16), we obtain that

$$\tilde{\mathbf{w}}^{t+1} = \mathbf{Q}\tilde{\mathbf{w}}^t + \begin{pmatrix} -\gamma\mathbf{x}^* + \gamma\begin{pmatrix}\mathbf{I}_{\mathsf{N}n} & -\mathbf{I}_{\mathsf{N}n}\end{pmatrix}\mathbf{h}(\mathbf{w}^t) \\ \mathbf{0}_{2nd} \\ \mathbf{0}_{2nd} \end{pmatrix}. \tag{3.17}$$

According to (3.17), the error dynamics associated to $\mathbf{z}^{t+1}$ and $\mathbf{z}^t$ do not depend on $\mathbf{x}^t$

$$\begin{pmatrix}\tilde{\mathbf{z}}^{t+1} \\ \tilde{\mathbf{z}}^t\end{pmatrix}=\begin{pmatrix}\mu\mathbf{F} & (1-\mu)\mathbf{I}_{2nd} \\ \mathbf{I}_{2nd} & \mathbf{0}_{2nd\times 2nd}\end{pmatrix}\begin{pmatrix}\tilde{\mathbf{z}}^t \\ \tilde{\mathbf{z}}^{t-1}\end{pmatrix}=\mathbf{L}\begin{pmatrix}\tilde{\mathbf{z}}^t \\ \tilde{\mathbf{z}}^{t-1}\end{pmatrix}, \tag{3.18}$$

so they can be independently analyzed. Moreover, (3.18) can be rewritten in the form of (3.3a) by adding and subtracting the identity matrix from $\mathbf{L}$. On the other hand, the error dynamics associated to $\mathbf{x}^t$

$$\tilde{\mathbf{x}}^{t+1}=\tilde{\mathbf{x}}^t+\gamma(\mathbf{H}(\mathbf{x}^t+\mathbf{A}_x\mathbf{z}^t)-\mathbf{x}^t)-\gamma\mathbf{H}(\mathbf{g}(\mathbf{x}^t)+\mathbf{A}_z\mathbf{z}^t) \tag{3.19}$$

are the error dynamics of the reduced system of the inner singularly perturbed system, which are the same to those analyzed in [Carnevale et al., 2023b]. In particular, Theorem III.1 in [Carnevale et al., 2023b] proves that Algorithm 3 is globally exponentially stable when $\rho > 0$, $\alpha \in (0,1)$ and $\gamma > 0$ sufficiently small. Conveniently, the proof of Theorem III.1 in [Carnevale et al., 2023b] refers to the same dynamics in (3.19), and therefore the result can be applied. However, this result holds only if the dynamics in (3.18) are also globally exponentially stable. Otherwise, Theorem 4 applied to the dynamics of $\mathbf{x}^t$ and $\mathbf{z}^t$ does not hold. Therefore, the next step is to prove the global exponential stability of (3.18). The error dynamics in (3.18) are globally exponentially stable if and only if $\{|\sigma_l(\mathbf{L})| < 1\}_{l=1}^{4nd}$, where $\sigma_i(\bullet)$ denotes the i-th eigenvalue of a matrix. Taking into account that $\det(\mathbf{L} - \sigma \mathbf{I}_{4nd}) = 0$ and that $\mathbf{F} = (1-\epsilon\alpha)\mathbf{I}_{2nd} - \epsilon\alpha\mathbf{P}$, the eigenvalues of $\mathbf{L}$ are

$$-\sigma^2 + \sigma(\mu - \mu\alpha\epsilon \pm \mu\alpha\epsilon) + (1-\mu) = 0. \tag{3.20}$$

The $\pm$ in (3.20) comes from the fact that $\det(\mathbf{P}) = (-1)^p$, where p is the number of permutations. If p is even,

$$-\sigma^2 + \mu\sigma + (1-\mu) = 0 \Rightarrow \sigma = \frac{\mu}{2} \pm \sqrt{\frac{\mu^2}{4} + (1-\mu)}. \tag{3.21}$$

From (3.21), $-1 < \sigma < 1$. If $\mu \in (1,2)$. If p is odd, then

$$-\sigma^2 + \mu(1 - 2\alpha\epsilon)\sigma + (1-\mu) = 0. \tag{3.22}$$

Equation (3.22) implies that

$$-1 < \frac{\mu(1-2\epsilon\alpha)}{2} \pm \sqrt{\frac{\mu^2}{4}(1-2\epsilon\alpha)^2 + (1-\mu)} < 1. \tag{3.23}$$

By solving the four inequality equations in (3.23) and taking the more restrictive conditions, we obtain that $\epsilon\alpha \in (0,1)$, and since $\alpha \in (0,1)$, then $\epsilon \in (0,1)$. Therefore, if $\epsilon \in (0,1)$ and $\mu \in (1,2)$, then the error dynamics in (3.18) are globally exponentially stable. This implies that the error dynamics in (3.19) are globally exponentially stable as well and the conditions for the inner singularly perturbed system in Theorem 4 hold. Subsequently, the error dynamics in (3.17) are globally exponentially stable and there exist a Lyapunov function $w(\tilde{\mathbf{w}}^t, t)$ such that (3.8a), (3.8b) and (3.8c) are satisfied. □

Proposition 3. *Consider Algorithm 4 with $\rho > 0$, $\alpha, \epsilon \in (0,1)$ and $\lambda, \mu \in (1,2)$. Then, there exists $\bar{\gamma}, c_1, c_2 > 0$ such that, for any $\gamma \in (0, \bar{\gamma})$, $(\mathbf{x}^0, \mathbf{z}^0, \mathbf{z}^{-1}, \mathbf{y}^0, \mathbf{s}^0)$, it holds that $||\mathbf{x}^t - \mathbf{1}_{\mathsf{N},n}^\top \mathbf{x}^*|| \leq c_1 e^{-c_2 t}$.*

Proof. Let $\tilde{\mathbf{w}}^t$ correspond to the error state of the slow dynamics in (3.3a) and $\tilde{\mathbf{r}}^t$ the error state of the fast error dynamics in (3.3b). According to Lemmas 1 and 2, the conditions (3.7a)-(3.8c) hold for the error dynamics of $\tilde{\mathbf{w}}^t$ and $\tilde{\mathbf{r}}^t$ respectively. Besides, from the definition of $\mathbf{r}_{\mathsf{eq}}(\mathbf{w}^t)$ it can be verified that condition (3.4) holds. Then, Theorem 4 can be invoked to prove the global exponential stability of $\tilde{\mathbf{w}}^t$ and $\tilde{\mathbf{r}}^t$. By changing the coordinates from $\tilde{\mathbf{w}}^t$ and $\tilde{\mathbf{r}}^t$ to $\mathbf{w}^t$ and $\mathbf{r}^t$, the proof ends. □

Proposition 3 establishes that Algorithm 4 linearly converges to the global optimum $\mathbf{x}^*$. Nevertheless, acceleration is not proved yet. This is proved in the next theorem.

Theorem 5. *Consider $\rho > 0$, $\alpha \in (0,1)$ and $\lambda \in (1,2)$, and $\gamma > 0$ sufficiently small. Also, let β_1 and β_2 be the convergence rate of the dynamics of $\mathbf{z}^t$ for Algorithm 3 and 4 respectively. Then, Algorithm 4 always converges faster than Algorithm 3 ($\beta_2 < \beta_1$) for any $\epsilon \in (0,1)$ and $\mu \in (1,2)$.*

Proof. In Lemma 2 Eq. (3.18) it is proven that the dynamics of $\tilde{\mathbf{z}}^t$ are determined by $\mathbf{L}$. Therefore, the convergence rate of $\tilde{\mathbf{z}}^t$ and subsequently $\mathbf{z}^t$ is given by $\max(\{|\sigma_l(\mathbf{L})|\}_{l=1}^{4nd})$. Therefore, we have that

$$\max(\{|\sigma_l(\mathbf{L})|\}_{l=1}^{4nd}) = \max\left(\left|\frac{\mu}{2}+\sqrt{\frac{\mu^2}{4}+(1-\mu)}\right|, \left|\frac{\mu(1-2\epsilon\alpha)}{2}+\sqrt{\frac{\mu^2}{4}(1-2\epsilon\alpha)^2+(1-\mu)}\right|\right).$$

Moreover, notice that Algorithm 3 is Algorithm 4 with $\mu = \epsilon = 1$. Thus, $\beta_1 = \max(1, |1-2\alpha|)$ for Algorithm 3. Under the assumption that no integral term is present, $\beta_1 = |1-2\alpha|$; otherwise, $\beta_1 = 1$. On the other hand, under the constraints on μ, α, ϵ in Proposition 3, we have that $\left|\frac{\mu}{2}+\sqrt{\frac{\mu^2}{4}+(1-\mu)}\right| < 1$, and also $\left|\frac{\mu(1-2\epsilon\alpha)}{2}+\sqrt{\frac{\mu^2}{4}(1-\epsilon\alpha)^2+(1-\mu)}\right| < |1-2\alpha|$. Henceforth, $\beta_2 < \beta_1$. □

Proposition 3 establishes that the slow and fast dynamics of Algorithm 4 must evolve at different time-scales to ensure convergence. Therefore, the convergence rate of the auxiliary variables determines the convergence speed of the optimum estimates. Thanks to adding momentum, γ, ρ and α can be increased without violating the conditions of time-scale decoupling of singularly perturbed systems, improving the convergence speed.

3.1.6 Illustrative example

We compare our algorithm (A2DMM-GT) with the one in [Carnevale et al., 2023b] (ADMM-GT), the accelerated gradient tracking approaches in [Xin and Khan, 2019] (AGT) and [Nguyen et al., 2023] (A2GT), and the distributed gradient tracking in [Nedic et al., 2017] (DIGing).

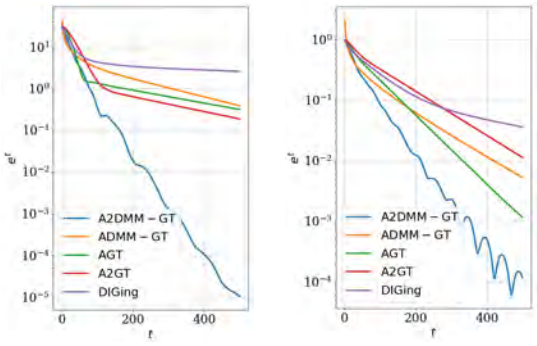


Figure 3.2: Evolution of the error with time. Left is the quadratic programming setup, right is the linear classification setup.

We consider two settings. First, a quadratic programming setup, that arises in robotic, smart grid or feature selection applications [Ubl and Hale, 2019]. We set $\mathsf{N} = 200$, $n = 2$ and $f_i(\mathbf{x}) = \mathbf{x}^\top \mathbf{B}_i \mathbf{x} + \mathbf{a}_i \mathbf{x}$, where $\mathbf{B}_i$ is a positive definite matrix with uniformly random eigenvalues in the range $[1, 5]$ and $\mathbf{a}_i$ is a random vector with elements uniformly drawn in the range $[-10, 20]$. Note that $f_i(\mathbf{x})$ fulfills the assumptions in terms of strong convexity and Lipschitzness of the gradients. Graph $\mathcal{G}$ is generated as a sparse undirected connected proximity graph, where the nodes are randomly

placed in a 2×2 square if and only if the new node is within $[0.1, 0.17]$ from an already placed node. The parameters for the all the approaches are hand-tuned to achieve fastest convergence possible: $\gamma = 1.6$, $\alpha = 0.9924$, $\rho = 3.028$, $\lambda = 0.2$, $\mu = 1.6$, $\epsilon = 0.6$ for A2DMM-GT; $\gamma = 0.4865$, $\alpha = 0.8924$, $\rho = 0.528$ for ADMM-GT; $\alpha = 0.001$, $\beta = 0.9$ for AGT (we use the notation in [Xin and Khan, 2019]); $\alpha = 0.0008$, $\beta = 0.7$, $\gamma = 0.2$ for A2GT (we use the notation in [Nguyen et al., 2023]); $\gamma = 0.0127$ for DIGing. We simulate 500 steps. Convergence is assessed using the evolution of the error $\mathbf{e}^t = ||\mathbf{x}^t - \mathbf{1}_{\mathsf{N},n}\mathbf{x}^*||$.

The second setup is a logistic regression scenario where nodes cooperate to train a linear classifier. In this case, $\mathsf{N} = 50$, where each node has a local dataset of $m = 10$ one-dimensional points p_i^j drawn from a normal distribution of mean 0 and standard deviation 1. The labels are also randomly drawn, such that $l_i^j \in \{-1, 1\}$. We define $\mathbf{x} = [q_1, q_2]$ and $f_i(\mathbf{x}) = \sum_{j=1}^{m} \log(1 + \exp^{-l_i^j(q_1 p_i^j + q_2)}) + C||\mathbf{x}||^2$, with $C = 1$ to ensure strong-convexity. The parameters for all methods are the same that in quadratic programming setup.

The evolution of the error (Fig. 3.2) shows that A2DMM-GT is the fastest method among all the approaches for both numerical settings, with different network configurations and local costs. Importantly, this is achieved without increasing the number of gradient computations per time step nor the communication burden, leading to a significant reduction in power consumption and total bandwidth usage in practical applications. This comes at the cost of increasing the memory burden compared to the non-accelerated methods. However, given current advances in MEMS technology, it is generally better to leverage memory against communication burden. The idea of exploiting current advances in MEMS and memory to achieve acceleration in convergence speed is further studied in the next section, where a novel discrete-time consensus technique for dynamic average consensus is proposed.

3.2 Consensus: accelerated multi-stage discrete-time dynamic average consensus

3.2.1 Related work

Several solutions have been proposed to improve the capabilities of discrete time dynamic average consensus algorithms. For instance, some works achieve an arbitrarily small steady-state error by exploiting the k-th differences of the input signal [Zhu and Martínez, 2010, Montijano et al., 2014b]. This is problematic when the inputs are noisy since noise breaks the boundedness of the input signal. An alternative is to rely only on the input signal, achieving an arbitrarily small steady-state error by concatenating a cascade of consensus filters [Franceschelli and Gasparri, 2016, Franceschelli and Gasparri, 2019], as we do here. Besides, this leads to other desirable properties like robustness against non-averaged initializations and changes in the network size. The counterpart is a slow convergence. Regarding robustness, there exist continuous-time algorithms [Olfati-Saber and Shamma, 2005, Bai et al., 2010, Nosrati et al., 2012, George and Freeman, 2019] that address initialization issues and time-varying networks.

The issue of slow convergence has been considered from two main perspectives. Ghosh et al. [Ghosh et al., 1996] present second order diffusion methods, which significantly speed up the convergence [Liu and Morse, 2011] by increasing the memory requirements with the previous estimate. These works are for static problems, while ours deals with the dynamic consensus problem. On the other hand, polynomial filters [Kokiopoulou and Frossard, 2008] consider a sequence of consensus iterations as the evaluation of a polynomial. For instance, Montijano et al. [Montijano et al., 2012, Montijano et al., 2014b] analyze Chebyshev polynomials, proving that they significantly increase the convergence speed. However, the k-th order differences of the input signal are exploited. Beyond consensus, compression techniques [Donge et al., 2023] or decomposition principles [Sadamoto et al., 2020] can be used to reduce the dimensionality of the data and computational cost.

Finally, regarding robustness against potential communication delays and packet losses, many applications deal with time-varying networks, where nodes connect and disconnect depending on events exogenous to the consensus protocol. To account for this, gossip algorithms [Boyd et al., 2006, Liu et al., 2011] propose consensus protocols computed at random asynchronous instants and neighbors. Other works either propose reformulations over linear proportional consensus protocols against packet drops [Fagnani and Zampieri, 2009] or continuous-time protocols against delays [Moradian and Kia, 2018].

3.2.2 Preliminaries

The system under study is a network composed by $\mathsf{N} > 1$ nodes. The network is described by an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{1, \ldots, i, \ldots, \mathsf{N}\}$ is the set of nodes and $\mathcal{E}$ is the set of edges. Nodes i and j can exchange information if and only if $(i, j) \in \mathcal{E}$, which implies that $(j, i) \in \mathcal{E}$. $\mathcal{N}_i = \{j | (j, i) \in \mathcal{E}\}$ is the neighborhood of node i. The adjacency matrix $\mathbf{A} \in \mathbb{R}^{\mathsf{N}\times\mathsf{N}}$ associated to $\mathcal{G}$ is such that $\mathbf{A}_{ij} = 1$ if $(i, j) \in \mathcal{E}$ and 0 otherwise. $\mathbf{A}_{ii} = 0$ always because we do not allow self-loops. The degree matrix $\mathbf{D}$ associated to $\mathcal{G}$ is such that $\mathbf{D}_{ij} = \mathsf{car}(\mathcal{N}_i)$ $\forall i = j$ and 0 otherwise. The Laplacian matrix associated to $\mathcal{G}$ is $\mathbf{L} = \mathbf{D} - \mathbf{A}$. For undirected graphs, it holds that $\mathbf{L}$ is symmetric and has real eigenvalues. The sorted eigenvalues of $\mathbf{L}$ are $\lambda_1(\mathbf{L}) < \ldots \leq \lambda_N(\mathbf{L}) \leq 2D_{\max}$, where the relation $D_{\max} = \max(\{\mathbf{D}_{ii}\}_{i=1}^{\mathsf{N}})$ holds. The second smallest eigenvalue, $\lambda_2(\mathbf{L})$, is called *algebraic connectivity*. We denote $\mathbf{v}_{r,i}(\mathbf{L}), \mathbf{v}_{l,i}(\mathbf{L})$ the right and left eigenvectors of $\mathbf{L}$ associated to the i-th eigenvalue.

3.2.3 Second order diffusion methods

The linear (first-order) discrete time *static* average consensus algorithm [Bullo et al., 2009] is

$$x_i(k+1) = \mathbf{W}_{ii}x_i(k) + \textstyle\sum_{j\in\mathcal{N}_i} \mathbf{W}_{ij}x_j(k), \tag{3.24}$$

with $x_i(0)$ the initial condition and $\mathbf{W} \in \mathbb{R}^{\mathsf{N}\times\mathsf{N}}$ a weighted matrix. In matrix form, (3.24) leads to

$$\mathbf{x}(k+1) = \mathbf{W}\mathbf{x}(k). \tag{3.25}$$

If $\mathbf{W}$ is doubly stochastic, i.e., $\mathbf{1}^T\mathbf{W} = \mathbf{1}^T$ and $\mathbf{W1} = \mathbf{1}$, then it is known that the protocol in Eq. (3.25) converges to the average of the initial states $\bar{x}_i(0) = (1/\mathsf{N})\sum_{i=1}^{\mathsf{N}} x_i(0)$.

Without loss of generality, we consider $\mathbf{W} = \mathbf{I} - \epsilon\mathbf{L}$. The convergence speed can be too slow, specially in networks with a small algebraic connectivity ($\lambda_2(\mathbf{W}) = 1 - \epsilon\lambda_2(\mathbf{L})$). To accelerate convergence, Ghosh et al. [Ghosh et al., 1996] proposed a second-order modification:

$$\mathbf{x}(k+1) = \gamma\mathbf{W}\mathbf{x}(k) + (1-\gamma)\mathbf{x}(k-1). \tag{3.26}$$

The properties of the protocol in Eq. (3.26) are well-studied. In particular, protocol (3.26) improves the convergence speed of (3.25) if $\gamma \in (1, 2)$.

3.2.4 Multi-stage discrete time dynamic average consensus

The concepts in subsection 3.2.3 apply to the static consensus problem, but this section deals with a dynamic consensus problem. Node i has an input $r_i(k) \in \mathbb{R}$ and an estimate $x_i(k) \in \mathbb{R}$ associated to a quantity of interest $r(k) \in \mathbb{R}$. $\mathbf{x}(k) = [x_1(k), \ldots, x_N(k)]^T$ is the joint estimate and $\mathbf{r}(k) = [r_1(k), \ldots, r_N(k)]^T$ is the joint input of the network. Nodes, by means of $\mathbf{x}(k)$, cooperate to track the average $\bar{r}(k) = \frac{1}{\mathsf{N}}\sum_{i=1}^{\mathsf{N}} r_i(k)$ by exchanging information with their neighbors $j \in \mathcal{N}_i$.

The multi-stage discrete time dynamic average consensus algorithms presented in the works [Franceschelli and Gasparri, 2016,Franceschelli and Gasparri, 2019] solve the problem for two cases: the synchronous and time-invariant topology, and the asynchronous and randomized topology. The former is solved by the following protocol:

$$\mathbf{x}^s(k+1) = (\mathbf{I} - \epsilon\mathbf{L})\mathbf{x}^s(k) + \alpha(\mathbf{x}^{s-1}(k) - \mathbf{x}^s(k)). \tag{3.27}$$

Here, $s = \{1, \ldots, m\}$ denotes the stages of the filter, $\mathbf{x}^s(k)$ is the estimate at instant k and stage s, $\mathbf{x}^0(k) = \mathbf{r}(k)$, parameter $\epsilon < \frac{1}{2D_{\max}}$ to ensure stability, and $\alpha \in (0, 1)$ is a parameter that trades-off steady-state accuracy and convergence speed. From (3.27) it is seen that the protocol is a chain of linear discrete dynamic average consensus filters indexed by s.

The asynchronous and randomized algorithm follows the same multi-stage architecture. In this case, each node selects randomly and at each instant a single node $j \in \mathcal{N}_i$ to communicate. For a given edge (i, j), the algorithm is:

$$\mathbf{x}^s(k+1) = \mathbf{P}_{ij}\mathbf{x}^s(k) + \frac{\alpha}{\mathbf{D}_{ii}}\mathbf{p}_i\mathbf{p}_i^T\mathbf{x}^{s-1}(k), \tag{3.28}$$

with $\mathbf{P}_{ij} = \mathbf{I} + \frac{\mathbf{p}_i\mathbf{p}_j^T}{2} - \frac{(1+2/\mathbf{D}_{ii})\mathbf{p}_i\mathbf{p}_i^T}{2}$ and $\mathbf{p}_i \in \mathbb{R}^{\mathsf{N}}$ a vector of zeros except for the $i-$th element, which is equal to 1.

As shown in [Franceschelli and Gasparri, 2016, Franceschelli and Gasparri, 2019], the steady-state error and convergence rate of protocols (3.27) and (3.28) are tied together by α: for a fixed m, small values of α lead to quick convergence but large steady-state error and vice versa. The trade-off can be alleviated by m, but at the expense of an increase of computation, memory and convergence time at final stages. Importantly, it

is proved in [Franceschelli and Gasparri, 2019] that both algorithms are robust against arbitrary initializations and noisy and unbounded inputs due to the absence of integral control actions and input differences respectively. We propose a second order diffusion protocol that, by adding an additional parameter, decouples the steady-state error and the convergence speed, overcoming the undesired trade-off in the multi-stage protocols.

As a remark, the stages of protocols (3.27), (3.28) run in parallel, in the sense that, to estimate at stage s and instant $k+1$, we only need information from $s-1$ and s at time k.

3.2.5 Accelerated multi-stage filter

The first proposed consensus protocol is based on two steps. Given the current input $\mathbf{r}_i(k)$ and estimates from neighbors $\{x_j^s(k)\}_{s=0}^m \; \forall j \in \mathcal{N}_i \cup \{i\}$, node i updates $\{x_i^s(k)\}_{s=0}^m$ following the multi-stage filter in Eq. (3.27), obtaining $\{\tilde{x}_i^s(k)\}_{s=0}^m$, where $\tilde{x}_i^s(k) \in \mathbb{R}$ for $s = 1, \dots, m$ is a temporal estimate used in the second step of the protocol. The updated estimate is then corrected using the second order method in Eq. (3.26), leading to the next estimate $\{x_i^s(k+1)\}_{s=0}^m$. Algorithm 5 details the protocol.

Algorithm 5 Accelerated Multi-Stage Dynamic Consensus Protocol at node i

1: **State of agent**: $x_i^s(-1)$ and $x_i^s(0)$, for $s = 1, \dots, m$
2: **Parameters**: $\gamma \in (1,2)$, $\epsilon \in (0, \frac{1}{2D_{\max}})$, $\alpha \in (0, \frac{1}{\gamma})$
3: **while** True **do**
4: Measure $r_i(k)$ and gather $x_j^s(k)$ for all s and $j \in \mathcal{N}_i$
5: Update $x_i^s(k)$ for $s = 1, \dots, m$ as follows:

$$\tilde{x}_i^s(k) = x_i^s(k) - \sum_{j \in \mathcal{N}_i} \epsilon (x_i^s(k) - x_j^s(k)) + \alpha (x_i^{s-1}(k) - x_i^s(k))$$

$$x_i^s(k+1) = \gamma(\tilde{x}_i^s(k)) + (1-\gamma) x_i^s(k-1) \; \forall s$$

6: **end while**

Note that Algorithm 5 is equal to (3.27) for $\gamma = 1$. First, for space convenience, in the following we use the sub-index $_k$ to abbreviate (k). We now prove the convergence properties of the proposed protocol. The next result shows that the second order method does not alter the steady-state properties of the original multi-stage filter in [Franceschelli and Gasparri, 2016,Franceschelli and Gasparri, 2019], which is important in the subsequent results about the convergence rate of the accelerated protocol.

Proposition 4. *Assume that* $\mathbf{r}_k = \mathbf{r}$, $\mathcal{G}$ *connected,* $\alpha \in (0,1)$, *and* $\epsilon \in (0, \frac{1}{2D_{\max}})$. *Then, the steady-state equilibrium* $\mathbf{x}^{m,*}$ *of the protocol in* (3.27) *and Algorithm 5 is*

$$\mathbf{x}^{m,*} = \bar{r}\mathbf{1} + \textstyle\sum_{i=2}^{\mathsf{N}} (1 + \epsilon\lambda_2(\mathbf{L})/\alpha)^{-m} \mathbf{v}_{r,i}(\mathbf{L}) \mathbf{v}_{l,i}^T(\mathbf{L}) \mathbf{r}. \tag{3.29}$$

Proof. From Algorithm 5 and using the matrix form, the operations at stage $s = 1$ and steady-state can be written in a single equation

$$\mathbf{x}^{1,*} = \gamma(\mathbf{I} - \epsilon\mathbf{L})\mathbf{x}^{1,*} + \gamma\alpha\mathbf{r} - \gamma\alpha\mathbf{x}^{1,*} + (1-\gamma)\mathbf{x}^{1,*}. \tag{3.30}$$

This leads to $((1+\gamma\alpha-1+\gamma-\gamma)\mathbf{I}+\gamma\epsilon\mathbf{L})\,\mathbf{x}^{1,*} = \gamma\alpha\mathbf{r}$. Dividing both sides by γ, and concatenating the m stages,

$$\mathbf{x}^{m,*} = (\alpha\mathbf{I}+\epsilon\mathbf{L})^{-1}\alpha^m\mathbf{r}, \tag{3.31}$$

which is the steady-state equilibrium in the proof of Theorem 3.1 in [Franceschelli and Gasparri, 2019]. The rest of the proof follows from there. □

Corollary 2. *Assume that $\mathbf{r}_k = \mathbf{r}$ is constant and $\mathcal{G}$ connected. Then, the 2-norm of the error at equilibrium of a network under Algorithm 5 is*

$$||\bar{r}\mathbf{1}-\mathbf{x}^{m,*}|| \leq (\mathsf{N}-1)(1+\epsilon\lambda_2(\mathbf{L})/\alpha)^{-m}||\bar{r}\mathbf{1}-\mathbf{r}||. \tag{3.32}$$

The statement is a direct consequence of Proposition 4 and Theorem 3.2 in [Franceschelli and Gasparri, 2016], showing that the steady-state error decreases when: the graph is more connected (greater $\lambda_2(\mathbf{L})$), the protocol has more stages (greater m), and α gets smaller. The following result proves necessary equivalences to demonstrate the convergence properties of Algorithm 5.

Proposition 5. *Let $\mathbf{y}_k^s = \mathbf{x}_k^s - \mathbf{x}^{s,*}$ be the error at the s-th stage of the filter in Algorithm 5. Then, the error dynamics can be expressed as*

$$\begin{pmatrix}\mathbf{y}_{k+1}^s\\ \mathbf{y}_k^s\end{pmatrix} = \hat{\mathbf{Q}}\begin{pmatrix}\mathbf{y}_k^s\\ \mathbf{y}_{k-1}^s\end{pmatrix} + \alpha\hat{\mathbf{R}}\begin{pmatrix}\Delta\mathbf{u}_k^s\\ \Delta\mathbf{u}_{k-1}^s\end{pmatrix}, \tag{3.33}$$

where $\hat{\mathbf{Q}} = \begin{pmatrix}\gamma\mathbf{Q} & (1-\gamma)\mathbf{I}\\ \mathbf{I} & \mathbf{0}\end{pmatrix}$, $\hat{\mathbf{R}} = \begin{pmatrix}\mathbf{R} & \mathbf{R}(1-\gamma)\\ \mathbf{0} & \mathbf{0}\end{pmatrix}$, $\mathbf{Q} = (1-\alpha)\mathbf{I}-\epsilon\mathbf{L}$, $\mathbf{R} = (\alpha\mathbf{I}+\epsilon\mathbf{L})^{-1}$, $\Delta\mathbf{u}_k^s = \mathbf{u}_k^s - \mathbf{u}_{k+1}^s$, $\mathbf{u}_k^1 = \mathbf{r}_k$ *and* $\mathbf{u}_k^s = \mathbf{x}_k^{s-1}$ *for* $s = 2,\dots,m$.

Proof. Using the steps in Algorithm 5 in a single operation and matrix form,

$$\mathbf{y}_{k+1}^s = \gamma\mathbf{Q}\mathbf{y}_k^s+\gamma\mathbf{x}^{s,*}-\mathbf{x}^{s,*}+(1-\gamma)\mathbf{x}_{k-1}^s = \gamma\mathbf{Q}\mathbf{y}_k^s+\gamma\mathbf{x}^{s,*}-\mathbf{x}^{s,*}+(1-\gamma)\mathbf{y}_{k-1}^s+(1-\gamma)\mathbf{x}^{s,*} \tag{3.34}$$

Eq. (3.34) can be rewritten using Eq. (3.31):

$$\mathbf{y}_{k+1}^s = \gamma\mathbf{Q}\mathbf{y}_k^s+(1-\gamma)\mathbf{y}_{k-1}^s+\alpha\mathbf{R}\Delta\mathbf{u}_k^s+(1-\gamma)\alpha\mathbf{R}\Delta\mathbf{u}_{k-1}^s. \tag{3.35}$$

Eq. (3.33) follows from Eq.(3.35), concluding the proof. □

Now we are ready to prove the convergence rate of Algorithm 5 .

Theorem 6. *Consider a network that executes Algorithm 5 with $\mathbf{r}_k = \mathbf{r}$ constant, with $\alpha \in (0, 1/\gamma)$, $\gamma \in (1,2)$ and $\epsilon \in (0, 1/2D_{\max})$. Then, the convergence rate for the s-th stage is $\beta^s = 1-\alpha\gamma$.*

Proof. Let us consider the following Lyapunov function:

$$V_k^s = \begin{pmatrix}\mathbf{y}_k^{s,T} & \mathbf{y}_{k-1}^{s,T}\end{pmatrix}\begin{pmatrix}\mathbf{y}_k^{s,T} & \mathbf{y}_{k-1}^{s,T}\end{pmatrix}^T = \hat{\mathbf{y}}_k^s\hat{\mathbf{y}}_k^{s,T}, \tag{3.36}$$

with $\hat{\mathbf{y}}_k^s = \begin{pmatrix} \mathbf{y}_k^{s,T} & \mathbf{y}_{k-1}^{s,T} \end{pmatrix}$. Then, the Lyapunov difference is

$$\Delta V_k^s = V_{k+1}^s - V_k^s = \hat{\mathbf{y}}_{k+1}^s \hat{\mathbf{y}}_{k+1}^{s,T} - \hat{\mathbf{y}}_k^s \hat{\mathbf{y}}_k^{s,T}. \tag{3.37}$$

Exploiting Eq. (3.33) and the fact that $\mathbf{r}_k$ is constant, Eq. (3.37) leads to

$$\Delta V_k^s = \hat{\mathbf{y}}_k^s (\hat{\mathbf{Q}}\hat{\mathbf{Q}}^T - \mathbf{I}) \hat{\mathbf{y}}_k^{s,T}. \tag{3.38}$$

Now, since $\gamma \in (1, 2)$ and the spectral radius of $\mathbf{Q}$ is less than 1 by construction, it can be shown that Eq. (3.38) is upper-bounded by the following inequality:

$$\Delta V_k^s \leq -\hat{\mathbf{y}}_k^s \begin{pmatrix} \gamma^2\mathbf{I} - \gamma^2\mathbf{Q}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \hat{\mathbf{y}}_k^{s,T} = -\hat{\mathbf{y}}_k^s \mathbf{F} \hat{\mathbf{y}}_k^{s,T}. \tag{3.39}$$

The eigenvalues of matrix $\mathbf{F}$ are the eigenvalues of $\gamma^2\mathbf{I} - \gamma^2\mathbf{Q}^2$ and $\mathbf{I}$. If $\alpha < \frac{1}{\gamma}$, then $\gamma^2\mathbf{I} - \gamma^2\mathbf{Q}^2 \succeq \gamma\mathbf{I} - \gamma\mathbf{Q}$ and $\sigma(\gamma\mathbf{I} - \gamma\mathbf{Q}) > 0$, where $\succeq$ denotes positive semidefiniteness. Therefore, $\mathbf{F}$ is positive definite, and

$$V_{k+1}^s - V_k^s \leq -\bar{\lambda} \hat{\mathbf{y}}_k^s \hat{\mathbf{y}}_k^{s,T} = -\bar{\lambda} V_k^s. \tag{3.40}$$

It turns out that $\bar{\lambda} = \min_{\lambda_i \in \sigma(\gamma\mathbf{I} - \gamma\mathbf{Q}) \cup \sigma(\mathbf{I})} \{\lambda_i\} = \alpha\gamma$. Thus, $V_{k+1}^s \leq (1 - \alpha\gamma) V_k^s$. □

Once the convergence rate of Algorithm 5 is proved, we can compare it with its non-accelerated counterpart.

Proposition 6. *Algorithm 5 always converges faster than the protocol in Eq.* (3.27) *if* $\gamma \in (1, 2)$.

Proof. The convergence rate of the protocol in Eq. (3.27) is $\beta^s = 1 - \alpha$, which coincides with the convergence rate obtained in [Franceschelli and Gasparri, 2016]. By comparing both convergence rates, we have that $1 - \alpha > 1 - \alpha\gamma$ for $\gamma \in (1, 2)$. Thus, the bound in Eq. (3.40) is smaller, and the decrease in the Lyapunov function defined in Eq. (3.36) is greater, which implies a faster convergence of Algorithm 5 than the protocol in Eq. (3.27). □

These results draw interesting properties. First, for any choice of m and α of the original algorithm, such that α also fulfills the condition $\alpha \in (0, 1/\gamma)$ with $\gamma \in (1, 2)$, then Algorithm 5 always converges faster than the non-accelerated protocol (3.27). Besides, while α ties together steady-state and convergence rate in both the original and accelerated algorithm, on the original protocol we can only exploit m to further tune the steady-state error. Meanwhile, on Algorithm 5, by introducing γ, we have an additional degree of freedom to also further tune the convergence, independent from the steady-state error. Thus, the accelerated protocol provides more flexibility in simultaneously tuning the steady-state error and convergence rate.

All the previous results address the case where the input is constant. The following result considers a dynamic input.

Proposition 7. *Consider a time-varying input* $\mathbf{r}_k$ *and a network under Algorithm 5, with* $\alpha \in (0, 1/\gamma)$*. Besides, define* $\Delta\mathbf{u}^s_\infty = \sup_{k=0,\dots,\infty}\Delta\mathbf{u}^s_k$*. Then,*

$$||\mathbf{y}^s_k|| \leq (1-\gamma\alpha)^k||\mathbf{y}^s_0|| + 1/(\alpha\gamma)\cdot\Delta\mathbf{u}^s_\infty. \tag{3.41}$$

Proof. The result follows from Input-to-State Stability results for linear systems. From Eq. (3.35) we have that

$$||\mathbf{y}^s_k|| \leq ||\mathbf{K}^k||\cdot||\mathbf{y}^s_0|| + \alpha\left(\textstyle\sum_{h=0}^{k}||\mathbf{K}^h||||\mathbf{R}||\right)\Delta\mathbf{u}^s_\infty, \tag{3.42}$$

with $\mathbf{K} = (\gamma\mathbf{Q} + (1-\gamma)\mathbf{I})$. First, note that $||\mathbf{R}|| \leq \frac{1}{\alpha}$. Second, $||\mathbf{K}|| \leq (1-\gamma) + \gamma(1-\alpha) = 1-\gamma\alpha$. Then,

$$||\mathbf{y}^s_k|| \leq (1-\gamma\alpha)^k||\mathbf{y}^s_0|| + \alpha\left(\textstyle\sum_{h=0}^{k}\frac{(1-\gamma\alpha)^h}{\alpha}\right)\Delta\mathbf{u}^s_\infty. \tag{3.43}$$

Eq. (3.43) directly yields to Eq. (3.41). □

Therefore, under dynamic inputs the system still converges. Moreover, the convergence rate only depends on the variations of the signal.

Corollary 3. *The convergence rate of Algorithm 5 is faster than the convergence rate of the original filter in* (3.27) *under time-varying input signals* $\mathbf{r}_k$*,* $\alpha \in (0, \gamma^{-1})$ *and* $\gamma \in (1, 2)$*.*

Proof. In Proposition 3.5 from [Franceschelli and Gasparri, 2016] it is shown that

$$||\mathbf{y}^s_k|| \leq (1-\alpha)^k||\mathbf{y}^s_0|| + 1/\alpha\cdot\Delta\mathbf{u}^s_\infty. \tag{3.44}$$

Comparing Eqs. (3.43) and (3.44), the first term experiences a faster decay in (3.43) since $1-\gamma\alpha < 1-\alpha$. The second term is lower in (3.43) since $(\alpha\gamma)^{-1} < \alpha^{-1}$. □

In summary, the proposed algorithm proves to enhance the convergence speed towards the dynamic average consensus without modifying the steady-state error at any of the stages of the filter. This is achieved by using an additional memory slot for the estimate at the previous instant of time.

To design the algorithm, a possible way of proceed is the following: (i) set $\beta^s \in (0, 1)$, which implies that $\alpha\gamma = 1-\beta^s \in (0, 1)$; (ii) therefore, $\alpha = (1-\beta^s)/\gamma$, which is always feasible because $(1-\beta^s)/\gamma < 1/\gamma$; (iii) at this point, set the desired steady-state error and choose m according to Corollary 2 to achieve the desired performance. Notably, the computational and memory cost of Algorithm 5 grows linearly with the number of stages, and thus increasing m is a scalable design decision, considering also the current advances in PMEMS technology. For instance, if we consider the Ethernet protocol, with a frame between 64 and 1518 bytes and a 18-byte header, we can send between 11 and 375 floating point numbers, equivalent to $m \in [11, 375]$. Thus, we can fully exploit the real structure of communication networks. In addition, by leveraging the results provided in [Franceschelli and Gasparri, 2019], Algorithm 5 inherits the robustness against initialization and noisy unbounded inputs.

3.2.6 Accelerated asynchronous randomized protocol

The second contribution of this section is the asynchronous and randomized version of Algorithm 5. Now, instead of using all the estimates from the neighborhood, node i selects one of its neighbors $j \in \mathcal{N}_i$ according to an independent and identical distribution (i.i.d.) with uniform probability. The rest of the protocol follows the same reasoning, detailed in Algorithm 6.

The main result in this section shows the equivalence in expectation between Algorithm 6 and Algorithm 5. Then, the results provided in subsection 3.2.5 can be extrapolated to the asynchronous and randomized setting.

Theorem 7. *Consider a network under Algorithm 6 with $\mathcal{G}$ connected, $\mathbf{r}(k) = \mathbf{r}$ constant, $\alpha \in (0, \frac{1}{\gamma})$, and $\gamma \in (1, 2)$. If the sequence of selected edges is i.i.d. with uniform probability, then Algorithm 6 preserves the steady-state error properties of the original filter in* (3.28)*, i.e,:*

- $\mathbf{x}^m(k)$ *converges in distribution to a random variable $\mathbf{x}^m_\infty$, and this distribution is unique.*
- $\lim_{k\to\infty} \mathsf{E}[\mathbf{x}^m(k)] = \mathsf{E}[\mathbf{x}^m_\infty] = \mathbf{x}^{m,*}$, *where $\mathsf{E}[\bullet]$ is the expectancy operator.*

Proof. The first statement of the proof is a direct extrapolation of the proof of Theorem 6 in [Franceschelli and Gasparri, 2019]. Algorithm 6 fulfills all the following requirements: discrete time, Schur stability, affine dynamics and the sequence of edges is i.i.d. with uniform probability. Regarding the second statement, Theorem 4.1 in [Franceschelli and Gasparri, 2016] proves

$$\mathsf{E}[\bar{\mathbf{x}}^1(k+1)] = (\mathbf{I} - \epsilon'\mathbf{L})\mathsf{E}[\mathbf{x}^1(k)] + \alpha'(\mathbf{r} - \mathsf{E}[\mathbf{x}^1(k)]), \tag{3.45}$$

with $\epsilon' = \frac{1}{2\mathsf{car}(\mathcal{E})}$ and $\alpha' = \frac{\alpha}{\mathsf{car}(\mathcal{E})}$. Accordingly, the protocol in Algorithm 6 is

$$\mathsf{E}[\mathbf{x}^1(k+1)] = \gamma(\mathbf{I} - \epsilon'\mathbf{L})\mathsf{E}[\mathbf{x}^1(k)] + \gamma\alpha'(\mathbf{r} - \mathsf{E}[\mathbf{x}^1(k)]) + (1-\gamma)\mathsf{E}[\mathbf{x}^1(k-1)]). \tag{3.46}$$

The rest of the proof follows from the fact that, if we replace $\mathsf{E}[\mathbf{x}^1(k)] = \mathsf{E}[\mathbf{x}^1(k+1)] = \mathsf{E}[\mathbf{x}^1(k-1)] = \mathsf{E}[\mathbf{x}^{m,*}(k)]$, then Eq. (3.46) and Eq. (3.30) are the same, so the same procedure can be followed to prove the second statement. □

Therefore, Algorithms 5 and 6 are equivalent in expectation. Interestingly, another advantage of our accelerated proposal is that α can be tuned to have lower values while preserving the convergence speed, leading to lower noise. In addition, Algorithm 6 is easier to implement in a real device because no synchronization is needed, and an inherent robustness against delays and packet losses is achieved, following the considerations in [Franceschelli and Gasparri, 2019]. Besides, since the tracking properties are independent on the initialization, the protocol is robust against a varying number of nodes. Regarding the estimates' variance, the sequence of consensus stages act as a filter, reducing the variance from stage to stage. This is observed in Algorithm 6, where the neighboring estimates are averaged

Algorithm 6 Accelerated Asynchronous Randomized Multi-Stage Dynamic Consensus Protocol at node i

1: **State of agent**: $x_i^s(-1)$ and $x_i^s(0)$, for $s = 1, \dots, m$
2: **Parameters**: $\gamma \in (1,2)$, $\alpha \in (0, \frac{1}{\gamma})$, $D_i = \mathbf{L}_{ii}$
3: **while** True **do**
4: Measure $r_i(k)$ and pick a random neighbor $x_j^s(k)$
5: Update $x_i^s(k)$ for $s = 1, \dots, m$ as follows:

$$\tilde{x}_i^s(k) = (x_i^s(k) + x_j^s(k))/2 + \alpha/D_i(x_i^{s-1}(k) - x_i^s(k))$$

$$x_i^s(k+1) = \gamma(\tilde{x}_i^s(k)) + (1-\gamma)x_i^s(k-1) \ \forall s$$

6: **end while**

and corrected by the estimate from the previous stage. We empirically verify this fact in section 3.2.7, leaving its theoretical characterization for future work.

Finally, while protocols (3.27) and (3.28) allow $\alpha \in (0,1)$, Algorithms 5 and 6 constrain $\alpha \in (0, 1/\gamma)$. This is not a limitation of the proposed algorithms. The discussion after Corollary 3 shows how to ensure that $\alpha < 1/\gamma$ always.

3.2.7 Illustrative examples

To evaluate Algorithms 5 and 6, we consider an undirected graph of $\mathsf{N} = 8$ nodes with $\mathcal{E} = \{(1,2),(1,5),(1,8),(2,3),(2,8),(3,4),(3,6),(4,8),(6,7),(7,8)\}$. The parameters are $\alpha = 0.04$, $\epsilon = 0.01$, and $m = 5$. The signals $r_i(k)$ evolve according to a uniform random process, such that, every 2000 steps, $r_i(k) \sim \mathcal{U}(0,1) \ \forall i \in \mathcal{V}$. Besides, we set the initial estimates for all stages as $\mathbf{x}^s(0) = [0.99, 0.27, 0.02, 0.48, 0.18, 0.24, 0.65, 0.50]^T$. This initialization is random, according to the inherited robustness of the multi-stage protocol [Franceschelli and Gasparri, 2019]. The value of γ can be computed by means of distributed algorithms that estimate the algebraic connectivity of the graph (see, e.g., [Franceschelli et al., 2013, Montijano et al., 2017]).

The evolution of the estimates for the synchronous and non-randomized algorithms is shown in Fig. 3.4. With the additional memory slot, the convergence time has been substantially improved by Algorithm 5 (Fig. 3.4 (right)) compared to the original non-accelerated protocol (Fig. 3.4 (left)), while maintaining the steady-state performance. To better compare the steady-state error and convergence speed, Fig. 3.3 draws the absolute error between the average estimate across the network $e^s(k) = \frac{1}{\mathsf{N}} \sum_{i=1}^{\mathsf{N}} |x_i^s(k) - r_i(k)|$ and the

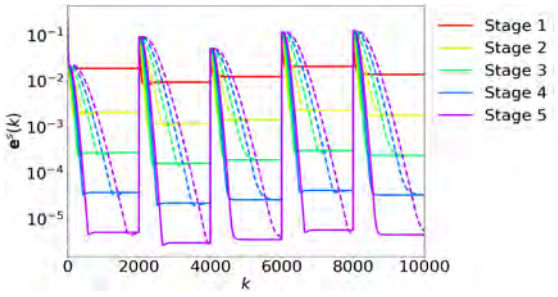


Figure 3.3: Evolution of the error between the average of the estimates and the input signals. The accelerated multi-stage filter is in solid lines, whereas the original multi-stage filter is in dashed lines.

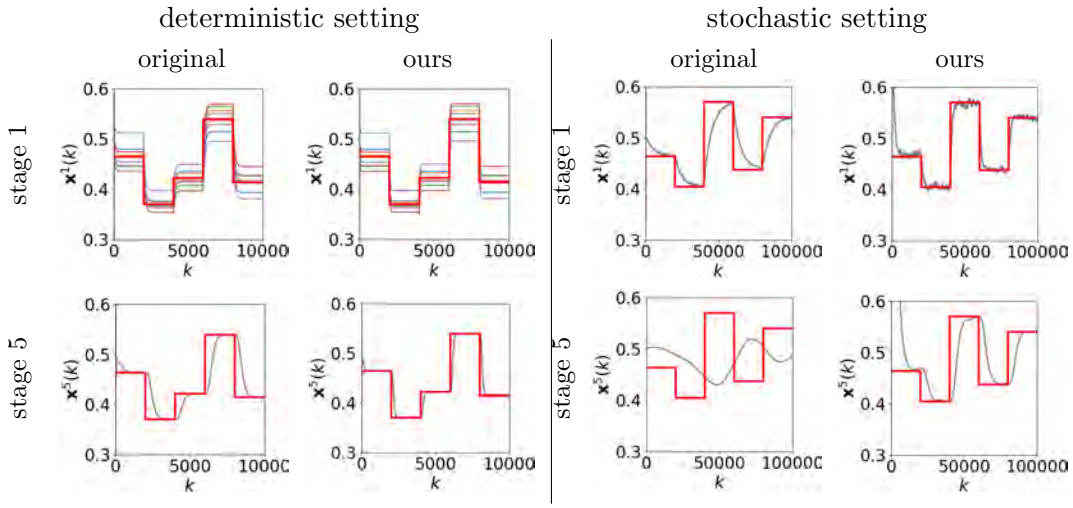


Figure 3.4: Simulation results for (left) the original multi-stage and (right) the accelerated multi-stage algorithm (ours). The red bold line is the average to be tracked, whereas the local estimates are depicted in thin random color lines.

average of the input signals $\mathbf{e}(k) = [e^1(k), \ldots, e^m(k)]^T$. For the same convergence speed, the accelerated filter can be designed with more stages and improve the steady-state error in various orders of magnitude.

In the randomized algorithms, we set $\alpha = 0.0005$. Besides, since the topology changes arbitrarily, we fix $\gamma = 1.7$. Finally, the signals $r_i(k)$ change every 20000 steps. Fig. 3.4 also represents the result of the experiments. With a low value of α, the noise due to the randomized links is filtered. The accelerated filter can compensate the degradation in convergence speed, while the original filter is too slow to converge before the signals change. Thus, a single additional memory slot per stage leads to an acceleration that overcomes the trade-off between speed and accuracy present in the original protocols in [Franceschelli and Gasparri, 2016, Franceschelli and Gasparri, 2019].

The methods presented in this section and the previous section address problem settings where robots aim to agree on common optimization goals or the average of quantities of interest. Nevertheless, there are problem settings, such as in multi-robot collision avoidance, where robots aim to disagree on, e.g., their motion. The next section explores novel dissensus techniques based on nonlinear opinion dynamics for fast and flexible decision-making in collision avoidance in dynamic environments.

3.3 Dissensus: adaptive optimal collision avoidance driven by opinion

3.3.1 Related work

Research on collision avoidance was primarily dominated by geometrical approaches. Departing from robot observations, methods like potential fields [Khatib, 1986], dynamic

window [Brock and Khatib, 1999], inevitable collision states [Petti and Fraichard, 2005] or social forces [Ferrer et al., 2013] compute control actions that steer the robot towards collision-free regions; they are simple to compute but lack formal performance guarantees, leading to over-conservative policies. By contrast, VO-based methods [Vesentini et al., 2024] preserve the computational simplicity but guarantee optimality in terms of deviation with respect to the desired predefined velocity either for holonomic or non-holonomic robots [Alonso-Mora et al., 2013, Rufli et al., 2013, Bareiss and Van den Berg, 2015]. Specially relevant is ORCA [Van Den Berg et al., 2011], which builds a half-space geometrical constraint per agent that is incorporated in a linear program. ORCA and other methods based on Reciprocal VO (RVO) assume reciprocity [Van den Berg et al., 2008], i.e, the robot and agent cooperate equally to avoid the collision, which is not the case in mixed cooperative/non-cooperative environments such the ones considered in this chapter.

More recent solutions [Qin et al., 2023, Han et al., 2022] propose deep learning approaches for collision avoidance in robot navigation [Long et al., 2017, Everett et al., 2021]. Either from a supervised [Tai et al., 2018, Pokle et al., 2019, Xie et al., 2021, Sebastián et al., 2023b] or reinforcement learning perspective [Long et al., 2018, Fan et al., 2020, Ourari et al., 2022, Cui et al., 2023, Martinez-Baselga et al., 2023, Sebastian et al., 2023], deep learning develop end-to-end policies that map the perception observations into control commands [Han et al., 2020] or sub-goals that are fed into a low-level horizon planning controller [Brito et al., 2021]. The lack of guarantees of purely learning-based methods is addressed by reachability methods [Bajcsy et al., 2019], control barrier functions [Wang et al., 2017] and model predictive control [Morgan et al., 2014]. Different from these alternatives, our proposed approach is inexpensive to compute, is not over-conservative as it minimizes the deviation with respect to the desired velocity, and does not suffer from symmetry deadlocks and oscillations.

One key characteristic of the aforementioned methods is that, in non-cooperative environments with unknown intents, a prediction model of the behavior of the other agents is required to effectively avoid collisions [de Groot et al., 2023, de Groot et al., 2024]. The first option is to use learned predictors that directly obtain future trajectory states of the agents given current and past samples [Katyal et al., 2020, Liu et al., 2023]. The second option is to rely on a learned model [Mavrogiannis et al., 2022, Poddar et al., 2023]. In both cases, the success of the collision avoidance module is subject to the accuracy of the complex prediction modules; instead, we exploit dissensus techniques to adapt to unknown degrees of cooperation through a novel nonlinear opinion dynamics adaptive law that is inexpensive to compute, does not require learning and is only based on perception observations. Opinion dynamics [Altafini, 2012, Jia et al., 2015, Cisneros-Velarde et al., 2020] originate from studies on how to model social interactions. The ineffectiveness of linear models to represent behaviors like saturation of information leads to nonlinear opinion dynamics [Bizyaeva et al., 2022], standing out as a promising dissensus method for fast and flexible decision-making. Regarding collision avoidance, an important property of nonlinear opinion dynamics is that they ensure the existence of a bifurcation in the state-space, i.e., the existence of two simultaneous stable equilibrium points. This feature has been exploited [Cathcart et al., 2023] to design a collision avoidance method that avoids deadlocks, where the opinion represents the preference of the robot to move left or right; nevertheless,

this approach does not guarantee collision avoidance.

3.3.2 Problem formulation

Consider a robot navigating in an environment populated with $\mathsf{N} > 0$ agents. The robot follows single integrator holonomic dynamics given by

$$\dot{\mathbf{p}}_r = \mathbf{v}_r^{\mathsf{pre}}, \tag{3.47}$$

where $\mathbf{p}_r \in \mathbb{R}^2$ is the position of the robot and $\mathbf{v}_r^{\mathsf{pre}} \in \mathbb{R}^2$ is a prescribed velocity command generated by a higher-level motion planner. On the other hand, each agent is associated with an index $i \in \{1, \ldots, \mathsf{N}\}$, and has a position $\mathbf{p}_i \in \mathbb{R}^2$ and a velocity $\mathbf{v}_i \in \mathbb{R}^2$. The robot can only sense the agents that are within a disc centered in the robot position and of perception radius $r_p > 0$

$$\mathsf{D}_p(\mathbf{p}_r, r_p) = \{\mathbf{p} \mid ||\mathbf{p} - \mathbf{p}_r|| < r_p\}, \tag{3.48}$$

where $\mathbf{p} \in \mathbb{R}^2$. Therefore, $\mathcal{N} = \{i \mid \mathbf{p}_i \in \mathsf{D}_p(\mathbf{p}_r, r_p)\}$ is defined as the set of neighboring agents of the robot. Our goal is to develop an algorithm that avoids collisions between the robot and the agents. To that end, we define the set of locations that lead to collisions as a disc centered in the position of the robot and security radius $r_s > 0$

$$\mathsf{D}_c(\mathbf{p}_r, r_s) = \{\mathbf{p} \mid ||\mathbf{p} - \mathbf{p}_r|| < r_s\}, \tag{3.49}$$

where r_s is given by the geometry of the robot and safety requirements. Agents are also characterized by a collision radius $r_i > 0$. The degree of cooperation of agent i with respect to the robot is modeled through a continuous variable $\alpha_i \in [0, 1]$, where $\alpha_i = 1$ means that the agent is fully cooperative with the robot (i.e., agent i will do all the efforts at its hands to avoid collision), whereas $\alpha_i = 0$ means that the agent is completely non-cooperative (i.e., agent i will ignore the robot and continue its movement without any collision avoidance effort). We assume that agents are not competitive, i.e., agents do not try to force collision with the robot as in, e.g., pursuit-evasion [Chung et al., 2011] or herding [Sebastián et al., 2022a] problems. As in realistic mixed cooperative/non-cooperative environments, the degree of cooperation of the agents is unknown.

To model collisions, we exploit the concept of Velocity Obstacle [Fiorini and Shiller, 1993, Fiorini and Shiller, 1998]. Given a certain time instant $t \geq 0$, the Velocity Obstacle of the robot and agent i is the set of velocities that can lead to a collision

$$\mathsf{VO}_i = \{\mathbf{v}_r \mid \exists t \in [0, \tau] \text{ s.t. } t\mathbf{v}_r \in \mathsf{D}_c(\mathbf{p}_i - \mathbf{p}_r, r_s + r_i)\}. \tag{3.50}$$

In VO_i, $\tau > 0$ is the time horizon to check collisions between the robot and agent i happens for any velocity $\mathbf{v}_r \in \mathbb{R}^2$. We make use of VO_i to develop an optimization program that uses the adapted degree of cooperation to compute a velocity $\mathbf{v}_r^*$ as close as possible to $\mathbf{v}_r^{\mathsf{pre}}$ that guarantees collision avoidance.

To adapt to the unknown degree of cooperation, we rely on the concept of nonlinear opinion dynamics [Bizyaeva et al., 2022]. Opinion dynamics emerge as a means of modeling

interactive behaviors where nodes in a network can choose among a discrete set of options regarding one or more topics. Let $x_j \in [-1, 1]$ be the opinion that node j has about a certain topic. Typically, $x_j = 0$ represents neutrality whereas $x_j = \{-1, 1\}$ represent extreme opinions over a topic with two possible opinions. By interacting with other nodes, node j can evolve its opinion. For a topic with two opinions, one possible nonlinear opinion dynamics design is

$$\dot{x}_j = -d_j x_j + g(x_j) f\left(a_j x_j + \sum_{k=1, k\neq j}^{\mathsf{M}} c_{jk} x_k\right) + b_j. \tag{3.51}$$

In the equation above, $d_j > 0$ tunes how fast the current opinion vanishes with time; $b_j \in \mathbb{R}$ is a bias that models inherent priorities that the node j has on the topic; $g(x_j)$ is an attention term that evolves with time and weights the importance of the influence of other nodes; $a_j > 0, c_{jk} > 0$ are consensus-like weights that model the exchange of opinions among the $\mathsf{M} > 0$ nodes of the network; and $f(\bullet)$ is a nonlinear function, such as tanh or $\mathsf{sigmoid}$, that bounds the influence of the interactions. More importantly, the nonlinear function in the opinion dynamics enforces a bifurcation phenomenon. The bifurcation phenomenon is characterized by the appearance of a pitchfork bifurcation in the evolution of the state variable x_j that leads to two non-zero stable equilibrium, each of them associated to one of the opinions. Whether the state evolves to one or another depends on the interactions between nodes. For more details on this, we refer the reader to [Bizyaeva et al., 2022].

We exploit nonlinear opinion dynamics to design a novel adaptive law that adjusts the unknown degree of cooperation of the agents in real-time, using only relative position and velocity measurements from onboard sensors and without the need of any communication infrastructure, prior knowledge on their degree of cooperation or other unfeasible assumptions. As a practical note, all the expressions that involve continuous-time dynamic systems are implemented using an Euler forward discretization scheme, with sample time $T > 0$. We select $T = 0.05$s to match the real-time operation of the onboard sensing and computation capabilities of the robot.

3.3.3 Optimal collision avoidance for unknown degrees of cooperation

We propose AVOCADO to address the robot collision avoidance problem in multi-agent dynamic environments. The first step towards deriving AVOCADO is to consider the geometry defined by VO_i^τ for all $i \in \{1, \ldots, \mathsf{N}\}$. Let assume that the relative velocity $\mathbf{v}_r^{\mathsf{pre}} - \mathbf{v}_i \in \mathsf{VO}_i^\tau$ for some agent i. This means that, if robot and agent i follow $\mathbf{v}_r^{\mathsf{pre}}$ and $\mathbf{v}_i$ respectively, then collision will happen at any time in $[0, \tau]$. The velocity $\mathbf{v}_r^{\mathsf{pre}}$ is provided by some higher-level planner, while $\mathbf{v}_i$ is known from onboard sensor measurements. The latter can be measured or estimated in real-time from onboard cameras or LiDAR [Eppenberger et al., 2020, Dong et al., 2020]. Since cooperation of agent i cannot be assumed, then the robot needs to choose a new velocity $\mathbf{v}_r^*$ to avoid the collision.

Let $\partial\mathsf{VO}_i^\tau$ denote the boundary of the velocity obstacle set VO_i^τ. Then, the minimum

change in velocity that makes $\mathbf{v}_r^{\mathsf{pre}} - \mathbf{v}_i \notin \mathsf{VO}_i^{\tau}$ is given by the vector $\mathbf{u}_i$

$$\mathbf{u}_i = \underset{\mathbf{v}_r \in \partial \mathsf{VO}_i}{\arg\min} ||\mathbf{v}_r - (\mathbf{v}_r^{\mathsf{pre}} - \mathbf{v}_i)|| - (\mathbf{v}_r^{\mathsf{pre}} - \mathbf{v}_i) \tag{3.52}$$

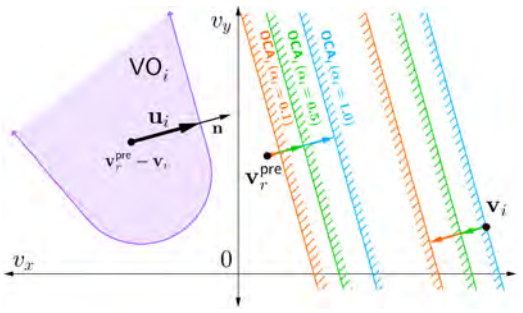


Figure 3.5: Geometry of VO_i (purple) and admissible velocities' regions associated to different degrees of cooperation (orange, green, blue). AVOCADO selects the closest velocity to the desired one that is inside all the admissible velocity sets.

Figure 3.5 depicts the geometrical reasoning behind Eq. (3.52). Given the Euclidean space of 2D velocities, vector $\mathbf{u}_i$ is a perpendicular vector to the boundary of the velocity obstacle set and with minimum magnitude. To ensure collision avoidance, the robot and agent i together must, at least, exert vector $\mathbf{u}_i$. Let α_i be the degree of cooperation of agent i with respect to the robot such that agent i exerts the α_i part of $\mathbf{u}_i$. Then, to ensure collision avoidance, the robot has to exert, at least, the $(1-\alpha_i)$ part of $\mathbf{u}_i$, which defines a constraint set of admissible velocities

$$\mathsf{OCA}_i = \{\mathbf{v}_r \mid (\mathbf{v}_r - (\mathbf{v}_r^{\mathsf{pre}} + (1-\alpha_i)\mathbf{u}_i)) \cdot \mathbf{n} \geq 0\}, \tag{3.53}$$

where $\mathbf{n}$ is the normal to $\partial\mathsf{VO}_i$ at the point given by the head of vector $\mathbf{u}_i$. The set OCA_i defines a half-space constraint of admissible velocities for the robot to avoid collision. As depicted in Fig. 3.5, the greater the value of α_i, the more cooperative is agent i and therefore the less severe the restriction on the admissible velocities for the robot. Note that $\alpha_i = 0.5$ corresponds to the ORCA algorithm [Van Den Berg et al., 2011], which corresponds to perfect reciprocity between the robot and agent i.

Given OCA_i for all $i \in \{1, \ldots, \mathsf{N}\}$, AVOCADO selects the closest velocity to $\mathbf{v}_r^{\mathsf{pre}}$ that respects all the constraints sets. Formally, this is given by the following linear program

$$\mathbf{v}_r^* = \underset{\mathbf{v}_r}{\arg\min} \quad ||\mathbf{v}_r^{\mathsf{pre}} - \mathbf{v}_r|| \tag{3.54a}$$

$$s.t. \quad \mathbf{v}_r \in \mathsf{OCA}_i \quad \forall i \in \mathcal{N}. \tag{3.54b}$$

We remark that (3.54) is a natural extension of ORCA to settings where the degree of cooperation of the agents is not necessarily reciprocal. The challenge, in this sense, is how to estimate α_i for all perceived agents $i \in \mathcal{N}$ from local sensing. Next, we describe how AVOCADO exploits opinion dynamics to adapt online to the degree of cooperation of each agent.

3.3.4 Nonlinear opinion dynamics adaptive law

To derive the adaptive law based on nonlinear opinion dynamics, we firstly define, for convenience, the following change of coordinates:

$$x_i = 2\alpha_i - 1 \Leftrightarrow \alpha_i = \frac{x_i + 1}{2}. \tag{3.55}$$

The purpose of this change is to shift the degree of cooperation from the interval $[0, 1]$ to the interval $[-1, 1]$, which better suits the definition of nonlinear opinion dynamics provided in Eq. (3.51). From now on, we refer to x_i as the shifted degree of cooperation of agent i with respect to the robot.

The (shifted) degree of cooperation associated to each agent is a quantity that only involves a pair-wise interaction between the robot and the agent. This motivates the following design for the nonlinear opinion dynamics adaptive law

$$\dot{x}_i = -d_i x_i + d_i A_i \tanh\left(a_i x_i + c_i y_i\right) + b_i. \tag{3.56}$$

Compared to (3.51), Eq. (3.56) simplifies the consensus term to a weighted sum of the current shifted degree of cooperation x_i and a quantity y_i that will be defined later and which represents the opinion would the agent i have about x_i. Ideally, the consensus term would include not only y_i, but also all the other agents. However, this would require some sort of communication infrastructure or k-hop neighboring dependence that is not feasible in a real setting that only depends on onboard sensing. Therefore, we assume that the opinion of the robot on the degree of cooperation of agent i is a pair-wise interaction isolated from other agents. In this sense, the design of (3.56) manifests a fundamental difference with respect to existing nonlinear opinion dynamics formulations, tailored towards a feasible implementation in a real robotic platform. The quantities a_i, b_i, c_i, d_i are gains with similar meaning to those in Eq. (3.51) and which parameterized the adaptive law. For instance, $b_i = -0.5$ represents some inductive bias on the degree of cooperation of agent i towards not cooperating with the robot. As nonlinear function, we choose $\tanh$ to respect the domain of x_i. Finally, A_i represents the attention, and is a variable that dynamically evolves with time, in contrast to the algebraic relationship $g(x_j)$ formulated in Eq. (3.51). As depicted in Fig. 3.6, the idea is to design and tune (3.56) such that, when the agent i is close to the robot and does not collaborate to avoid the collision, the (shifted) degree of cooperation associated to agent i evolves such that the robot takes the responsibility. On the other hand, if agent i modifies its velocity to avoid collision, then the (shifted) degree of cooperation evolves to enable the robot a more selfish behavior. The attention term A_i appears scaled in (3.56) by gain d_i. The reason behind this scaling will be explained later, in Section 3.3.5.

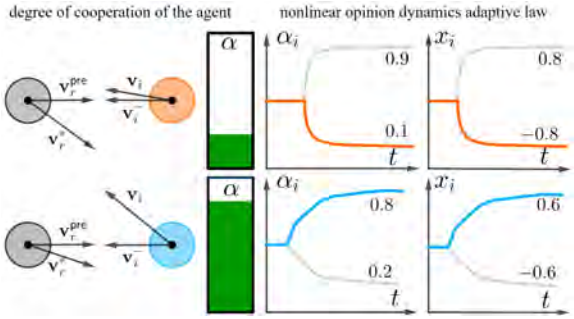


Figure 3.6: Evolution of the nonlinear opinion dynamics adaptive law for a non-cooperative (top, orange) or a cooperative (bottom, blue) agent.

There are two variables in Eq. (3.56) that are not readily available at the robot. The first one is the attention, which should capture the importance of adapting the shifted degree of cooperation or if, otherwise, the shifted degree of cooperation must not change. The second quantity is y_i, which a priori depends on the information at agent i. We will later prove how the robot can estimate y_i from just onboard sensor observations. Now, we

focus on the attention A_i.

The goal is to design an attention mechanism for the nonlinear opinion dynamics adaptive law that evolves such that the shifted degree of cooperation changes when the robot and agent i are close to each other, whereas it remains constant if the robot and the agent are far. The attention A_i is modeled as a dynamic quantity that evolves with time

$$\dot{A}_i = -\delta_i A_i + (1-\delta_i)\mathsf{tanh}(\kappa_i \tau_i^{-1}), \tag{3.57}$$

where $\kappa_i > 0$ and $\delta_i \in [0,1)$ are gains and $\tau_i > 0$ is the expected collision time. The dynamics in (3.57) evolve such that $A_i = 0$ when the robot and agent i are far from each other, and A_i tends to 1 when the robot and agent i are close to collision. To compute the time to collision, we exploit the geometry of the problem.

A potential collision between the robot and agent i happens at the intersection between (i) a circumference of radius $R = r_s + r_i$ and center $\mathbf{p}_i$ and (ii) the line $\mathbf{v}_r^{\mathsf{pre}} - \mathbf{v}_r$ passing through point $\mathbf{p}_r$. Let $\mathbf{p}_r = (p_r^x, p_r^y)$ and $\mathbf{p}_i = (p_i^x, p_i^y)$. The equation of the circumference with center $\mathbf{p}_i$ and radius R is

$$(p^x - p_i^x)^2 + (p^y - p_i^y)^2 = R^2, \tag{3.58}$$

where the circumference is parameterized by $\mathbf{p} = (p^x, p^y)$. Let $\tilde{\mathbf{v}}_i = \mathbf{v}_r^{\mathsf{pre}} - \mathbf{v}_i = (\tilde{v}_i^x, \tilde{v}_i^y)$. Then, the intersection between (i) the circumference defined in (3.58) and (ii) a line of slope given by vector $\tilde{\mathbf{v}}_i$ and point $\mathbf{p}_r$ happens at:

$$(p_r^x + \tau_i \tilde{v}_i^x - p_i^x)^2 + (p_r^y + \tau_i \tilde{v}_i^y - p_i^y)^2 = R^2. \tag{3.59}$$

The above equation leads to the second order equation

$$\beta_1 \tau_i^2 + \beta_2 \tau_i + \beta_3 = 0 \tag{3.60}$$

with $\beta_1 = (\tilde{v}_i^x)^2 + (\tilde{v}_i^y)^2$, $\beta_2 = 2\tilde{v}_i^x(p_r^x - p_i^x) + 2\tilde{v}_i^y(p_r^y - p_i^y)$ and $\beta_3 = R^2 - (p_r^x - p_i^x)^2 - (p_r^y - p_i^y)^2$. The solutions of the quadratic equation can be the following:

1. if $\beta_2^2 < 4\beta_1\beta_3$, then there is no intersection and, therefore, no potential collision. As a consequence, we set $\tau_i = \infty$.
2. if $\beta_2^2 = 4\beta_1\beta_3$, then the solution is given by a double root $\tau_i = \frac{-\beta_2}{2\beta_1}$. Otherwise, if $\beta_2^2 > 4\beta_1\beta_3$, then there are two real roots. In both cases, the following reasoning holds: (i) if both roots are positive, then we take the minimum among them; (ii) if both roots are negative, then there is no collision and $\tau_i = \infty$; (iii) if one root is positive and one root is negative, then collision has already happened and, therefore, motion is terminated.

Note that τ_i is inexpensive to compute and enables time-to-collision checking in real-time.

The next step is to define y_i and develop a method to compute it from sensor observations. According to the definition of the nonlinear opinion dynamics in Eq. (3.56), y_i represents the opinion that agent i has on the value of the shifted degree of cooperation

x_i. Recalling the reasoning behind the nonlinear opinion dynamics design in (3.51), the consensus term $a_i x_i + c_i y_i$ aims at fusing the values that the robot and agent i have on the degree of cooperation x_i. In the case of the robot, this is simply the current value of x_i; in the case of agent i, this is precisely the the opinion that agent i has on the value of the shifted degree of cooperation x_i. In typical opinion dynamics formulations it is assumed that these values can be communicated. This is not the case in our problem setting, so we need to recover y_i using an estimation method that relies solely on onboard sensing.

Figure 3.7 depicts the geometry of the problem. At a certain instant, the robot has a predefined velocity $\mathbf{v}_r^{\mathsf{pre}}$ but executes $\mathbf{v}_r^*$ to avoid collision. The difference between both velocities is $\mathbf{v}_r^* - \mathbf{v}_r^{\mathsf{pre}} = (1-\alpha_i)\mathbf{u}_i$ due to constraint (3.54b) in (3.54). On the other hand, agent i executes $\mathbf{v}_i$, and executed $\mathbf{v}_i^-$ in the previous instant, where $\mathbf{v}_i^- \in \mathbb{R}^2$ denotes the velocity of agent i sensed by the robot at the previous time instant. Since the robot has been in charge of exerting $(1-\alpha_i)\mathbf{u}_i$, the remaining part of $\mathbf{u}_i$ to avoid collision must be exerted by agent i, i.e., $\alpha_i \mathbf{u}_i$. Recall that $\mathbf{u}_i$ represents a change in the velocity. Henceforth, the remaining part of $\alpha_i \mathbf{u}_i$ is exerted through the change in velocity of agent i, namely, $\Delta\mathbf{v}_i = \mathbf{v}_i - \mathbf{v}_i^-$. However, in general, $\Delta\mathbf{v}_i \cdot \alpha_i\mathbf{u}_i \neq 0$, i.e., vectors $\Delta\mathbf{v}_i$ and $\alpha_i\mathbf{u}_i$ are not aligned. This means that it is the component of $\Delta\mathbf{v}_i$ parallel to $\alpha_i\mathbf{u}_i$ the one that exerts the change in velocities to avoid collision. Therefore, if we compute the magnitude of vector component of $\Delta\mathbf{v}_i$ parallel to $\mathbf{u}_i$ and compare it to the magnitude of $\mathbf{u}_i$, then we can obtain α_i.

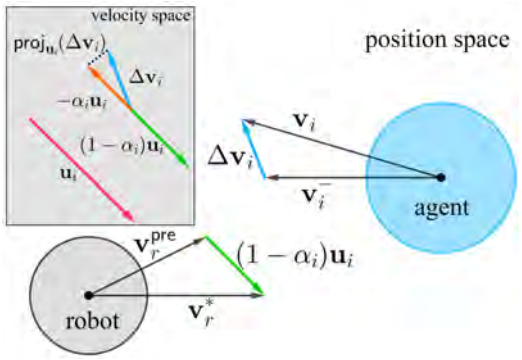


Figure 3.7: Geometry behind the projection estimator for y_i.

These insights motivate the following projection estimator of y_i:

$$y_i = \tanh\left(\varepsilon\left(\frac{||\text{proj}(\Delta\mathbf{v}_i, \mathbf{u}_i)||}{||\mathbf{u}_i||} - \frac{1}{2}\right)\right). \tag{3.61}$$

The projection operator $\text{proj}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}\cdot\mathbf{b}}{\mathbf{b}\cdot\mathbf{b}}\mathbf{b}$ projects the vector $\Delta\mathbf{v}_i$ on vector $\mathbf{u}_i$. Furthermore, by normalizing with the norm $||\mathbf{u}_i||$, the estimate of α_i, $\hat{\alpha}_i$, is extracted. The other operations translate the estimated degree of cooperation to the estimated shifted degree of cooperation y_i, where $\varepsilon > 0$ tunes how sensitive is the tanh function to changes in $\Delta\mathbf{v}_i$. In this sense, we remark that Eq. (3.61) is not a perfect estimator since it only relies on the sensor observations of the change in velocities $\Delta\mathbf{v}_i$. This motivates the use of the function tanh, which simultaneously restricts the domain of the estimation to $[-1, 1]$ and allows a fast evolution of the estimated shifted degree of cooperation. For instance, if the robot observes that agent i moves straight to it and $\Delta\mathbf{v}_i \approx 0$, then this means that $y_i \approx -1$ and the robot estimates that agent i is not cooperating.

3.3.5 Nonlinear opinion dynamics adaptive law tuning

To achieve fast and effective adaptation to the unknown degree of cooperation of the agent, the nonlinear opinion dynamics adaptive law of AVOCADO must be appropriately parameterized.

First, we examine the stability properties of the adaptive law and the conditions under which bifurcation, i.e., instability of the initially stable equilibrium point of Eq. (3.56), happens. To do so, we use the expression $\text{sech}(x) = 1/\cosh(x)$.

Proposition 8. *Let $d_i > 0$, $a_i > 0$ and $b_i \in [-1, 1]$. Then, $x_i = b_i/d_i$ is an unstable equilibrium point of (3.56) if $A_i > 1/\left(a_i sech^2\left(a_i \frac{b_i}{d_i}\right)\right)$.*

Proof. Initially, $A_i = 0$, $y_i = 0$ and $\dot{x}_i = 0$. Therefore, the robot has an initial opinion $x_i = b_i/d_i$ that is an equilibrium of the nonlinear opinion dynamics adaptive law in Eq. (3.56). Nevertheless, whether this equilibrium is stable or unstable depends on the parameters of the adaptive law. Let

$$\lambda = -d_i + d_i A_i a_i \text{sech}^2\left(a_i \frac{b_i}{d_i}\right) \tag{3.62}$$

be the unique eigenvalue of (3.56) after linearization at $x_i = b_i/d_i$. Equilibrium becomes unstable when $\lambda > 0$. Assuming that a_i, d_i and b_i are fixed, and taking into account that $\text{sech}(x) \in (0, 1]$ in all its domain, $x_i = b_i/d_i$ becomes an unstable equilibrium when $A_i > 1/\left(a_i \text{sech}^2\left(a_i \frac{b_i}{d_i}\right)\right)$. □

Corollary 4. *If $b_i = 0$, then $sech^2\left(a_i \frac{b_i}{d_i}\right) = 1$ and the condition on the attention level reduces to $A_i > 1/a_i$. Since $sech^2\left(a_i \frac{b_i}{d_i}\right) \in (0, 1)$, $A_i > 1/a_i$ is an upper bound on the attention value beyond which the equilibrium $x_i = b_i/d_i$ becomes an unstable equilibrium and, therefore, $A_i > 1/a_i$ is a sufficient condition for $x_i = b_i/d_i$ being an unstable equilibrium.*

When the robot and agent i are far from each other, the dynamics of the attention in (3.57) are designed such that $A_i = 0$. When the robot and agent i approach each other in a trajectory of potential collisions, $\mathsf{tanh}(\kappa_i \tau_i^{-1}) > 0$ and attention evolves towards $A_i > 0$, with a speed determined by δ_i. Since the attention dynamics in (3.57) is a linear system with bounded input, the attention is stable with equilibrium given by $A_i^{\mathsf{eq}} = \mathsf{tanh}(\kappa_i (\tau_i^{\mathsf{eq}})^{-1})$. Therefore, κ_i must be large enough to ensure that $A_i > \frac{1}{a_i}$ when the time to collision τ_i is large enough to avoid collision. Moreover, since $A_i \in [0, 1]$, a_i must be designed such that $\frac{1}{a_i} < 1$ to ensure that the condition $A_i > \frac{1}{a_i}$ is possible for some τ_i.

Now, we examine the stability properties of the nonlinear opinion dynamics adaptive law in Eq. (3.56) after bifurcation.

Proposition 9. *Let, $d_i > 0$, $a_i > 0$ and $b_i \in [-1, 1]$. Assume that $A_i > \frac{1}{a_i}$ holds. Then, the nonlinear opinion dynamics adaptive law in Eq. (3.56) exhibits two stable equilibrium points, given by the following expression:*

$$x_i^{\mathsf{eq}} = A_i \mathsf{tanh}(a_i x_i^{\mathsf{eq}} + c_i y_i) + \frac{b_i}{d_i}. \tag{3.63}$$

Besides, $|x_i^{\mathsf{eq}}| \in \left[\frac{|b_i|}{d_i} - 1, \frac{|b_i|}{d_i} + 1\right]$.

Proof. From the assumption that $A_i > \frac{1}{a_i}$ holds, in the transition $A_i = \frac{1}{a_i}$ bifurcation happens [Bizyaeva et al., 2022]. The dynamics in Eq, (3.56) are bounded due to the term $-d_i x_i$, so equilibrium(s) point(s) exist(s) after bifurcation. By enforcing the equilibrium condition $\dot{x}_i = 0$ in Eq. (3.56), Eq. (3.63) is obtained. We recall that $A_i \in [0, 1]$ and $|\mathsf{tanh}(\bullet)| \in [0, 1]$. Therefore, the absolute value of the equilibrium $|x_i^{\mathsf{eq}}|$ is bounded and such that $|x_i^{\mathsf{eq}}| \in \left[\frac{|b_i|}{d_i} - 1, \frac{|b_i|}{d_i} + 1\right]$. □

If there is no bias, $b_i = 0$ and $x_i^{\mathsf{eq}} \in [-1, 1]$; otherwise, the bias introduces a shift in the domain that naturally encodes a priori knowledge on the degree of cooperation of the agent. Proposition 9 motivates the use of the scaling factor d_i in the attention A_i: in the absence of this factor, the expression of the equilibrium points is $x_i^{\mathsf{eq}} = \frac{A_i \mathsf{tanh}(a_i x_i^{\mathsf{eq}} + c_i y_i) + b_i}{d_i}$, leading to an asymmetrical domain for $|x_i^{\mathsf{eq}}|$ with respect to b_i/d_i, i.e., not symmetrical with respect to the initial equilibrium before bifurcation. This asymmetry would derive in an intrinsic bias towards any of the two subsequent equilibrium points, which is an undesired effect that we overcome by including the scaling factor d_i.

Finally, c_i shall be designed to guarantee that the nonlinear term $\mathsf{tanh}(\bullet)$ in the adaptive law is enough sensitive to the behavior of agent i, reacting fast enough to avoid collisions. On the other hand, ε in (3.61) tunes how sensitive is the estimator of y_i to changes in the speed of agent i.

In summary, the nonlinear opinion dynamics adaptive law depends on six parameters: (i) $\varepsilon > 0$ determines how sensitive is the estimator of y_i and should be chosen large enough to ensure $y_i \approx 1$ with enough time to avoid collision but small enough to avoid noisy estimations; (ii) $\kappa_i > 0$ and $\delta_i \in [0, 1)$ determines how fast the adaptive law approaches the bifurcation point, and therefore establishes when the robot decides its opinion about the degree of cooperation of agent i; (iii) $d_i > 0$ sets the convergence speed of the dynamics of the adaptive law, associated to the forgetting factor $-d_i x_i$; (iv) a_i determines the attention level necessary to reach the bifurcation, taking into account the value of d_i; (v) b_i is a bias that encodes potential prior knowledge on the (shifted) degree of cooperation of agent i; (vi) c_i weights the importance of the estimate y_i compared to the current value of $a_i x_i$ such that, if $a_i >> c_i$, the robot tends to preserve its opinion over the (shifted) degree of cooperation, or heavily relies on the estimator if $c_i >> a_i$.

3.3.6 Exploiting attention to overcome deadlocks

The previous sections address the problem of unknown degree of cooperation in collision avoidance. However, there is an additional issue, typical in geometrical methods like ORCA, that is characterized by navigation deadlocks under symmetrical configurations [Battisti and Muradore, 2020]. To overcome them, AVOCADO exploits attention A_i.

Our solution consists in injecting some small noise to the perceived velocity of agent i

when the agent is far from the robot, decreasing the noise to zero when the robot and the agent are close to a potential collision. When the robot and the agent are far from each other, the impact of noise is negligible because attention $A_i \approx 0$ and bifurcation does not happen. This preserves the neutrality of the opinion on the (shifted) degree of cooperation. When the robot and the agent are close to each other, noise is removed, so the performance guarantees on the adaptive law and the collision avoidance are preserved.

Mathematically speaking, we reformulate Eq. (3.52) as

$$\mathbf{u}_i = \underset{\mathbf{v}_r \in \partial \mathsf{VO}_i}{\arg\min} ||\mathbf{v}_r - (\mathbf{v}_r^{\mathsf{pre}} - \mathbf{v}_i^{\mu})|| - (\mathbf{v}_r^{\mathsf{pre}} - \mathbf{v}_i^{\mu}) \text{ with } \mathbf{v}_i^{\mu} = \mathbf{v}_i + (1 - A_i)\mu(\sigma), \tag{3.64}$$

and $\mu(\sigma) \sim \mathcal{U}(-\sigma, \sigma)$ a uniformly distributed perturbation bounded by $\sigma > 0$.

In essence, our method adds a uniformly distributed noise to the velocity of agent i sensed by the robot. This noise depends on the attention mechanism developed for the nonlinear opinion dynamics adaptive law. This perturbed velocity is the one used to compute vector $\mathbf{u}_i$ and the OCA_i^{τ} constraint sets. Hence, the noise perturbs the potential symmetry between $\mathbf{v}_r^{\mathsf{pre}}$ and $\mathbf{v}_i$ only when the robot and agent i are sufficiently close to each other. The intensity of the perturbation increases when the distance between robot and agent i decreases in order to ensure that there is an effective deadlock breaking when robot and agent i approach each other. It is interesting to remark that this deadlock breaking approach can also be applied to other VO-based methods such as ORCA, since it only requires the implementation of the attention mechanism in Eq. (3.57).

AVOCADO is summarized in Algorithm 7. By combining a geometrical approach with a nonlinear opinion dynamics adaptive law, AVOCADO is able to adapt in real-time to the unknown degree of cooperation of the agents in a multi-agent setting. No communication is involved, only relying on the perceived position and velocity of the nearby agents. All the mathematical operations are inexpensive to compute, except for the optimization program in line 11. Nonetheless, since it entails a linear program, with an efficient implementation it is proven that a robot can process thousands of nearby agents in milliseconds [Van Den Berg et al., 2011]. We validate the computational simplicity of AVOCADO in Section 3.3.7.

3.3.7 Simulated results

First, we evaluate AVOCADO in simulated scenarios. In Section 3.3.10 we describe the results obtained in real settings with robots and humans.

In Section 3.3.8 we analyze the impact of the different parameters in the behavior of AVOCADO. Next, in Section 3.3.9 we conduct extensive multi-agent simulations using two navigation settings. For each setting, we compare qualitatively and quantitatively AVOCADO with existing state-of-the-art planners against other cooperative robots and non-cooperative agents. In all the simulated scenarios, we consider that robots are cooperative, i.e., they act using the same motion planner under comparison. For instance, if AVOCADO is assessed, then all robots use AVOCADO. Importantly, a robot being cooperative does not mean knowledge on the degree of cooperation nor communication exchange, but just the

Algorithm 7 AVOCADO

Parameters: $r_s, r_p > 0$, $\varepsilon > 0$, $\kappa_i > 0$, $d_i > 0$, $a_i, b_i, c_i \in \mathbb{R}$, $\sigma > 0$, $\delta_i \in [0, 1)$
for all t **do**
  Get $\mathbf{p}_r$ and $\mathbf{v}_r^{\mathsf{pre}}$ from a higher-level planner.
  Measure $\mathbf{p}_i, \mathbf{v}_i \quad \forall i \in \mathcal{N}$ from sensors and get the stored $\mathbf{v}_i^- \quad \forall i \in \mathcal{N}$.
  Update the attention level A_i using $\mathbf{p}_r$, $\mathbf{v}_r^{\mathsf{pre}}$, $\mathbf{p}_i$ and $\mathbf{v}_i$ through Eqs. (3.57) and (3.60), for all $i \in \mathcal{N}$.
  Apply Eq. (3.64) to compute $\mathbf{v}_i^{\mu} \quad \forall i \in \mathcal{N}$.
  Calculate $\mathbf{u}_i$ through Eq. (3.64).
  Estimate y_i through Eq. (3.61).
  Update the opinion on the shifted degree of cooperation using the adaptive law in Eq. (3.56).
  Build all admissible sets OCA_i through Eq. (3.53), for all $i \in \mathcal{N}$.
  Solve optimization problem (3.54) to obtain $\mathbf{v}_r^*$.
  Apply $\mathbf{v}_r^*$ and store $\mathbf{v}_i$ as $\mathbf{v}_i^- \quad \forall i \in \mathcal{N}$.
end for

robot uses a planner to avoid collisions. In contrast, agents are non-cooperative, i.e., they are blind against the robots. Non-cooperative agents resolve (3.54), but their neighbor set $\mathcal{N}_i$ never includes the robots, only other agents. This is done to ensure collision avoidance among agents but complete non-cooperation with the robots, so the behavior is more complex than a simple dynamic obstacle with a fixed trajectory. A mixed cooperative/non-cooperative navigation scenario involves cooperative robots and non-cooperative agents. Finally, robots and agents have a disc shape of radius $r_s = 0.2$m and a sensor range of $r_p = 2.5$m.

The maximum velocity, $\mathbf{v}_r^{\mathsf{max}}$, of the robots is set to 1m/s and $\mathbf{v}_i^{\mathsf{max}}$ of the agents is set to 0.75m/s. The agents have a lower velocity than the robots, allowing the robots to escape if the agents move quickly towards them. For robots using AVOCADO, $\mathbf{v}_r^{\mathsf{pre}} = \mathbf{v}_r^{\mathsf{max}} \frac{\mathbf{p}_{r,j}^* - \mathbf{p}_{r,j}^t}{||\mathbf{p}_{r,j}^* - \mathbf{p}_{r,j}^t||}$, where $\mathbf{p}_{r,j}^*$ denotes the desired goal for robot j and $\mathbf{p}_{r,j}^t$ is the position of robot j at time t. Therefore, robots using AVOCADO are not helped by any higher-level planner to choose the desired velocity.

3.3.8 Head-on scenarios

We simulate head-on scenarios to analyze the impact of the different parameters in AVOCADO.

A robot and either a cooperative robot or a non-cooperative agent face each other, where their goals are the starting position of the other. Fig. 3.8 illustrates these scenarios for different values of bias b_i (a, c, e for the non-cooperative agent; b, d, f for the cooperative robot). The robot using AVOCADO starts, in scenarios a) and b), assuming that the other robot or agent is non-cooperative with a low degree of cooperation; in scenarios c) and d), the robot initially assumes reciprocity, with a degree of cooperation

equal to 0.5; and in e) and f) the robot initially assumes a great degree of cooperation. Extreme bias values can lead to collisions, as in b) and e). In a), the robot smoothly avoids the collision with the agent, as the highly negative biased initial degree of cooperation (pessimistic robot) corresponds to reality. However, when the bias is incorrect, even if the agent is cooperative as in b), the robot takes some time to recover from the incorrect prior, resulting in shaky trajectories and even collisions in b). The opposite conflict arises when there is an initial highly positive bias in the degree of cooperation (optimistic). In case e), the robot is unable to correct the estimated degree of cooperation fast enough to realize that the agent is non-cooperative and take all the responsibility to avoid collision. Collision is avoided in case f), but the extreme bias leads to sub-optimal trajectories. Finally, a balanced bias allows the robot to react effectively to non-cooperative agents in c) and cooperative agents in d).

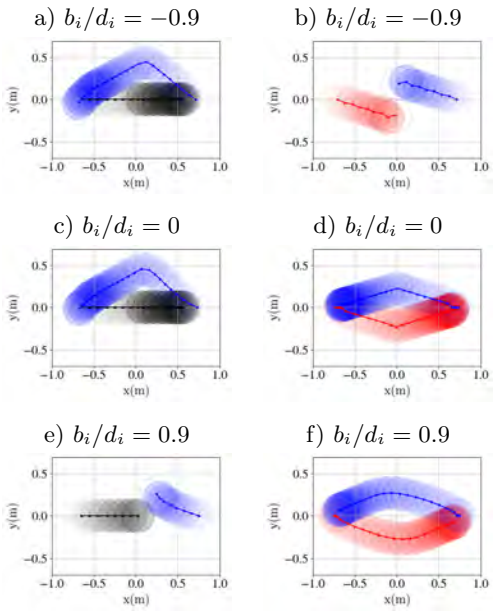


Figure 3.8: Head-on scenarios for different ratio b_i/d_i and degree of cooperation. The robot (blue) navigates using AVOCADO, whereas the agent is either non-cooperative (black, left column) or navigates using AVOCADO (red, right column). The motion of the robot and the agent is represented with disks of increasing transparency as time advances, and solid dots represent the center of the disks.

The impact of the other parameters is evaluated by studying the evolution of x_i when the parameters change. The default parameters are specified in Table 3.1. Now, we focus on the non-cooperative head-on scenario.

Parameter	a_i	b_i	d_i	κ_i	ε	δ_i	b_i
Value	0.3	0.7	2	14.15	3.22	0.57	0

Table 3.1: Default parameter values

Fig. 3.9 depicts the evolution of x_i for the different cases. When the simulation starts, $x_i = b_i/d_i$. Then, as the robot detects that agent i does not cooperate, x_i decreases, trying to reach $x_i = -1$. Finally, when the collision is avoided and $A_i = 0$, $x_i = b_i/d_i$ again. In some cases, as in Fig. 3.9e) with $\delta = 0.1$, x_i does not recover to $x_i = b_i/d_i$ because agent i leaves the sensor range of the robot before letting the nonlinear opinion dynamics to converge to $x_i = b_i/d_i$. First, regarding d_i (Eq. (3.56)), greater values imply a greater forgetting factor and therefore, a faster convergence according to the eigenvalue study from Eq. (3.62). For very large values ($d_i = 6.5$), the nonlinear opinion dynamics are too reactive to the estimate y_i and to the noise introduced (Eq. 3.64), leading to abrupt changes in x_i. Second, regarding a_i, c_i (Eq. (3.56)), since $c_i = 1 - a_i$ and $a_i \in \{0.1, 0.3, 0.6, 0.9\}$, greater values of a_i imply less reactivity against changes in the estimate y_i and, henceforth, slowest convergence of the nonlinear opinion dynamics.

Third, regarding ε (Eq. (3.61)), it is observed that beyond some value around $\varepsilon = 1$, the estimate of y_i is reactive enough to guarantee a fast convergence of the nonlinear opinion dynamics. Fourth, the behavior of κ_i (Eq. (3.57)) is similar to that of ε, in the sense that there is a point beyond which increasing κ does not provide any improvement. Fifth, the same holds for δ_i (Eq. (3.57)). Finally, regarding the ratio b_i/d_i (Eq. (3.56)), it is very interesting to observe that different values lead to different initial and final equilibrium, equal to $x_i = b_i/d_i$. In all the cases, since the agent is always non-cooperative, x_i evolves towards a low value in order to take the responsibility of avoiding the collision. The plots of Fig. 3.9 show that, in general, very low parameters' values are not desirable, as they make the opinion grow slowly and the robot to react late. Nevertheless, very large values can make the nonlinear opinion dynamics adaptive law to be too sensitive to changes in the agent and the estimate y_i, leading to a non-smooth evolution of x_i. Therefore, it is suggested to design the parameters as low as possible to guarantee collision avoidance but obtain smooth maneuvers. The nominal parameters described in Table 3.1 have been tuned using Bayesian Optimization, constrained the domain of the parameters according to the conclusions drawn from Fig. 3.9.

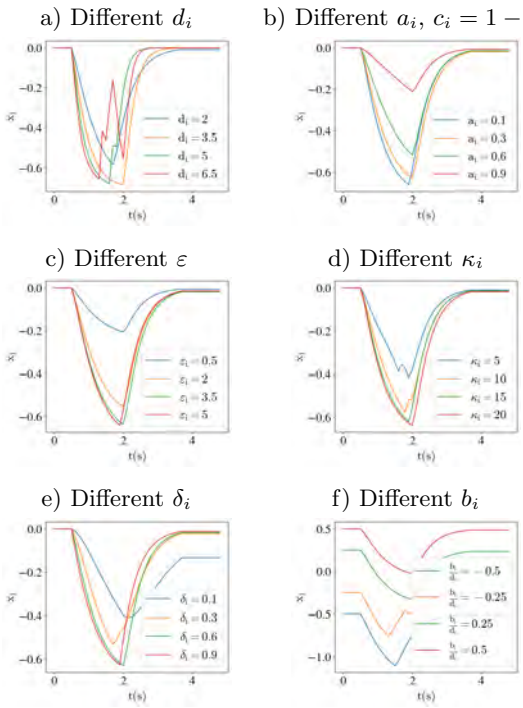


Figure 3.9: Evolution of the degree of cooperation in the head-on scenario against a non-cooperative agent, varying the value of each parameter of AVOCADO.

3.3.9 Multi-agent scenarios

After studying the influence of the different parameters in AVOCADO, we conduct a series of multi-agent simulations to evaluate the performance of AVOCADO compared to existing state-of-the-art planners. We compare AVOCADO with ORCA [Van Den Berg et al., 2011] and RVO-RL [Han et al., 2022] as baselines planners that consider reciprocal collision avoidance, and T-MPC [Mavrogiannis et al., 2022, Poddar et al., 2023] and SARL [Chen et al., 2019] as baseline planners for non-cooperative collision avoidance. The former is based on a model predictive control formulation and the latter is a neural-network-based planner trained using reinforcement learning. In this sense, SARL is retrained to satisfy the constraints in perception from the limited sensing radius of robots and agents. Moreover, we set $\sigma = 0.0001$ for AVOCADO.

We design two evaluation settings. The *circle* scenario is characterized by an evenly spaced initial position of robots and agents, forming a circle and with the goal of reaching the initial position of the opposite robot or agent. On the other hand, the *crossing* scenario

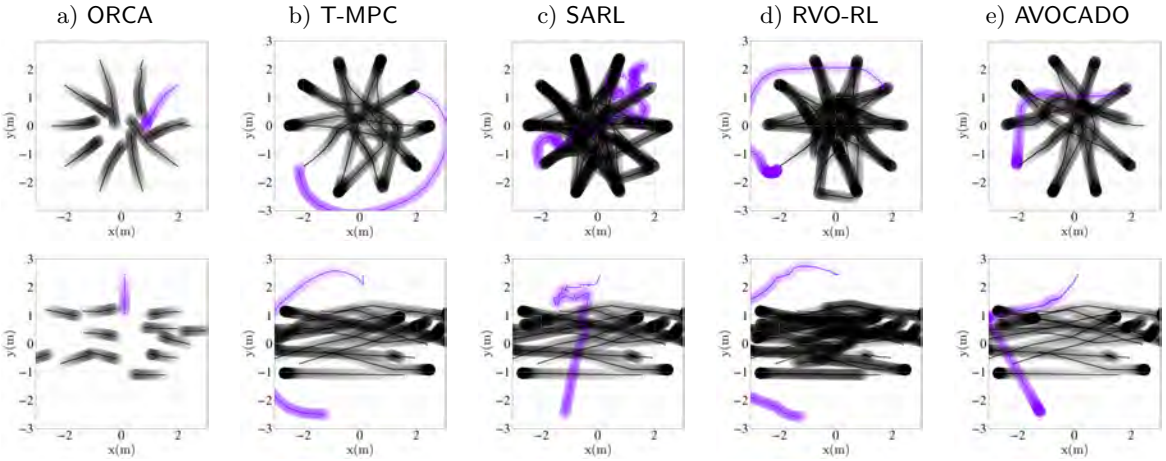


Figure 3.10: Non-cooperative *circle* (top) and *crossing* (bottom) scenarios with 10 agents and different planners. The robot is colored in purple, and the non-cooperative agents are in black. When the robot collides with an agent, it becomes transparent. The episode finishes when all the agents and the robot reach the goal or collide, or the simulation lasts more than 100s.

is characterized by a random initial position of all the agents and robots in the border of a square. Robots and agents must navigate to a individually assigned random goal in the opposite side of the square. Cooperative robots are placed in the sides oriented to one axis and non-cooperative agents are placed in the sides oriented to the other axis. This arrangement enforces that cooperative robots traverse non-cooperative agents that move perpendicularly as a traffic flow they have to cross. Besides, to ensure that robots do not take trivial motion strategies (e.g., wait until all the agents have reached the goal), once an agent reaches its goal, the goal is modified to be the initial position of the agent, so the environment is always dynamic.

We show the qualitative behavior of the robots in non-cooperative *circle* and *crossing* scenarios in Fig. 3.10. The planners address the navigation conflict at the center of the circle in two different ways. T-MPC and RVO-RL completely avoid dangerous zones by taking a detour; this might not be possible in a constrained scenario with boundaries. Among the other three approaches, ORCA is the one that takes more risks, since it assumes reciprocity in the degree of cooperation, colliding in both scenarios. SARL and AVOCADO achieve adaptation to the unknown degree of cooperation of the agents and other robots and manage to reach the goal in both scenarios; nonetheless, the movements exerted by the robots using AVOCADO are much smoother than those exerted by SARL, which is a key aspect when transferring navigation policies to real robots.

Fig. 3.11 shows examples of scenarios only populated with robots. Fig. 3.11a demonstrates the deadlocks suffered by reciprocal approaches like ORCA when symmetries are encountered. AVOCADO overcomes this problem, as seen in Fig. 3.11e, by exploiting the attention mechanism in the nonlinear opinion dynamics adaptive law. T-MPC and RVO-RL take long unnecessary detours, as they are designed for scenarios with fewer robots and to take larger safety margins. T-MPC even presents many collisions, as it is not prepared to interact with other robots, and some of the RVO-RL robots do not reach the goal before the time out of 100s. Trajectories of robots using SARL lack, again, smoothness, probably due

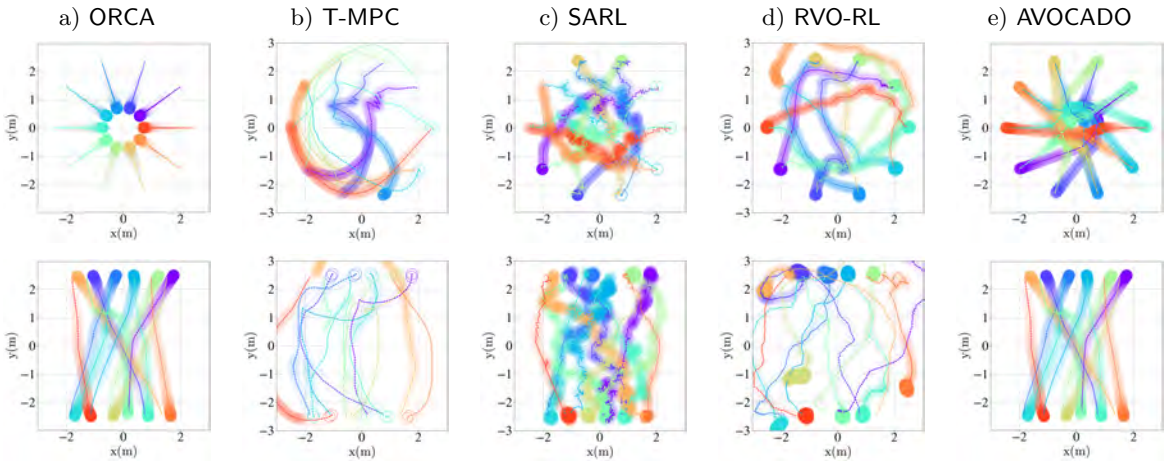


Figure 3.11: Cooperative *circle* (top) and *crossing* (bottom) scenarios with 10 robots and different planners. Each robot is depicted with a different color. The representation follows the same rules in Fig. 3.10.

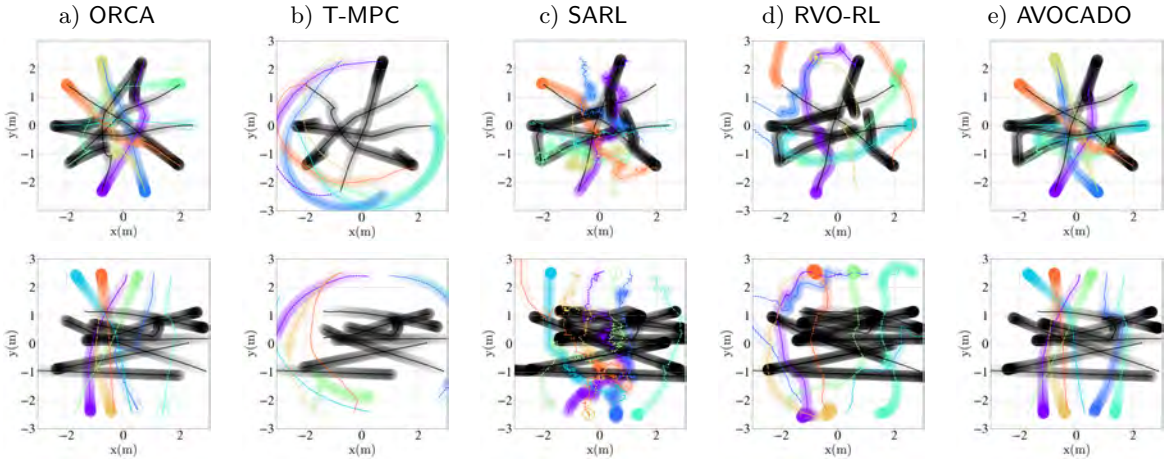


Figure 3.12: Mixed cooperative/non-cooperative *circle* (top) and *crossing* (bottom) scenario with 4 agents, 6 robots and different planners. Each robot is depicted with a different color. The representation follows the same rules in Fig. 3.10.

to a reciprocal dance problem, specially noticeable in *circle* scenarios where this leads to collisions. All the robots using AVOCADO reach their goals avoiding collisions, with short path lengths and smooth trajectories, in both scenarios.

Mixed cooperative/non-cooperative scenarios are the most challenging for most of the planners, as they assume some sort of degree of cooperation, either reciprocity or complete absence of cooperation. Fig. 3.12 provides examples of such scenarios. T-MPC and RVO-RL avoid passing through the center of the stage, as there are many non-cooperative agents. However, they encounter cooperative robots during the detour, so they are unable to avoid collisions. SARL faces again the reciprocal dance problem when evading both non-cooperative agents and other robots avoiding collisions. ORCA and AVOCADO perform similar trajectories, but AVOCADO adaptation capabilities makes it safer, avoiding all collisions unlike ORCA.

We conduct a systematic series of *circle* and *crossing* runs to extract quantitative metrics to compare all the planners, using an increasing total number of agents. The circle size of *circle* scenarios is set to $\max(2.5, \frac{2.3\mathsf{N}r_s}{\pi})$m and the square size of *crossing* scenarios is set to $1.5\mathsf{N}r_s$m. These numbers guarantee that there is enough space for the initial and final position of all robots and agents while enforcing collision conflicts. Simulation runs range from $\mathsf{N} = 10$ to $\mathsf{N} = 25$ in steps of 3. For each value of N, we test five different proportions P of cooperative robots. In particular, we set the number of cooperative robots equal to $\lceil \mathsf{PN} \rceil$, with $\mathsf{P} = \{0.01, 0.25, 0.5, 0.75, 1\}$, where $\lceil \bullet \rceil$ is the operator that rounds a real number to the closest greater integer. The others are non-cooperative agents. We run 128 random scenarios for each combination of planner, number of cooperative robots and non-cooperative agents. Each run in *circle* scenarios randomizes the identity of each agent (non-cooperative or the planner under evaluation), while each run in *crossing* scenarios randomizes the initial and goal positions of each agent. We compare the state-of-the-art planners with 4 versions of AVOCADO with the following parameters:

- AVOCADO_1: Table 3.1.
- AVOCADO_2: Table 3.1 but $d_i = 5$.
- AVOCADO_3: Table 3.1 but $b_i = 1$ ($\frac{b_i}{d_i} = 0.5$).
- AVOCADO_4: Table 3.1 but $b_i = -1$ ($\frac{b_i}{d_i} = -0.5$).

We define the success rate as a evaluation metric. Let M be the number of runs; $\text{succ}_{j,k} \in \{0, 1\}$ the indicator that is equal to 1 if $||\mathbf{p}_{r,j} - \mathbf{p}^*_{r,j}|| < \xi$ at the end of run k or 0 otherwise, where $\mathbf{p}_{r,j}$ is the position of robot j, $\mathbf{p}^*_{r,j}$ is the desired goal of robot j, and $\xi > 0$ is a small tolerance. Then, the success rate is defined as

$$\frac{1}{\mathsf{MN}} \sum_{k=1}^{\mathsf{M}} \sum_{j=1}^{\mathsf{N}} \text{succ}_{j,k}.$$

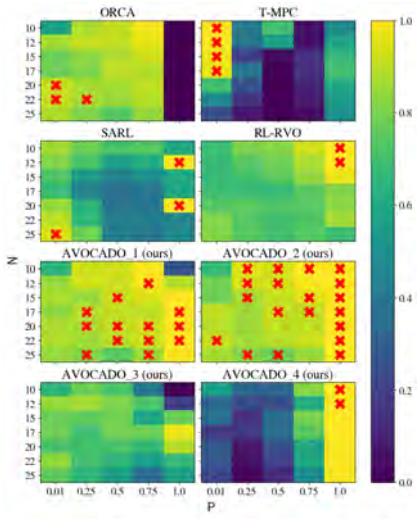


Figure 3.13: Success rates of robots with the different planners in *circle* scenarios. Red crosses mark the best planner.

The success rates are in Fig. 3.13 and Fig. 3.14 for *circle* and *crossing* scenarios, respectively. AVOCADO outperforms all other approaches in success rate. AVOCADO_2, with a higher value of d_i than AVOCADO_2, has a greater success rate in *circle* scenarios where the number of cooperative robots is greater, as it is more sensitive to the introduced noise that breaks symmetries. AVOCADO_1 is more stable and has a better performance in partially cooperative scenarios. AVOCADO_3 presents a worse success rate due to the bias, that assumes low degrees of cooperation when it is often not true. In this sense, it is very interesting to see how AVOCADO_4, with a bias towards great degrees of cooperation, presents the worst results among the AVOCADO versions in

Fig. 3.13 unless in fully-cooperative cases. However, AVOCADO_4 achieves the best success rates in Fig.3.14. This is probably due to the fact that, in *crossing* scenarios, the robot first faces non-cooperative perpendicular traffic, where being cautious is desirable; meanwhile, in *circle* scenarios the robot faces agents coming in their same direction, leading to conflicts similar to the one previously seen in Fig. 3.8b. ORCA experiences deadlocks from geometrical symmetries, so it fails in all cooperative *circle* scenarios. The performance of T-MPC degrades when the number of cooperative agents increases, since, as it is observed in [Mavrogiannis et al., 2022, Poddar et al., 2023], the method is suited for scenarios with less than 10 robots or agents. RVO-RL, tuned for fully-cooperative environments, achieves its best performance for $\mathsf{P} = 1$, decreasing the success rate as non-cooperative agents appear. SARL, due to the non-smooth behavior, obtains poor success rates.

We show in Fig. 3.15a the mean navigation time that every planner takes to reach the goal in *square* scenarios with $\mathsf{P} = 0.5$. The mean time to goal is computed taking into account only the robots that do not collide:

$$\frac{1}{\mathsf{MN}} \sum_{k=1}^{\mathsf{M}} \sum_{\substack{j=1 \\ \text{succ}_{j,k}=1}}^{\mathsf{N}} t_{j,k},$$

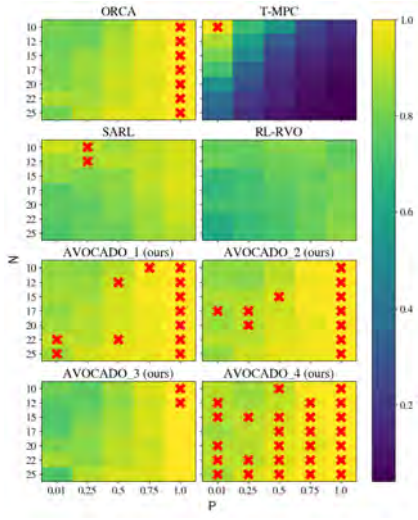
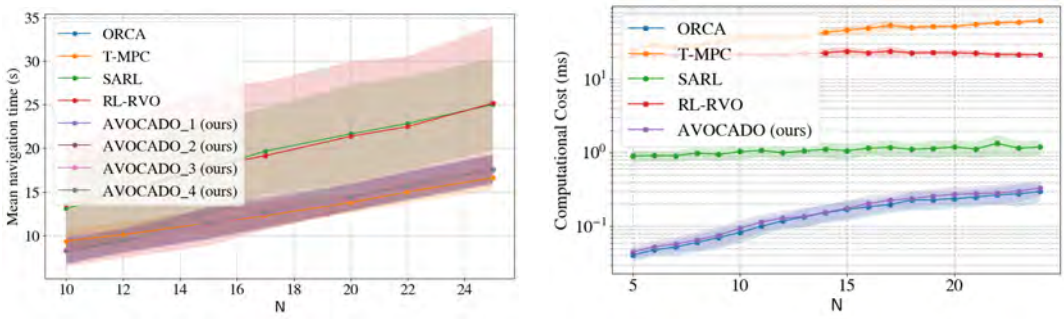


Figure 3.14: Success rates of robots with the different planners in *crossing* scenarios.

where $t_{j,k} > 0$ denotes the instant when robot j reaches its goal at run k. The results are aligned with the qualitative results depicted in Figs. 3.10, 3.11, and 3.12). RL-RVO and SARL exhibit the largest time to goal because their trajectories are irregular and present many detours. Meanwhile, the rest of the planners reach the goal in similar times. T-MPC manifests slightly lower times to reach the goal even tough the resulting trajectories are longer than those of AVOCADO or ORCA. This is due to the fact that T-MPC exerts higher velocities and it is, therefore, more risky, which explains its success rate metrics shown in Figs. 3.13 and 3.14.

Finally, regarding computation times, Fig. 3.15b collects the mean and standard deviation computation times for all the planners under comparison. The pure geometrical methods (AVOCADO and ORCA) share the same inexpensive computational cost, requiring hundreds of milliseconds to compute a solution in crowded environments. The time increases with the number of robots and agents since the number of entities under consideration grow, but this growth is linear with the number of agents and robots within the sensor range. The other methods take, by orders of magnitude, much more time to compute their navigation commands. This computational burden may prevent their use in real hardware applications with constrained resources, specially if there are other higher-level tasks that use those resources.



(a) (b)

Figure 3.15: (a) Mean and standard deviation of navigation times (in seconds) of successful robots for the different planners, gathered in 128 *square* scenarios with $\mathsf{P} = 0.5$. (b) Mean and standard deviation of the computational times (in milliseconds) for the different planners. Y-axis is in a logarithmic scale.

3.3.10 Experimental results

After evaluating AVOCADO in simulated environments, in this section we conduct experiments with real ground robots. We use three Turtlebot 2 robotic platforms that use AVOCADO, and up to three pedestrians as external non-cooperative agents. The experiments involve 19 pedestrians, 15 of them external to the project. The pedestrians, as the robots, have a fixed starting point and final goal for each experiment, but they have no instructions on how to behave and interact with the robots. In this way, the pedestrians decide by themselves their degree of cooperation with the robots and other pedestrians, their own velocity and the trajectory to follow to reach the goal, making each experiment different and unpredictable from the perspective of the robots. The robots use AVOCADO with the same parameterization as the default ones detailed in Table 3.1. The purpose is to prove that AVOCADO presents zero-shot-transfer capabilities.

We conduct the experiments in an arena of 6×6m. We used an Optitrack Prime^X 13W system of markers and 12 cameras to localize the robots and the pedestrians. We also design and implement an Extended Kalman Filter with a constant velocity assumption to track the positions and velocities of all the robots and pedestrians.

We design three representative experimental scenarios, running them using different combinations of pedestrians, leading to a total of 33 experiments. Videos of the experiments can be found in the supplementary material. The three scenarios are as follows:

- **Head-on**: two robots are placed next to each other in two corners of the arena, and two pedestrians are placed in the other two corners of the arena, each of them facing one of the robots. The goal of the robots and the pedestrians is to exchange their position with the pedestrian or robot that is in front of them.
- **Circle**: as in the simulations, pedestrians and robots are arranged in an evenly spaced circular formation, alternating robots and pedestrians. The goal of all the players is to go to the opposite side of the circle.
- **Crossing**: two robots are located in the medium point of two opposite sides of

a) Group 2 b) Group 4

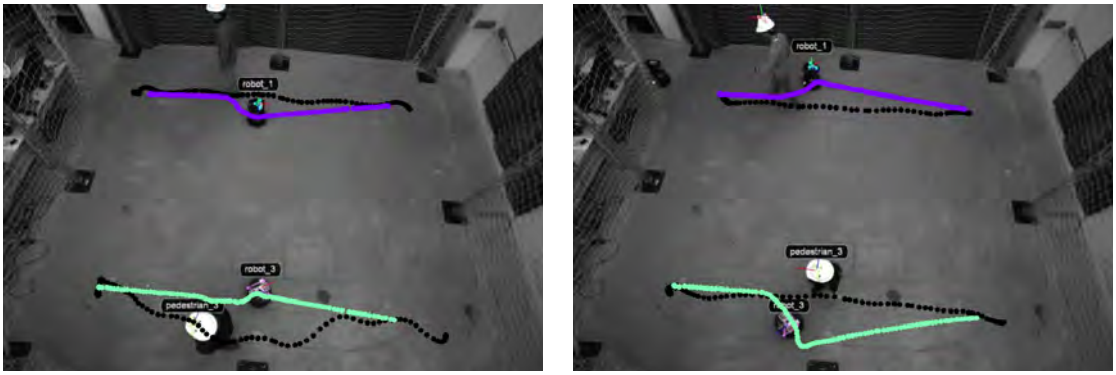


Figure 3.16: Head-on experiment, groups 2 and 4. Two robots face two pedestrians, exchanging their initial position. The pedestrians trajectories are represented in black and the robots with different colors.

the arena, facing each other. In one of the other sides we place a pedestrian and a robot, while two pedestrians are placed in the remaining side of the arena. The team robot-pedestrian and the team pedestrian-pedestrian are told to cross the arena towards the position of the other team. The goal of the first two robots is to cross the perpendicular traffic flow to exchange their position.

Fig. 3.16 shows two examples of head-on experiments. In the first experiment (a) one of the pedestrians (bottom) decides to evade the robot. AVOCADO detects that and adapts to the situation, following a straight trajectory. Reciprocal dance problem is not observed. The other pedestrian in Fig. 3.16a slightly modifies the trajectory to cooperate with the robot. Fig. 3.16b shows the same experiment with other pedestrians, that decide to not cooperate at all, demonstrating how the robots are able to take all the responsibility to do the collision avoidance maneuver.

In Fig. 3.17a, pedestrians participating in the circle experiment walked with different velocities. When the person starting in the down-right leaves the intersection, the person starting in the top-center is still starting the trajectory. The robots adapt to the different velocities and safely reach the goal with smooth trajectories. The purple robot traverses the intersection moving to its right side, as there is free space there. Fig. 3.17b depicts a situation where the human starting in the down-right and the one starting in the top-center have similar behaviors as in the previous experiment in terms of speed. The pedestrian starting in the bottom-left, however, is faster than its previous homologous. The robots accommodate to this situation by being more cooperative and more reactive (yellow and blue) or choosing a different trajectory in the space that pedestrian is leaving behind (purple).

A standard crossing scenario is represented in Fig. 3.18a. All the agents forming the traffic in the middle of the arena respect their lane. The two robots in opposite sides choose a side to cross the intersection, passing between the two nearby agents and the two far ones. The girl starting behind the purple robot overtakes it safely, through the open space that is in their left. The robot collaborates but giving her more space to do it. Fig. 3.18b, however, shows a different situation. The pedestrian behind the other pedestrian decides

a) Group 3 b) Group 5

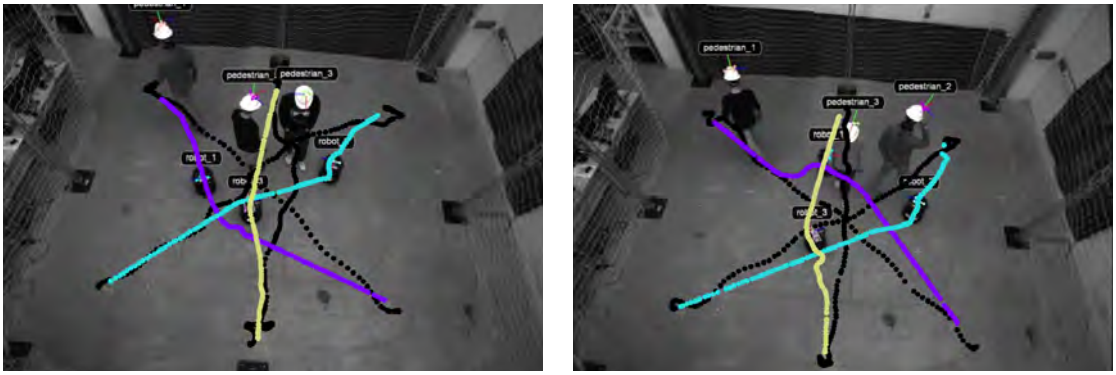


Figure 3.17: Circle experiment, groups 3 and 5. Three robots and three pedestrians, evenly spaced in a circle, exchange their positions with the opposite player.

a) Group 4 b) Group 5

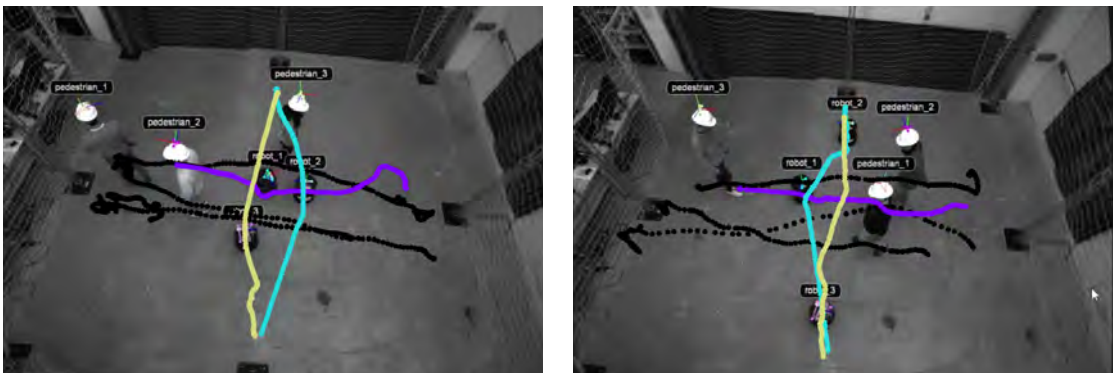


Figure 3.18: Crossing experiment, groups 4 and 5. Two robots, in opposite sides, exchange positions by crossing an intersection with one robot and three pedestrians.

to accelerate and overtake through the central lane. The purple robot has to slow down to evade a frontal collision. This delay in the central traffic makes the blue robot avoid it through the right side, and the yellow adapt to it as it arrives there.

Overall, extensive real experiments with multiple robots and humans prove that AVOCADO is effective navigation strategy for collision avoidance in mixed cooperative/non-cooperative environments. AVOCADO is a zero-shot approach since no further tuning is required to transfer the algorithm from simulations to real robots. In this sense, AVOCADO preserves the properties observed in simulations in terms of success rate, smoothness, adaptation to unknown degrees of cooperation and computational efficiency.

3.4 Conclusions

This chapter has presented a suite of novel distributed algorithms that enhance coordination and control in multi-robot systems, particularly for complex tasks like herding. These algorithms address the challenges of distributed optimization, consensus, and dissensus,

providing efficient and robust solutions for multi-robot collaboration:

- **Accelerated distributed optimization**: A novel accelerated distributed optimization method for consensus optimization has been proposed. This method combines the benefits of momentum with gradient tracking and ADMM algorithms, resulting in fast convergence, computational simplicity, and robustness. By incorporating momentum, the algorithm overcomes limitations in the convergence rate of existing ADMM-GT approaches, providing faster optimization without increasing computational or communication overhead.
- **Accelerated dynamic average consensus**: Two accelerated discrete-time dynamic average consensus protocols have been introduced. These protocols leverage a sequence of proportional consensus filters and a second-order recurrence to overcome the trade-off between convergence speed and steady-state error. The algorithms are robust against initialization errors, input noise, and communication delays, making them suitable for real-world multi-robot applications where these factors are prevalent.
- **Adaptive collision avoidance**: AVOCADO, a novel collision avoidance strategy for mixed cooperative/non-cooperative multi-agent environments, has been presented. AVOCADO utilizes a nonlinear opinion dynamics adaptive law to estimate the unknown degree of cooperation of external agents in real-time. This allows the robot to effectively avoid collisions even with non-cooperative agents, addressing a significant challenge in multi-robot navigation. The algorithm's computational efficiency and zero-shot transfer capabilities make it suitable for real-world deployment on robotic platforms.

These contributions collectively enhance the capabilities of distributed multi-robot systems, enabling efficient coordination and control in complex, dynamic environments. The algorithms presented in this chapter offer improved performance and robustness compared to existing approaches, paving the way for more sophisticated and reliable multi-robot applications. Future research directions include extending these algorithms to broader classes of distributed control problems and incorporating learning-based techniques to further enhance adaptability and performance in complex tasks. It also remains to be explored how to combine Implicit Control, presented in Chapter 2, with the techniques presented in this chapter to derive a fully distributed input-nonaffine control method multi-robot problems.

Part II

Ellipsoidal Methods for Distributed Stochastic Estimation

Chapter 4

Event-triggered and Certifiable Optimal Distributed Stochastic Estimation

The first part of the thesis revolves around distributed input-nonaffine control, where the goal is, given known dynamics of the system and a team of robots with information on the configuration of the system, to drive the system towards a desired configuration. Nonetheless, up to now, few attention has been paid to how to provide the robots with such information on the configuration of the system. Specially relevant is the work exposed in Chapter 2: robots needed full-state feedback for the multi-robot Implicit Control of herds. The prototypical solution was to use an extended distributed Kalman filter to obtain global awareness of the herd and other robot mates. However, from a broader perspective, a couple of questions arise.

The first one, closely related to the extended distributed Kalman filter, involves the optimality properties of the distributed Kalman filter. It is well-known that the centralized Kalman filter is optimal in the mean square error sense. Then, what about the distributed setting? This chapter explores this question, finding as answer the first event-triggered certifiable optimal distributed Kalman filter under unknown correlations, overcoming the traditional trade-off between convergence speed, communication efficiency and memory in distributed stochastic filtering settings.

The second question is also related to the the problems encountered in the first part of the thesis. A widespread mathematical representation of quantities of interest is that of ellipsoids. Examples can be found in the caging, dynamic assignment and CBF methods developed in Chapter 2, where the herd is represented as a 2D ellipse. On the other hand, stochastic estimates are also ellipsoids given by the covariance matrix associated to the uncertainty in the estimate. The question then is how to exploit the geometrical properties of ellipsoids to enhance distributed stochastic estimation over positive semi-definite matrices. This is explored in Chapter 5.

For now, in this Chapter 4 we focus on the first of the questions.

4.1 Related work

The Kalman Filter (KF) [Kalman, 1960] is a cornerstone in control theory. Its elegance and optimality has motivated its extension towards distributed setups. However, two major issues prevent a direct deployment in real setups. First, the conservation of the KF optimality. Second, the efficient usage of the communication bandwidth.

Regarding optimality, the works by Olfati-Saber [Olfati-Saber, 2005,Olfati-Saber, 2007b] opened an era of consensus-based Distributed Kalman Filters (DKF). However, it is proved that the computation of the estimator's gains is not scalable [Olfati-Saber, 2009, Zhu et al., 2013]. Since then, several works have aroused searching for an Optimal DKF (O-DKF) [Kamal et al., 2013, Battistelli et al., 2014, Das and Moura, 2016] by either keeping track of the whole network error covariance matrix or neglecting the information encoded in the covariance error matrices. In a recent work [Battilotti et al., 2021] it is discussed that, to reach optimal centralised KF performance, the state-of-the-art DKFs typically resort to multiple consensus steps among instants. The absence of solutions which optimally and scalably integrate the neighbouring estimates and error covariances motivated the development of Diffusion KFs (DfKF). The most popular approach is the Covariance Intersection (CI) method [Julier and Uhlmann, 1997, Julier and Uhlmann, 2001, Hu et al., 2011]. The CI method is generally suboptimal [Reinhardt et al., 2015], thus arising a necessity of certification. The solution to approach optimality goes through convex optimisation problems [Cattivelli and Sayed, 2010], sequential algorithms [Deng et al., 2012] or diffusion steps that require global topology knowledge [Zhang et al., 2015].

Regarding communications, the amount of information sent through the network is related to its energy consumption and the available bandwidth [Wu et al., 2012], so an Event-Triggered (ET) estimator is desirable. The most popular ET rule is the Send-on-Delta (SoD) [Suh et al., 2007], which computes the difference between estimates [Li et al., 2015, Liu et al., 2015a, Liu and Sun, 2021, Zhang and Jia, 2017], measurements [Suh et al., 2007, Liu and Sun, 2021, Zhang and Jia, 2017] and/or innovations [You and Xie, 2012, Liu et al., 2015b, Battistelli et al., 2018] to trigger the information when the difference exceeds some tuned threshold. SoD is popular because it is easy to compute and agnostic to the employed estimator. A work conducted by Trimpe and Campi [Trimpe and Campi, 2015] compared the most popular ET rules, concluding that leveraging the error covariance in the decision improves performance, e.g., the Kullback-Leibler (KL) divergence. Regarding KFs, ET Centralised Kalman Filters (CKF) [Battistelli et al., 2012, Wu et al., 2012, You and Xie, 2012, Han et al., 2013, Trimpe and D'Andrea, 2014] are subject to a critical arrival rate beyond which the error covariance becomes unbounded [Sinopoli et al., 2004], underlining the importance of a carefully designed ET protocol. On the other hand, both the SoD [Battistelli et al., 2018, Zhang and Jia, 2017, Liu and Sun, 2021] and KL [Battistelli et al., 2018, Battistelli et al., 2019] protocols are very popular in non-optimal DKFs.

4.2 Problem formulation

The target system is described by linear dynamics

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{w}(k), \tag{4.1}$$

where $k \in \mathbb{N}_{\geq 0}$ denotes the discrete time, $\mathbf{x} \in \mathbb{R}^n$ is the state of the system to estimate, $\mathbf{A} \in \mathbb{R}^{n \times n}$ is a matrix comprising the dynamics of the target system, and $\mathbf{w}(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ is a white process. The system is tracked by a network of sensors described by a directed communication graph $\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))$. The number of nodes is $\mathsf{N} = |\mathcal{V}|$, where $|\bullet|$ denotes cardinality. $\mathcal{N}_i(k) = \{j | (i,j) \in \mathcal{E}(k)\}$ is the set of neighbours of node i at instant k and $\mathcal{J}_i(k) = \mathcal{N}_i(k) \cup i$. We use $\lfloor \bullet \rfloor_{ij}$ to denote the ij-th element or block component of a matrix. Sensors are described by a linear model

$$\mathbf{z}_i(k) = \mathbf{H}_i \mathbf{x}(k) + \mathbf{v}_i(k). \tag{4.2}$$

In this expression, $\mathbf{z}_i \in \mathbb{R}^{m_i}$ is the measurement of node i, $\mathbf{H}_i \in \mathbb{R}^{m_i \times n}$ is the (unbiased) sensor model of node i, m_i is the dimension of $\mathbf{z}_i$ and $\mathbf{v}_i(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_i)$ is a white process. The measurements are independent between nodes. Notice that we are considering heterogeneous sensors. Unless unclear, from now on we will omit the dependencies with k. Regarding observability, we assume network-observability, defined as follows.

Definition 1. *System* (4.1) *is network-observable if the pair* $(\mathbf{A}, [\mathbf{H}_1^\top, \mathbf{H}_2^\top, \ldots, \mathbf{H}_N^\top]^\top)$ *is observable.*

The objective of the network is to cooperatively estimate the state of system (4.1). Each node has an estimate of the state of system (4.1), $\hat{\mathbf{x}}_i$, with associated error covariance matrix $\hat{\mathbf{P}}_i := \mathbb{E}[(\hat{\mathbf{x}}_i - \mathbf{x})(\hat{\mathbf{x}}_i - \mathbf{x})^\top]$. Besides, similarly to other typical filter algorithms, the nodes use an auxiliary variable for the prediction stage, $\bar{\mathbf{x}}_i$, and $\bar{\mathbf{P}}_i := \mathbb{E}[(\bar{\mathbf{x}}_i - \mathbf{x})(\bar{\mathbf{x}}_i - \mathbf{x})^\top]$. In the classical CKF, the estimate $\hat{\mathbf{x}}_i(k-1), \hat{\mathbf{P}}_i(k-1)$ is first used to obtain the prediction $\bar{\mathbf{x}}_i(k), \bar{\mathbf{P}}_i(k)$ by propagation through the target dynamics. Then, the measurement is used to correct the prediction, leading to the updated estimate $\hat{\mathbf{x}}_i(k), \hat{\mathbf{P}}_i(k)$. The CKF is not scalable in networked applications because a central computation unit is needed to aggregate the measurements from all the nodes. An alternative is to develop a DKF, where each node uses only local and neighbouring information.

Given, $\bar{\mathbf{P}}_i(k)$, we define $\bar{\mathbf{S}}_i(k) = \bar{\mathbf{P}}_i^{-1}(k)$ as the predicted error covariance matrices in information form. Specifically, we are interested in viewing these matrices as ellipsoids, defined as follows.

Definition 2. *Given* $\bar{\mathbf{S}}_i$ *and assuming unbiased sensors, the ellipsoid* ε_i^i *is*

$$\varepsilon_i^i := \{\, \mathbf{x} \,|\, \mathbf{x}^\top \bar{\mathbf{S}}_i^{-1} \mathbf{x} \leq 1 \,\}.$$

In our proposal, this absence of global knowledge leads to an NP-hard optimisation problem, caused by the unknown correlations between the nodes' estimates, forcing the use of convex relaxations to solve it. Despite enabling tractability, the solution of the relaxed problem is not guaranteed to be the optimum of the original problem. This motivates the need of certifiability on the optimisation, formally defined as follows:

Definition 3 (**From Definition 19 in [Yang et al., 2020]**). *Given an optimization problem $\mathbb{O}(\mathbb{D})$ that depends on input data $\mathbb{D}$, we say that an algorithm $\mathbb{A}$ is certifiable if, after solving $\mathbb{O}(\mathbb{D})$, $\mathbb{A}$ either provides a certificate for the optimality of its solution or declares failure otherwise.*

Besides, to avoid divergence of the estimation, the result of the optimisation must be consistent, defined as follows:

Definition 4. *An approximation $\tilde{\mathbf{P}}$ of $\mathbf{P}$ and $\tilde{\mathbf{x}}$ of $\mathbf{x}$ is consistent if and only if $\tilde{\mathbf{P}} - \mathbf{P} \succeq \mathbf{0}$ and $\mathbb{E}[(\tilde{\mathbf{x}} - \mathbf{x})] = \mathbf{0}$, where $\succeq$ means positive semi-definiteness.*

Many DKFs rely on time-triggered communications of all the nodes, requiring an intensive bandwidth usage. Instead, event-triggered communications save energy and bandwidth. In this regard, we define two concepts. The first is Periodic Joint Connectivity (PJC):

Definition 5. *Consider a sequence of graphs $\{\mathcal{G}(k), \mathcal{G}(k+i), \ldots, \mathcal{G}(k+t)\}$ with $t \in \mathbb{N}$. If $\mathbf{G}(k,t) = \cup_{i=0}^{t} \mathcal{G}(k+i)$ is a strongly connected graph, then $\mathbf{G}(k,t)$ is a periodic joint connected graph with period t at instant k.*

Second, we define a metric to measure the communication usage in the network. Before, we define the concept of "triggering function":

Definition 6. *A triggering function $g_i(\bullet, k) \in \{0, 1\}$ is an indicator function that depends only on local information at node i and such that, if $g_i(\bullet, k) = 1$, then node i broadcasts at instant k, and, if $g_i(\bullet, k) = 0$, then node i does not broadcast at instant k.*

Definition 7. *The Number of Broadcasts (*NoB*) at instant k in network $\mathcal{G}(k)$ is*

$$\text{NoB}(k) := \sum_{i=1}^{\mathsf{N}} g_i(\bullet, k). \tag{4.3}$$

We now formulate the problem addressed in the chapter.

Problem 1. *Find an event-triggered stable algorithm that certificates locally and in real time if, at each k, each node i minimises $\mathbb{E}[||\hat{\mathbf{x}}_i(k) - \mathbf{x}(k)||^2]$, under the following restrictions:*

1. ***Locality**: At instant k and $\forall i$, node i only uses $\mathbf{A}$, $\mathbf{H}_i$, $\mathbf{Q}$, and $\mathbf{R}_i$ as parameters. From instant k to $k+1$ and $\forall i$, node i stores $\hat{\mathbf{x}}_i(k)$, $\hat{\mathbf{P}}_i(k)$, $\mathbf{A}$, $\mathbf{Q}$, $\mathbf{H}_i$ and $\mathbf{R}_i$.*
2. ***One-hop communication**: At instant k and $\forall i$, node i communicates at most once with its neighbours $j \in \mathcal{N}_i(k)$.*
3. ***Consistency and connectivity**: The ET rule ensures consistency $\forall i, k$, and PJC of $\mathcal{G}(k)$ $\forall k$ and some finite t.*
4. ***Network-observability**.*

Remark 1. *From Problem 1, each node seeks to minimize* $\mathbb{E}[||\hat{\mathbf{x}}_i(k) - \mathbf{x}(k)||^2]$. *Therefore, globally, the network minimises* $\sum_i \mathbb{E}[||\hat{\mathbf{x}}_i(k) - \mathbf{x}(k)||^2]$. *Note that this definition holds because the minimisation refers to the current instant rather than a finite or infinite horizon of time.*

We will use * to denote optimality in the sense of Problem 1. We assume that communication delays, quantisation effects and dropouts are negligible, a common assumption in many works (see [Shi et al., 2014, Liu et al., 2015b, Battistelli et al., 2012, You and Xie, 2012] among others).

4.3 ECO-DKF algorithm

The proposed solution for Problem 1 is the ECO-DKF algorithm. We first offer an overview of ECO-DKF and then, in Sections 4.4, 4.5 and 4.6, we formally study the main properties of the estimator.

ECO-DKF algorithm is based on a novel optimization problem to aggregate predicted KF estimates. We propose to use the outer Löwner-John (LJ) method [John, 2014], which computes the smallest ellipsoid that surrounds the intersection of a set of ellipsoids,

$$\bar{\mathbf{S}}_i^*, \boldsymbol{\lambda}_i^* = \underset{\bar{\mathbf{S}}, \boldsymbol{\lambda}}{\arg\min} \quad \operatorname{tr}(\bar{\mathbf{S}}^{-1}) \tag{4.4a}$$

$$s.t. \quad \mathbf{0} \prec \bar{\mathbf{S}} \preceq \sum_{j=1}^{|\mathcal{J}_i|} \lambda_j \bar{\mathbf{S}}_j, \tag{4.4b}$$

$$\sum_{j=1}^{|\mathcal{J}_i|} \lambda_j \leq 1, \ \lambda_j \geq 0 \ \forall j \in \mathcal{J}_i, \tag{4.4c}$$

where $\bar{\mathbf{S}}_j = \bar{\mathbf{P}}_j^{-1}$ characterise the ellipsoids, $\operatorname{tr}(\bullet)$ is the trace of a matrix, λ_j is the j-th element of vector $\boldsymbol{\lambda}$, and $\prec$ and $\preceq$ denote definiteness and semi-definiteness. The selection of the trace as the optimisation cost function follows from minimising the square error $\mathbb{E}[||\hat{\mathbf{x}}_i - \mathbf{x}||^2]$ in Problem 1, proved later on Theorem 9. The input of the outer LJ method is the predicted covariance, in information form, from node i and its neighbours. Meanwhile, $\boldsymbol{\lambda}_i$ weights the importance of each prediction. The output of (4.4) is used to aggregate the predictions as follows

$$\bar{\mathbf{x}}_i^* = \bar{\mathbf{P}}_i^* \sum_{j \in \mathcal{J}_i(k)} \lambda_{ij}^* \bar{\mathbf{s}}_j \text{ and } \bar{\mathbf{P}}_i^* = (\bar{\mathbf{S}}_i^*)^{-1}. \tag{4.5}$$

The study of this aggregation is developed in Section 4.4, including the certification guarantees on (4.4) which permits the formal results about optimality in Section 4.5. To the best of our knowledge, this is the first time optimisation problem (4.4) is applied in a DKF.

Interestingly, the output of (4.4) allows to develop a novel ET rule to decide if a node broadcasts a message to its neighbours:

$$g_i(\boldsymbol{\lambda}_i^*(k-1), k) \equiv g_i(\boldsymbol{\lambda}_i^*) = \begin{cases} 1, \text{ if } (\max(\boldsymbol{\lambda}_i^*) = \lambda_{ii}^*) \vee (\mathcal{J}_i = \{i\} \wedge \operatorname{Ber}(p) = 1) \\ 0, \text{ otherwise} \end{cases} \tag{4.6}$$

where $\lambda_{ii}^* \in [0,1]$ is the weight of $\boldsymbol{\lambda}_i^*$ associated to node i, $\vee$ is the logical "OR", $\wedge$ is the logical "AND", and $\text{Ber}(p)$ represents the Bernoulli distribution of probability p. The messages are defined by $\text{msg} = \{\mathbf{U}_i, \mathbf{u}_i, \bar{\mathbf{S}}_i, \bar{\mathbf{s}}_i\}$, where

$$\mathbf{u}_i = \mathbf{H}_i^\top \mathbf{R}_i^{-1} \mathbf{z}_i, \quad \mathbf{U}_i = \mathbf{H}_i^\top \mathbf{R}_i^{-1} \mathbf{H}_i, \quad \bar{\mathbf{s}}_i = \bar{\mathbf{P}}_i^{-1} \bar{\mathbf{x}}_i, \quad \bar{\mathbf{S}}_i = \bar{\mathbf{P}}_i^{-1}.$$

The quantities are in information form [Anderson and Moore, 2012] to comply with restriction 1 of Problem 1. We use the term "broadcast" because, when a node sends its information, there is no distinction to which neighbours the message arrives. A selective communication requires establishment of particular communication channels and a priori knowledge on the set of neighbours. Besides, we do not allow multiple communications within an instant.

Now, we are ready to present ECO-DKF in Algorithm 8. Given $\hat{\mathbf{x}}_i^*(k-1)$, $\hat{\mathbf{P}}_i^*(k-1)$ and $\boldsymbol{\lambda}_i^*(k-1)$, each node first computes the prediction of the state through the target system dynamics,

$$\bar{\mathbf{P}}_i(k) = \mathbf{A}\hat{\mathbf{P}}_i^*(k-1)\mathbf{A}^\top + \mathbf{Q}, \tag{4.7}$$

$$\bar{\mathbf{x}}_i(k) = \mathbf{A}\hat{\mathbf{x}}_i^*(k-1). \tag{4.8}$$

Then, each node obtains its measurement $\mathbf{z}_i(k)$ and decides whether to broadcast a message or not according to the ET rule (4.6). Node i receives messages from broadcasting neighbours. The information from the messages is aggregated to compute the values that will be used in the correction step. In the case of the measurements, this is direct because they are independent

$$\mathbf{y}_i = \sum_{j \in \mathcal{J}_i} \mathbf{u}_j \text{ and } \mathbf{Y}_i = \sum_{j \in \mathcal{J}_i} \mathbf{U}_j. \tag{4.9}$$

On the other hand, the aggregation of predictions is harder because we must consider the unknown correlations between them. To solve this, we use (4.4) and (4.5).

Finally, each node calculates the estimated error covariance matrix

$$\hat{\mathbf{P}}_i^* = \left(\bar{\mathbf{S}}_i^* + \mathbf{Y}_i\right)^{-1}. \tag{4.10}$$

This is analogous to the correction step of the information form [Anderson and Moore, 2012] DKF presented by Olfati-Saber (Eq. (10) in [Olfati-Saber, 2009]), where the optimal predicted information covariance matrix, $\bar{\mathbf{S}}_i^*$, is fused with all the measurement information covariance matrices, $\mathbf{Y}_i$, because the measurements are uncorrelated and $\bar{\mathbf{S}}_i^*$ is consistent (proven in Proposition 12). The estimated state is obtained as follows:

$$\hat{\mathbf{x}}_i^* = \bar{\mathbf{x}}_i^* + \hat{\mathbf{P}}_i^*(\mathbf{y}_i - \mathbf{Y}_i \bar{\mathbf{x}}_i^*). \tag{4.11}$$

It is noteworthy to see that Eq. (4.11) is similar to

$$\hat{\mathbf{x}}_i = \bar{\mathbf{x}}_i + \hat{\mathbf{P}}_i^* \left(\mathbf{y}_i - \mathbf{Y}_i \bar{\mathbf{x}}_i\right) + \gamma \hat{\mathbf{P}}_i^* \sum_{j \in \mathcal{N}_i} (\bar{\mathbf{x}}_j - \bar{\mathbf{x}}_i), \tag{4.12}$$

which comes from the consensus-based DKF in [Olfati-Saber, 2007b]. However, in ECO-DKF the consensus is implicit in the optimisation, overcoming the dependence for stability on the parameter γ.

Algorithm 8 ECO-DKF in node i

Initialisation: $\hat{\mathbf{P}}_i^*(0) = \mathbf{P}_0$, $\hat{\mathbf{x}}_i^*(0) = \mathbf{x}_0$, $\boldsymbol{\lambda}_i^*(0) = \boldsymbol{\lambda}_0$
while True **do**
  Prediction: $\bar{\mathbf{P}}_i(k) = \mathbf{A}\hat{\mathbf{P}}_i^*(k-1)\mathbf{A}^\top + \mathbf{Q}$, $\bar{\mathbf{x}}_i(k) = \mathbf{A}\hat{\mathbf{x}}_i^*(k-1)$
  Measurement: get $\mathbf{z}_i(k)$
  Broadcasting:
    if $g_i(\boldsymbol{\lambda}_i^*(k-1), k) == 1$ **then**
      Send $\{\mathbf{U}_i(k), \mathbf{u}_i(k), \bar{\mathbf{S}}_i(k), \bar{\mathbf{s}}_i(k)\}$
    end if
    Receive $\{\mathbf{U}_j(k), \mathbf{u}_j(k), \bar{\mathbf{S}}_j(k), \bar{\mathbf{s}}_j(k)\}$ $\forall j \in \mathcal{N}_i(k)$
  Aggregation of measurements' data: $\mathbf{Y}_i(k) = \sum_{j\in\mathcal{J}_i(k)} \mathbf{U}_j(k)$, $\mathbf{y}_i(k) = \sum_{j\in\mathcal{J}_i(k)} \mathbf{u}_j(k)$
  Aggregation of predictions' data: $\bar{\mathbf{S}}_i^*(k)$,$\boldsymbol{\lambda}_i^*(k) \leftarrow$ Solution of (4.4)

$$\bar{\mathbf{P}}_i^*(k) = (\bar{\mathbf{S}}_i^*(k))^{-1},\ \bar{\mathbf{x}}_i^*(k) = \bar{\mathbf{P}}_i^*(k)\sum_{j\in\mathcal{J}_i(k)} \lambda_{ij}^*(k)\bar{\mathbf{s}}_j(k)$$

  Correction: $\hat{\mathbf{P}}_i^*(k) = (\bar{\mathbf{S}}_i^*(k) + \mathbf{Y}_i(k))^{-1}$, $\hat{\mathbf{x}}_i^*(k) = \bar{\mathbf{x}}_i^*(k) + \hat{\mathbf{P}}_i^*(k)(\mathbf{y}_i(k) - \mathbf{Y}_i(k)\bar{\mathbf{x}}_i^*(k))$
end while

We briefly discuss the communication and computational burden. Regarding communication, the size of each message is constant with the number of nodes, so the proposal is scalable. Besides, nodes do not need any global knowledge of the topology nor sensor models of neighbours. Regarding the computational cost, the bottleneck is (4.4). However, the current hardware can solve large instances of this optimisation problem in real-time [Yang et al., 2020].

4.4 Certifiable covariance bounding

To achieve the optimality pursued in Problem 1 it is necessary to deal with the optimal aggregation of neighbouring predictions under unknown correlations. This fusion can be described as finding the minimum volume ellipsoid containing the intersection of the p ellipsoids [Boyd et al., 1994] formed by the matrices $\bar{\mathbf{P}}_j$ in the set $\mathcal{J}_i(k)$. Thus, we first define the concept of intersection of ellipsoids.

Definition 8. *The intersection of p ellipsoids, at node i, is the convex set*

$$\mathcal{F}_i := \varepsilon_1^i \cap \ldots \cap \varepsilon_j^i \cap \ldots \cap \varepsilon_p^i.$$

From Definitions 2 and 8, volume minimisation can be transformed into a maximisation over the information matrices. In particular since the expected square error is equivalent to the trace of $\hat{\mathbf{P}}_i^*$, the objective of ECO-DKF is to optimise this metric.

Problem 2 (**Adapted from [Boyd et al., 1994]**). *Find $(\varepsilon_i^i)^*$ such that it contains $\mathcal{F}_i$ and* $\mathrm{tr}(\bar{\mathbf{S}}_i^*)$ *is maximised.*

Problem 2 is solved by the following optimisation problem [Henrion et al., 2001]:

$$\max_{\mathbf{X}} \quad \text{tr}(\mathbf{X}) \tag{4.13a}$$

$$s.t. \quad \mathbf{X} \succeq \mathbf{0},\ \text{tr}(\mathbf{X}\bar{\mathbf{S}}_j) \leq 1 \ \ \forall j \in \mathcal{J}_i(k), \tag{4.13b}$$

$$\text{rank}(\mathbf{X}) = 1. \tag{4.13c}$$

This problem is NP-hard due to the non-convex constraint (4.13c). Thus, it is necessary to find a convex relaxation to make the optimisation tractable. The simplest solution is to drop the non-convex constraint,

$$\max_{\mathbf{X}} \quad \text{tr}(\mathbf{X}) \tag{4.14a}$$

$$s.t. \quad \mathbf{X} \succeq \mathbf{0},\ \text{tr}(\mathbf{X}\bar{\mathbf{S}}_j) \leq 1 \ \ \forall j \in \mathcal{J}_i(k), \tag{4.14b}$$

However, it is not possible to use (4.14) in ECO-DKF because it does not provide an estimate nor an error covariance matrix, but $\mathbf{X}$. Instead, we propose to use the outer Löwner-John (LJ) method [John, 2014], which leads to the relaxed optimisation problem in (4.4). The input of the outer LJ method is the $|\mathcal{J}_i(k)|$ ellipsoids described by matrices $\bar{\mathbf{S}}_j$ $\forall j \in \mathcal{J}_i(k)$. Then, it computes the smallest ellipsoid that includes $\mathcal{F}_i(k)$. The intersection always exist since the ellipsoids are all centered at $\mathbf{0}$ [Henrion et al., 2001]. From (4.4b), $\bar{\mathbf{S}}_i^*$ is upper-bounded by the weighted sum of matrices $\bar{\mathbf{S}}_j$ $\forall j \in \mathcal{J}_i(k)$. Therefore, $\bar{\mathbf{S}}_i^*$ depends on the topology and the target system dynamics, for they determine the set of predicted error covariance matrices considered in the optimisation (Eq. (4.4b)) and the value of each prediction (Eqs. (4.7)-(4.8)) respectively. Problem (4.4) outputs an estimate and error covariance matrix. Nevertheless, problem (4.14) is important because it enables the certification of optimality. This is formally stated in the next Proposition, whose proof can be found in [Sebastián et al., 2021c]:

Proposition 10. *Let* $\mathbf{X}^*$ *be the solution of* (4.14)*. Define*

$$\mathbb{C}_i := \text{rank}\,(\mathbf{X}^*) \ \ \textit{and}\ \rho_i := \text{tr}(\mathbf{X}^*)\vartheta(\bar{\mathbf{S}}_i^*) \in [0,1], \tag{4.15}$$

where $\vartheta(\bar{\mathbf{S}}_i^*)$ *denotes the minimum eigenvalue of* $\bar{\mathbf{S}}_i^*$*, obtained by solving* (4.4)*. If it holds that* $\mathbb{C}_i = \rho_i = 1$*, then the solution of* (4.4) *is the optimum of the original non-relaxed problem* (4.13)*.*

Proof. If constraint (4.13c) holds for $\mathbf{X}^*$, then it is also the optimum of (4.13). This can be demonstrated by contradiction. Consequently, $\mathbb{C}_i = 1$ is a certificate of this equivalence.

The next part is to show that the optimums of (4.14) and (4.4) are equivalent. This is described in detail in Section 2 of [Henrion et al., 2001], so we only sketch here the procedure. Obtained the Lagrangian of (4.14) and assuming that $\mathcal{F}$ does not reduce to zero, by Slater's condition both dual and primal have the same optimum. Then, by the S-procedure, (4.4) is equivalent to the dual and hence equivalent to (4.14).

If it is the case, then outputs of (4.14) and (4.4) can be compared to see how close is the solution of (4.4) to the solution of Problem 4.13. To do so, Theorem 4 of [Henrion et al., 2001] states that the optimal value of (4.4) is bounded by $1/\vartheta(\bar{\mathbf{S}}_i^*)$, where $\bar{\mathbf{S}}_i^*$ is the

output of the optimisation in (4.4). Therefore, ρ_i assesses the output of (4.4) comparing it with the optimal value certified by the constraint (4.13c). If $\mathbb{C}_i = 1$ and $\rho_i = 1$, then the solution of (4.4) is the optimum of the original Problem 2. □

The proposition gives a procedure where each node can check, locally and in real-time, the performance of its optimisation process and assess it by finding the optimal value for Problem 2. This is of key importance not only to verify the optimality of the estimation, but also the certificate can be used in other parts of the control pipeline to, e.g., assure optimality of the controller. In our case, we will leverage this process to design an ET communication rule.

4.5 Stability and optimality

In this section we analyse the stability and optimality of ECO-DKF. For clarity, we assume that all nodes broadcast their information to neighbours at all instants, such that the underlying communication graph is undirected and connected. In the next section we will relax the assumption.

Define $\eta_i = \hat{\mathbf{x}}_i - \mathbf{x}$,

$$\begin{aligned}
\eta =&[\eta_1^\top, \dots, \eta_{\mathsf{N}}^\top]^\top, & \mathbf{H} =&\text{block-diag}(\mathbf{H}_1, \dots, \mathbf{H}_{\mathsf{N}})\\
\mathbf{z} =&[\mathbf{z}_1^\top, \dots, \mathbf{z}_N^\top]^\top, & \mathbf{R} =&\text{block-diag}(\mathbf{R}_1, \dots, \mathbf{R}_{\mathsf{N}}),\\
\mathbf{y} =&[\mathbf{y}_1^\top, \dots, \mathbf{y}_N^\top]^\top, & \mathbf{U} =&\text{block-diag}(\mathbf{U}_1, \dots, \mathbf{U}_{\mathsf{N}}),\\
\mathcal{A} =&\mathbf{I} \otimes \mathbf{A}, & \hat{\mathbf{P}}^* =&\text{block-diag}(\hat{\mathbf{P}}_1^*, \dots, \hat{\mathbf{P}}_{\mathsf{N}}^*),\\
\mathcal{Q} =&\mathbf{I} \otimes \mathbf{Q}, & \bar{\mathbf{P}}^* =&\text{block-diag}(\bar{\mathbf{P}}_1^*, \dots, \bar{\mathbf{P}}_{\mathsf{N}}^*),
\end{aligned}$$

and

$$\mathbf{Y} \overset{Eq.\ (4.9)}{=} \mathbf{N}_w \mathbf{U} \text{ , where } \begin{array}{l} \lfloor \mathbf{N}_w \rfloor_{ij} = \mathbf{I} \text{ if } j \in \mathcal{J}_i \\ \lfloor \mathbf{N}_w \rfloor_{ij} = 0 * \mathbf{I} \text{ otherwise} \end{array} . \tag{4.16}$$

To demonstrate the global asymptotic stability of ECO-DKF, we adapt and prove two Lemmas from [Olfati-Saber, 2009], extending them from the discrete-time centralized Kalman filter setting to our ECO-DKF setting. The first builds some useful matrices.

Lemma 3 (**Adapted from Lemma 2 in [Olfati-Saber, 2009]**). *Given Eqs.* (4.10)-(4.11)*, the following holds:*

1. $\mathbf{F} = \mathbf{I} - \hat{\mathbf{P}}^*\mathbf{Y} = \hat{\mathbf{P}}^*(\bar{\mathbf{P}}^*)^{-1}$.
2. $\hat{\mathbf{P}}^* = \mathbf{F}\mathbf{G}\mathbf{F}^\top$ *with* $\mathbf{G} = \mathcal{A}\hat{\mathbf{P}}^*\mathcal{A}^\top + \mathcal{Q} + \mathbf{T}\mathbf{R}^{-1}\mathbf{T}^\top$ *and* $\mathbf{T} = \bar{\mathbf{P}}^*\mathbf{N}_w\mathbf{H}^\top$.

Proof. Let $\mathbf{F} = \mathbf{I} - \hat{\mathbf{P}}^*\mathbf{Y}$, by Eq. (4.10) we can write $\hat{\mathbf{P}}^* = ((\bar{\mathbf{P}}^*)^{-1} + \mathbf{Y})^{-1}$. Then,

$$\hat{\mathbf{P}}^*((\bar{\mathbf{P}}^*)^{-1} + \mathbf{Y}) = \mathbf{I} \Rightarrow \mathbf{I} - \hat{\mathbf{P}}^*\mathbf{Y} = \hat{\mathbf{P}}^*(\bar{\mathbf{P}}^*)^{-1} \tag{4.17}$$

and the first statement is proved. Regarding the second statement, notice that Eq. (4.11) can be stacked as follows

$$\hat{\mathbf{x}}^* = \bar{\mathbf{x}}^* + \hat{\mathbf{P}}^*(\mathbf{y} - \mathbf{Y}\bar{\mathbf{x}}^*). \tag{4.18}$$

Then, taking into account that $\mathbf{y} = \mathbf{N}_w\mathbf{H}^\top\mathbf{R}^{-1}\mathbf{z}$ and $\mathbf{Y} = \mathbf{N}_w\mathbf{U} = \mathbf{N}_w\mathbf{H}^\top\mathbf{R}^{-1}\mathbf{H}$, Eq. (4.18) is rewritten from its information form as

$$\hat{\mathbf{x}}^* = \bar{\mathbf{x}}^* + \hat{\mathbf{P}}^*\mathbf{N}_w\mathbf{H}^\top\mathbf{R}^{-1}(\mathbf{z} - \mathbf{H}\bar{\mathbf{x}}^*) = \bar{\mathbf{x}}^* + \mathbf{K}(\mathbf{z} - \mathbf{H}\bar{\mathbf{x}}^*) \tag{4.19}$$

with $\mathbf{K} = \hat{\mathbf{P}}^*\mathbf{N}_w\mathbf{H}^\top\mathbf{R}^{-1}$ the Kalman Gain of the standard KF formulation. Let now consider the update of $\hat{\mathbf{P}}^*$, where by update, denoted as $(\hat{\mathbf{P}}^*)^+$, we refer to the operations that compute $\hat{\mathbf{P}}^*(k)$ from $\hat{\mathbf{P}}^*(k-1)$ and $\bar{\mathbf{P}}^*(k-1)$

$$(\hat{\mathbf{P}}^*)^+ = \mathbf{F}\bar{\mathbf{P}}^+\mathbf{F}^\top + \mathbf{K}\mathbf{R}\mathbf{K}^\top, \tag{4.20}$$

which comes from the standard KF [Olfati-Saber, 2009]. By substituting $\mathbf{K}$ we obtain

$$(\hat{\mathbf{P}}^*)^+ = \mathbf{F}(\mathcal{A}\hat{\mathbf{P}}^*\mathcal{A}^\top + \mathcal{Q})\mathbf{F}^\top + \hat{\mathbf{P}}^*\mathbf{N}_w\mathbf{H}^\top\mathbf{R}^{-1}\mathbf{H}\mathbf{N}_w^\top(\hat{\mathbf{P}}^*)^\top. \tag{4.21}$$

Given statement 1 of the Lemma,

$$(\hat{\mathbf{P}}^*)^+ = \mathbf{F}(\mathcal{A}\hat{\mathbf{P}}^*\mathcal{A}^\top + \mathcal{Q} + \bar{\mathbf{P}}^*\mathbf{N}_w\mathbf{H}^\top\mathbf{R}^{-1}\mathbf{H}\mathbf{N}_w^\top(\bar{\mathbf{P}}^*)^\top)\mathbf{F}^\top, \tag{4.22}$$

and statement 2 is proved. □

The next Lemma proves global asymptotic stability for general dynamics, used later to prove ECO-DKF global asymptotic stability.

Lemma 4 (**Adapted from Lemma 3 in [Olfati-Saber, 2009]**). *Suppose that the error dynamics without noise are $\eta^+ = \mathbf{F}\mathcal{A}\eta$, with $\mathbf{F}$ as in Lemma 3. Then, the error dynamics is globally asymptotically stable system with a Lyapunov function $\mathcal{V}(\eta) = \eta^\top(\hat{\mathbf{P}}^*)^{-1}\eta$, provided that $g_i(\boldsymbol{\lambda}_i^*) = 1$ for all i, k and $\mathcal{G}(k)$ is undirected and connected for all k.*

Proof. Given $V(\eta) = \eta^\top(\hat{\mathbf{P}}^*)^{-1}\eta$ as Lyapunov function candidate,

$$\begin{aligned}
\delta V =& (\eta^+)^\top((\hat{\mathbf{P}}^*)^+)^{-1}\eta^+ - \eta^\top(\hat{\mathbf{P}}^*)^{-1}\eta = \\
& \eta^\top\left(\mathcal{A}^\top\mathbf{F}^\top((\hat{\mathbf{P}}^*)^+)^{-1}\mathbf{F}\mathcal{A} - (\hat{\mathbf{P}}^*)^{-1}\right)\eta = \\
& \eta^\top\left(\mathcal{A}^\top\mathbf{G}^{-1}\mathcal{A} - (\hat{\mathbf{P}}^*)^{-1}\right)\eta = \\
& -\eta^\top\left((\hat{\mathbf{P}}^*)^{-1} - \mathcal{A}^\top(\mathcal{A}\hat{\mathbf{P}}^*\mathcal{A}^\top + \mathbf{W})^{-1}\mathcal{A}\right)\eta = \\
& -\eta^\top\Lambda\eta,
\end{aligned} \tag{4.23}$$

with $\mathbf{W} = \mathcal{Q} + \mathbf{T}\mathbf{R}^{-1}\mathbf{T}^\top \succ \mathbf{0}$ and $\Lambda = (\hat{\mathbf{P}}^*)^{-1} - \mathcal{A}^\top(\mathcal{A}\hat{\mathbf{P}}^*\mathcal{A}^\top + \mathbf{W})^{-1}\mathcal{A}$. The rest of the proof directly follows from Lemma 3 of [Olfati-Saber, 2009], showing that $\Lambda \succ \mathbf{0}$. □

Finally, to prove stability of ECO-DKF we employ a Lemma, whose proof can be found in [Sebastián et al., 2021c], which allows to rewrite ECO-DKF filter equations and show that the optimum of optimization problem (4.4) is such that inequality (4.4b) is an equality.

Lemma 5. *Given optimisation problem* (4.4), $\bar{\mathbf{S}}_i^* = \sum_{j\in\mathcal{J}_i} \lambda_{ij}^* \bar{\mathbf{S}}_j$, *for all* i.

Proof. Problem (4.4) is a convex maximisation with respect to $\bar{\mathbf{S}}$ and λ, which means that its optimum is unique. Notice that the cost function is the trace of $\bar{\mathbf{S}}$. Therefore, maximisation of $\mathrm{tr}(\bar{\mathbf{S}})$ with respect to λ implies that the maximum is found when constraint (4.4c) is an equality and $\sum_{j=1}^p \lambda_j^* \bar{\mathbf{S}}_j$ is the convex combination of $\bar{\mathbf{S}}_j$ with maximum trace. This can be demonstrated by contradiction: if $\sum_{j=1}^p \lambda_j < 1$ then there exists another convex combination of $\bar{\mathbf{S}}_j$ with a higher trace, which means that the optimum is with $\sum_{j=1}^p \lambda_j = 1$; in addition, the values of λ_j must be such that $\sum_{j=1}^p \lambda_j \bar{\mathbf{S}}_j$ is maximised, otherwise there exists a different convex combination of $\bar{\mathbf{S}}_j$ with a higher trace.

Now, notice that $\sum_{j=1}^p \lambda_j^* \bar{\mathbf{S}}_j$ is in constraint (4.4b) bounding the value of $\bar{\mathbf{S}}$ such that matrix $\sum_{j=1}^p \lambda_j^* \bar{\mathbf{S}}_j - \bar{\mathbf{S}}$ must be positive semi-definite. Therefore, the optimal value of problem (4.4) must be the supremum of constraint (4.4b) and the optimal value of $\bar{\mathbf{S}}$ must share eigenvalues with $\sum_{j=1}^p \lambda_j^* \bar{\mathbf{S}}_j$. Since they share eigenvalues, the trace of both quantities are equal and $\mathrm{tr}(\bar{\mathbf{S}}_i^*) = \mathrm{tr}(\sum_{j=1}^p \lambda_j^* \bar{\mathbf{S}}_j)$ holds. Given the uniqueness of optimum, the optimal value of $\bar{\mathbf{S}}$ is $\sum_{j=1}^p \lambda_j^* \bar{\mathbf{S}}_j$ and the statement of the Lemma is proved. □

Lemma 5 says that the optimisation in (4.4) becomes a standard discrete-time consensus protocol tuned to optimise the trace of the final consensus value of each node. Interestingly, the optimisation will implicitly assign largest λ_{ij}^* to nodes equipped with better sensors and estimates, which is a positive side effect.

These ingredients lead to the stability of the filter.

Theorem 8. *Algorithm 8 is a globally asymptotically stable estimator if* $g_i(\boldsymbol{\lambda}_i^*) = 1$ *for all* i, k.

Proof. First, rewrite Eq. (4.11) like Eq. (4.12). The aggregated prediction $\bar{\mathbf{x}}_i^*$ in Eq. (4.11) is

$$\bar{\mathbf{x}}_i^* = \bar{\mathbf{P}}_i^* \sum_{j\in\mathcal{J}_i} \lambda_{ij}^* \bar{\mathbf{P}}_j^{-1} \bar{\mathbf{x}}_j. \tag{4.24}$$

This expression can be rewritten as

$$\bar{\mathbf{x}}_i^* = \bar{\mathbf{P}}_i^* \sum_{j\in\mathcal{J}_i} \lambda_{ij}^* \bar{\mathbf{P}}_j^{-1} \bar{\mathbf{x}}_i + \bar{\mathbf{P}}_i^* \sum_{j\in\mathcal{N}_i} \lambda_{ij}^* \bar{\mathbf{P}}_j^{-1} (\bar{\mathbf{x}}_j - \bar{\mathbf{x}}_i). \tag{4.25}$$

From Lemma 5, $\bar{\mathbf{P}}_i^* \sum_{j\in\mathcal{J}_i} \lambda_{ij}^* \bar{\mathbf{P}}_j^{-1} = \mathbf{I}$. Using (4.25) in (4.11) gives

$$\hat{\mathbf{x}}_i^* = \bar{\mathbf{x}}_i + \hat{\mathbf{P}}_i^* (\mathbf{y}_i - \mathbf{Y}_i \bar{\mathbf{x}}_i) + (\mathbf{I} - \hat{\mathbf{P}}_i^* \mathbf{Y}_i) \bar{\mathbf{P}}_i^* \sum_{j\in\mathcal{N}_i} \lambda_{ij}^* \bar{\mathbf{P}}_j^{-1} (\bar{\mathbf{x}}_j - \bar{\mathbf{x}}_i). \tag{4.26}$$

Eq. (4.26) is equivalent to Eq. (4.12) but weighting each term with the result of optimising (4.4). Then, the noiseless dynamics of η_i is

$$\eta_i^+ = \mathbf{A}\eta_i - \hat{\mathbf{P}}_i^* \mathbf{Y}_i \mathbf{A} \eta_i + (\mathbf{I} - \hat{\mathbf{P}}_i^* \mathbf{Y}_i) \bar{\mathbf{P}}_i^* \sum_{j\in\mathcal{N}_i} \lambda_{ij}^* \bar{\mathbf{P}}_j^{-1} \mathbf{A} (\eta_j - \eta_i) \tag{4.27}$$

which can also be written in compact form for the whole network as

$$\eta^{+} = \mathbf{F}\,\mathbf{L}_w\,\mathcal{A}\eta. \tag{4.28}$$

where $(\bullet)^{+}$ is the update operator defined as in Lemma 4 and the proof of Lemma 3, $\mathbf{F}$ is defined as in Lemma 3, and $\mathbf{L}_w$ is such that $\lfloor \mathbf{L}_w \rfloor_{ij} = \bar{\mathbf{P}}_i^{*}\lambda_{ij}^{*}\bar{\mathbf{P}}_j^{-1}$ for all $j \in \mathcal{J}_i$ and $\mathbf{0}$ otherwise. The last expression is similar to the error dynamics proved as globally asymptotically stable in Lemma 4, but with $\mathbf{L}_w$ in between. In the proof of Lemma 4 it is shown that

$$\Lambda = (\hat{\mathbf{P}}^{*})^{-1} - \mathcal{A}^{\top}\mathbf{G}^{-1}\mathcal{A} \succ \mathbf{0}. \tag{4.29}$$

Instead, we have that $\Lambda^{'} = (\hat{\mathbf{P}}^{*})^{-1} - \mathcal{A}^{\top}\mathbf{L}_w^{\top}\mathbf{G}^{-1}\mathbf{L}_w\mathcal{A}$. If

$$\mathcal{A}^{\top}\mathbf{G}^{-1}\mathcal{A} \succeq \mathcal{A}^{\top}\mathbf{L}_w^{\top}\mathbf{G}^{-1}\mathbf{L}_w\mathcal{A} \quad \Rightarrow \quad \mathbf{I} \succeq \mathbf{L}_w, \tag{4.30}$$

then $\Lambda^{'} \succ \mathbf{0}$ and global asymptotic stability is proved. Given Lemma 5, $\mathbf{L}_w$ is a row-stochastic matrix. Therefore, by linear algebra results, the absolute value of any eigenvalue of $\mathbf{L}_w$ is less than or equal to 1. This means that the eigenvalues of matrix $\mathbf{L}_w - \mathbf{I}$ are all negative or equal to 0 and

$$\mathbf{0} \succeq \mathbf{L}_w - \mathbf{I} \Rightarrow \mathbf{I} \succeq \mathbf{L}_w. \tag{4.31}$$

Thus, (4.30) holds and $\Lambda^{'} \succ \mathbf{0}$, concluding the proof. □

Notice that stability holds independently on certification. The next step is to demonstrate optimality under positive certification.

Theorem 9. *Assume that the solution of* (4.4) *is certified as optimal. Then, Algorithm 8 is optimal in the sense of Problem 1 and* (4.4) *provides the optimal consensus gain of Eq.* (4.12)*.*

Proof. We reformulate Problem 1 as an optimal Bayesian estimation problem over the network $\mathcal{G}$. Let $Z(k) = \{\mathbf{z}(0), \ldots, \mathbf{z}(k)\}$, where $\mathbf{z} = [\mathbf{z}_1^{\top}, \ldots, \mathbf{z}_N^{\top}]^{\top}$. The posterior that minimises the MSE is given by $\hat{\mathbf{x}}(k) = \mathbb{E}[\mathbf{x}(k)|Z(k)] = \int \mathbf{x}(k)P(\mathbf{x}(k)|Z(k))d\mathbf{x}(k)$. To obtain $P(\mathbf{x}(k)|Z(k))$ from $P(\mathbf{x}(k-1)|Z(k-1))$, the Bayesian prediction and correction equations are:

$$\begin{aligned} P(\mathbf{x}(k)|Z(k-1)) &= \int P(\mathbf{x}(k)|\mathbf{x}(k-1))P(\mathbf{x}(k-1)|Z(k-1))d\mathbf{x}(k-1) \\ P(\mathbf{x}(k)|Z(k)) &= \frac{P(\mathbf{z}(k)|\mathbf{x}(k))P(\mathbf{x}(k)|Z(k-1))}{\int P(\mathbf{z}(k)|\mathbf{x}(k))P(\mathbf{x}(k)|Z(k-1))d\mathbf{x}(k)} \end{aligned} \tag{4.32}$$

Under unknown correlations, $P(\mathbf{x}(k)|Z(k-1)) = \mathcal{N}(\bar{\mathbf{x}}(k), \bar{\mathbf{P}}(k))$:

$$\bar{\mathbf{x}}(k) = \mathcal{A}\hat{\mathbf{x}}^{*}(k-1) \quad \text{and} \quad \bar{\mathbf{P}}(k) = \mathcal{A}\hat{\mathbf{P}}^{*}(k-1)\mathcal{A}^{\top} + \mathcal{Q}, \tag{4.33}$$

with $\hat{\mathbf{P}}^{*}(k-1) = \text{block-diag}(\hat{\mathbf{P}}_1^{*}(k-1), \ldots, \hat{\mathbf{P}}_N^{*}(k-1))$. This is the prediction step. Regarding the correction step, we have that $\mathbf{y} = [\mathbf{y}_1^{\top}, \ldots, \mathbf{y}_N^{\top}]^{\top}$, $\mathcal{R} = \text{block-diag}(\mathbf{Y}_1^{-1}, \ldots, \mathbf{Y}_{\mathsf{N}}^{-1})$, and $P(\mathbf{z}(k)|\mathbf{x}(k)) = \mathcal{N}(\mathcal{H}\mathbf{x}(k), \mathcal{R})$, where $\mathcal{H}$ is such that $\mathbf{Y}^{-1}\mathbf{y}(k) = \mathcal{H}\mathbf{x}(k)$. On the other hand, $P(\mathbf{x}(k)|Z(k)) = \mathcal{N}(\bar{\mathbf{x}}^{*}(k), \bar{\mathbf{P}}^{*}(k))$, where $\bar{\mathbf{x}}^{*}(k)$ and $\bar{\mathbf{P}}^{*}(k)$ are the fused predictions

at each node. In particular, given the restrictions in Problem 1, the fused prediction that ensures consistency (constraint 3) under unknown correlations (constraint 1) and one-hop communications (constraint 2) is given by the outer Löwner-John (LJ) method [John, 2014]. Among the different measures of ellipsoid size, $\text{tr}(\bar{\mathbf{S}}^{-1})$ in problem (4.4) minimises the MSE because

$$\text{MSE} = \mathbb{E}\left[\frac{1}{\mathsf{N}}\sum_{i=1}^{\mathsf{N}}||\hat{\mathbf{x}}_i^* - \mathbf{x}||^2\right] = \text{tr}(\text{block-diag}(\hat{\mathbf{P}}_1^*, \dots, \hat{\mathbf{P}}_N^*)) = \text{tr}(\hat{\mathbf{P}}^*) = \text{tr}((\hat{\mathbf{S}}^*)^{-1}).$$

By assumption of the theorem, the solution of (4.4) is optimal, so the associated unbiased optimal fused prediction at each node is $\bar{\mathbf{x}}_i^* = \bar{\mathbf{P}}_i^* \sum_{j\in\mathcal{J}_i} \lambda_{ij}^* \bar{\mathbf{P}}_j^{-1}\bar{\mathbf{x}}_j$, which is equal to Eq. (4.5).

Using the fundamental Gaussian identity (Appendix D in [Mahler, 2007]),

$$\begin{aligned}&\mathcal{N}(\mathcal{H}\mathbf{x}(k), \mathcal{R})\mathcal{N}(\bar{\mathbf{x}}^*(k), \bar{\mathbf{P}}^*(k)) = \\ &\mathcal{N}(\mathcal{H}\bar{\mathbf{x}}^*(k), \mathcal{R} + \mathcal{H}\bar{\mathbf{P}}^*(k)\mathcal{H}^\top)\mathcal{N}(\bar{\mathbf{x}}^*(k), ((\bar{\mathbf{P}}^*(k))^{-1} + \mathcal{H}\mathcal{R}^{-1}\mathcal{H}^\top)^{-1}).\end{aligned}$$

Besides, the integral

$$\int \mathcal{N}(\mathcal{H}\mathbf{x}(k), \mathcal{R})\mathcal{N}(\bar{\mathbf{x}}^*(k), \bar{\mathbf{P}}^*(k))d\mathbf{x}(k) = \mathcal{N}(\mathcal{H}\bar{\mathbf{x}}^*(k), \mathcal{R} + \mathcal{H}\bar{\mathbf{P}}^*(k)\mathcal{H}^\top),$$

and, therefore,

$$P(\mathbf{x}(k)|Z(k)) = \mathcal{N}(\hat{\mathbf{x}}^*(k), \hat{\mathbf{P}}^*(k)).$$

Finally, by definition,

$$(\hat{\mathbf{P}}^*(k))^{-1}\hat{\mathbf{x}}^*(k) = (\bar{\mathbf{P}}^*(k))^{-1}\bar{\mathbf{x}}^*(k) + \mathcal{H}\mathcal{R}^{-1}\mathbf{z}(k),$$

and

$$\hat{\mathbf{P}}^*(k)(\hat{\mathbf{P}}^*(k))^{-1}\hat{\mathbf{x}}^*(k) = \hat{\mathbf{x}}^*(k) = \bar{\mathbf{x}}^*(k) + \mathbf{K}(k)(\mathbf{z}(k) - \mathcal{H}\bar{\mathbf{x}}^*(k)).$$

Using Lemma 3, the update/correction operations are written in information form:

$$\begin{aligned}\hat{\mathbf{P}}^*(k) &= ((\bar{\mathbf{P}}^*(k))^{-1} + \mathcal{R}^{-1})^{-1} \\ \hat{\mathbf{x}}^*(k) &= \bar{\mathbf{x}}^*(k) + \bar{\mathbf{P}}^*(k)(\mathbf{y}(k) - \mathcal{R}^{-1})\bar{\mathbf{x}}^*(k)).\end{aligned} \tag{4.34}$$

Equations (4.33) and (4.34), together with optimisation problem (4.4), are the ECO-DKF filter in Algorithm 8. Besides, the expression in Eq. (4.26) can be reformulated as

$$\hat{\mathbf{x}}_i^* = \bar{\mathbf{x}}_i + \hat{\mathbf{P}}_i^*(\mathbf{y}_i - \mathbf{Y}_i\bar{\mathbf{x}}_i) + \gamma\hat{\mathbf{P}}_i^* \sum_{j\in\mathcal{N}_i} \lambda_{ij}^* \bar{\mathbf{P}}_j^{-1}(\bar{\mathbf{x}}_j - \bar{\mathbf{x}}_i) \tag{4.35}$$

with $\gamma = (\mathbf{I} - \hat{\mathbf{P}}_i^*\mathbf{Y}_i)\bar{\mathbf{P}}_i^*(\hat{\mathbf{P}}_i^*)^{-1}$. The latter is equivalent to Eq. (4.12) but weighting each term with the result of optimising (4.4).

Therefore, Algorithm 8 is optimal in the sense of Problem 8 and (4.4) provides the optimal consensus gain of Eq. (4.12). □

4.6 Event-triggered scheme

In this section we describe the proposed ET general scheme and we study its theoretical properties to show that the stability and optimality results are preserved under PJC of the network.

4.6.1 Description of the ET scheme

Under an ET scheme, a node should communicate when it possesses relevant information for the network. Our ET rule leverages the output of (4.4) to decide the triggering. Remember that (4.4) finds the optimal combination of estimates under positive certification. In particular, $\boldsymbol{\lambda}_i^*$ weights how important is each prediction in the optimal fused estimate. If at instant k, $\max(\boldsymbol{\lambda}_i^*) = \lambda_{ii}^*$, then the greatest contribution to $\bar{\mathbf{x}}_i^*$ is $\bar{\mathbf{x}}_i$, so the information at node i is the best in its neighbourhood and should be broadcast. The result is the ET rule in Eq. (4.6).

Some aspects must be underlined. On the one hand, at instant k, node i receives $|\mathcal{N}_i(k)|$ messages from broadcasting neighbours. Thus, the network topology is time-varying. On the other hand, two situations are possible in the ET rule at node i and instant k: $|\mathcal{N}_i(k)| > 0$ and $|\mathcal{N}_i(k)| = 0$. The former is the core of the ET rule because it determines the communication usage. The latter, despite influencing in the bandwidth as well, must be designed to avoid that the network converges to N individual KFs. As we will see later, this is key to assure stability.

4.6.2 Theoretical analysis

The two main properties analysed in Section 4.5 are stability and optimality. Regarding stability, the next Theorem shows that ECO-DKF is globally asymptotically stable under PJC.

Theorem 10. *ECO-DKF is global asymptotically stable for any time-varying graph sequence that satisfies PJC with $t < \infty$.*

Proof. The error dynamics η^+ in Theorem 8 now depends on time

$$\eta^+ = \mathbf{F}(k)\, \mathbf{L}_w(k)\, \mathcal{A}\eta, \tag{4.36}$$

From Theorem 8, since $\mathbf{L}_w(k)$ is row stochastic for any graph, the absolute value of its eigenvalues is always less than or equal to 1. Besides, $\mathbf{F}(k)$ is always a positive definite matrix and $\mathbf{I} \succeq \mathbf{F}(k)$ according to its definition in Lemma 3. Therefore, as a worst case scenario, we can consider $\mathbf{L}_w(k) = \mathbf{I} \Rightarrow \eta^+ = \mathbf{F}(k)\mathcal{A}\eta$.

Let j be one of the nodes. The error dynamics, η_j^+, are then determined by the eigenvalues and eigenvectors of $\mathbf{F}_j(k)\mathbf{A}$. The convergence rate of the KF is upper bounded by its steady-state gain $\mathbf{K}^{ss}$ [Grewal and Andrews, 2014], so let instead consider

$$\eta_j^+ = \mathbf{F}_j^{ss}\mathbf{A}\eta_j. \tag{4.37}$$

Let $\boldsymbol{v}_j = \{v_j^1, \ldots, v_j^r, \ldots, v_j^n\}$ and $V_j = \{\mathbf{v}_j^1, \ldots, \mathbf{v}_j^r, \ldots, \mathbf{v}_j^n\}$ be the eigenvalues and eigenvectors of $\mathbf{F}_j^{ss}\mathbf{A}$. The initial conditions are expressed as $\eta_j(0) = \sum_{r=1}^n \gamma_j^r \mathbf{v}_j^r$ and the norm of the error dynamics in Eq. (4.37) can be upper bounded by $||\eta_j^+(k)|| \leq \sum_{r=1}^n || \left(v_j^r\right)^k \gamma_j^r \mathbf{v}_j^r||$. Network observability implies that for each node j and eigenvalue v_j^r, there exist at least a

node j' such that $\kappa|\Re(v_j^r)| < 1$, with $\kappa \in \mathbb{R}_{>0}$ and $\Re(v_j^r)$ the real part of v_j^r. Otherwise, it would not be possible to reconstruct the state, even if the network was fully connected. In the worst case, due to the PJC with period equal to t, the estimation of node j' associated to v_j^r is updated through $t(\mathsf{N}-1)$ ECO-DKF instants before reaching node j. At each instant, ECO-DKF corrects the estimates in two main phases: the optimisation in (4.4) and the KF steps. The former consists in finding the intersection of $|\mathcal{N}_i(k) \cup \{i\}|$ ellipsoids. Since the intersection can not be larger than the smallest ellipsoid, then the result is upper bounded by the best estimate among estimates at instant k. Thus, in the worst case the estimation error associated to v_j^r remains that of j'. On the other hand, in the worst case, the estimation error associated to v_j^r is only predicted in the KF steps without using the measurements to perform the correction step of the filter, and the estimation error associated to v_j^r increases as a function of the eigenvalues of $\mathbf{A}$. Therefore, when the estimation of node j' associated to v_j^r reaches node j, the error associated to v_j^r is, in the worst case, the error for j' increased by a *constant* value that depends on the eigenvalues of $\mathbf{A}$ and $t(\mathsf{N}-1)$.

Since the estimation error associated to v_j^r in j' tends to zero when $k \to \infty$, then so does the error in node j. Applying this result to all the eigenvectors and nodes in the network, the result is proved. □

The next Proposition addresses optimality.

Proposition 11. *Theorem 9 holds under PJC and network-observability conditions.*

Proof. At instant k, node i receives $|\mathcal{N}_i(k)|$ messages and conducts optimisation (4.4) and certification (4.14) steps. The case where $|\mathcal{N}_i(k)| = 0$ is trivial and $\bar{\mathbf{x}}_i^*(k) = \bar{\mathbf{x}}_i(k)$ and $\bar{\mathbf{P}}_i^*(k) = \bar{\mathbf{P}}_i(k)$, according to restrictions 1 and 2 in Problem 1. In the general case, certifiability and optimality in Proposition 10 and Theorem 9 are proved for any $|\mathcal{N}_i(k)|$. Therefore, Theorem 9 holds irrespective of the connectivity and observability of the network. □

The previous results are interesting not only because they demonstrate that ECO-DKF is still optimal and stable under PJC, but also because this can be done without requiring any additional information. Regarding the ET rule (4.6), by letting each node to communicate with probability $p > 0$ when it does not receive information, we force that nodes do not isolate forever, and ET rule (4.6) guarantees PJC with an expected value of t bounded.

To complete the solution, we show that ECO-DKF is a consistent estimator under PJC conditions.

Proposition 12. *ECO-DKF is a consistent estimator under PJC conditions.*

Proof. Recalling the definition of (4.4), the outer LJ method gives a fused $\bar{\mathbf{S}}_i^*$ which is always consistent. Moreover, considering that we assume unbiased sensors, $\bar{\mathbf{x}}_i^*$ preserves consistency. This is because: (i) constraint (4.4c) ensures positive weights and their sum less or equal to 1, so it is a non-increasing combination of estimates; (ii) Lemma 5 shows

that the sum of the weights is always equal to 1, so it is a non-decreasing combination of estimates. Thus, the aggregation of estimates in (4.5) is also unbiased. Besides, Theorem 10 states that ECO-DKF is globally asymptotically stable under PJC conditions, which means that once estimates are aggregated, the update and prediction steps preserve consistency as well. Therefore, the whole algorithm is consistent. □

4.7 Illustrative results

We first evaluate the performance of the Time-Triggered (TT) ECO-DKF, i.e., ECO-DKF with $g_i(\boldsymbol{\lambda}_i^*(k-1), k) = 1$ for all i, k, to verify the stability and optimality properties. Then, different ET rules are compared, studying their impact in the communication bandwidth and convergence speed.

4.7.1 Time-triggered ECO-DKF vs time-triggered DKFs

We compare ECO-DKF in the TT setting with respect to other TT DKFs: (i) *AtA-ECO-DKF*, our proposal but assuming that $(i, j) \in \mathcal{E}(k)\ \forall i \neq j, k$, (ii) *OCDFK*, Algorithm 3 from [Olfati-Saber, 2007b] with consensus gain $\gamma_i(k) = 10^{-4}/(1 + ||\mathbf{M}_i(k)||_F)$ to ensure stability, with $||\bullet||_F$ the Frobenius norm, (iii) *TCDFK*, Algorithm 2 from [Talebi and Werner, 2019] with one consensus step for a fair comparison, (iv) *HDfKF*, complete algorithm in [Hu et al., 2011], (v) *HADfKF*, simplified algorithm in [Hu et al., 2011], (vi) *CKF*, centralised equivalent KF. The target system is the same 2D particle in a circular orbit used in [Sebastián et al., 2021c], with the same parameters.

We analyse two scenarios. Experiment 1 initialises a random sensor network, with appropriate parameters to obtain a sparse connected topology. Then, a random uniform distribution decides the quantities sensors measure among two options: measuring x or y. Note that local observability never holds, to test ECO-DKF in a network-observability setting. $\mathbf{H}_i$ is picked uniformly in the range $[1, 3]$. A Bernoulli distribution with $p = 0.5$ decides the diagonal of $\mathbf{R}_i$ in the range $[3, 5] \times 10^{-2}$ or $[3, 5]$, i.e., high-quality or low-quality. This is done 100 times, computing the averaged MSE:$= \frac{1}{100}\sum_{i=1}^{\mathsf{N}} \mathbb{E}[||\hat{\mathbf{x}}_i - \mathbf{x}||^2]$ over the experiments as in [Cattivelli and Sayed, 2010] and [Hu et al., 2011]. Experiment 2 is the same as Experiment 1, but only one sensor is of high quality, and sensors now can measure x, y or both.

The results of Experiments 1 are shown in Fig. 4.1a. The best performance among the distributed estimators is obtained by ECO-DKF, with a difference of more than an order of magnitude with the other state-of-the-art filters in steady-state MSE. Besides, ECO-DKF is also the fastest filter, with the fastest asymptotic convergence to the CKF. The OCDKF achieves similar MSE performance than the AtA-ECO-DKF: adding more neighbours hinders the optimization problem in ECO-DKF. The other consensus-based DKF, TCDKF, is far from them because it needs multiple consensus steps within instants for a good performance [Talebi and Werner, 2019]. The diffusion-based DKF, HDfKF and HADfKF, exhibit worse suboptimal performance as well. These differences also hold

in Experiment 2, as it is shown in Fig. 4.1b. Therefore, ECO-DKF obtains the best performance among DKFs.

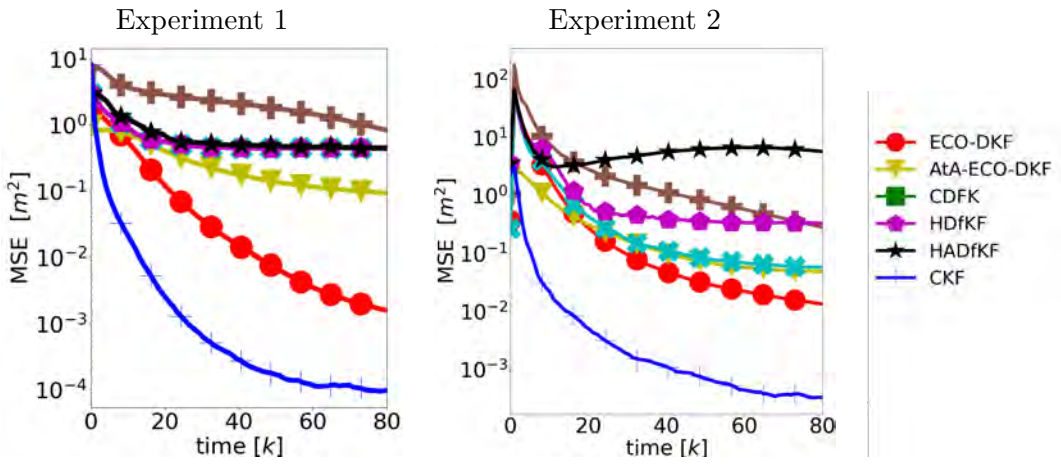


Figure 4.1: MSE for the different estimators in the TT case.

We have assessed the tightness of the relaxation by checking the certification values. A 99.33% of the times in Experiment 1 and 95.92% of the times in Experiment 2, $\mathbb{C}_i = \rho_i = 1$, so the SDP relaxation is *tight*. It is also noteworthy that the lower values of ρ are at 0.65, and this is only in the initialisation.

In relation with times, the computation of optimisations (4.4) and (4.14), run at each iteration, takes 50ms using a standard laptop. This is consistent with the election of the sample time $T = 100$ms. Problem (4.4) is solved in a polynomial number of iterations and arithmetic operations [Nesterov and Nemirovskii, 1994], so the computational cost scales with the number of neighbours. Whether it is preferable to compute more or communicate more will depend on the network resources, but in general, communicating multiple times may imply a total run time and risk of disturbances greater than ours. In this sense, we recall that ECO-DKF only communicates, at most, once per instant k.

4.7.2 Evaluation of event-triggered schemes in ECO-DKF

We now evaluate the impact of the ET rules in the NoB and MSE. We repeat Experiment 1 and 2 but with different ET rules: TT ECO-DKF (#O), the original TT ECO-DKF tested in Subsection 4.7.2; connected ECO-DKF (#C), which corresponds to ET rule (4.6) with $p(k) = 1\ \forall k$; disconnected ECO-DKF (#D), which corresponds to ET rule (4.6) with $p(k) = 0\ \forall k$; stochastic ECO-DKF (#S), which corresponds to ET rule (4.6) with $p(k) \sim \mathcal{NBB}(\alpha, \beta, n, k_l, k)$. Here, $\mathcal{NBB}(\alpha, \beta, n, k_l, k)$ is a normalised beta-binomial distribution whose shape is given by α, β, n. Given the shape, $p(k)$ increases when k_l increases, where $k_l \in \mathbb{N}$ is the number of instants that have passed since the last time $g_i(\boldsymbol{\lambda}_i^*) = 1$. The normalisation of the beta binomial is to force that, at some point, $p(k) = 1$, i.e., to force PJC. Finally, we also test a Jensen-Shannon (JS) divergence ECO-DKF (#J), which

corresponds to a SoD rule based on the JS divergence

$$g_i(\boldsymbol{\lambda}_i^*) = \begin{cases} 1, \text{ if } D_{JS}(\bar{\mathbf{S}}_i^*(k)||\bar{\mathbf{S}}_i^*(k-k')) > \tau \\ 0, \text{ otherwise} \end{cases} \forall i \in \mathcal{V} \tag{4.38}$$

where τ is a tuned threshold, k' is the last time $g_i(\boldsymbol{\lambda}_i^*) = 1$ and JS is a symmetric version of the KL divergence. Note that ET rule (4.38) implicitly depends on $\boldsymbol{\lambda}_i^*$ since it is the divergence between the latest computation of (4.4) and the latest time node ET rule (4.38) was equal to 1. For #S, we have calibrated two combinations of parameters: $\alpha = \{5, 2\}$, $\beta = \{1, 2\}$ and $n = \{10, 15\}$. For #J we have empirically tuned the threshold to be $\tau = 100$.

Figure 4.2 shows the evolution of NoB with time for the different ET rules, taking the average over the simulation runs. We can see that for #O, NoB = N always, since it is a TT version of ECO-DKF. In contrast, #C reduces to a half the NoB, which is a significant decrease with a very simple tuning-free ET rule. As expected, #D evolves to 0 as the network tends to be disconnected. Regarding the stochastic rules (#S), depending on the shape of the beta-binomial distribution the network converges to a different steady-state NoB. This also affects the NoB transient, in the sense that a sharper distribution at lower k' values imply less time to trigger, reason why in #S2 there is a very low NoB before $k = 30$. Finally, the Jensen-Shannon trigger protocol achieves a trade-off between TT ECO-DKF (in the first instants) and a connected ECO-DKF (for $k > 60$).

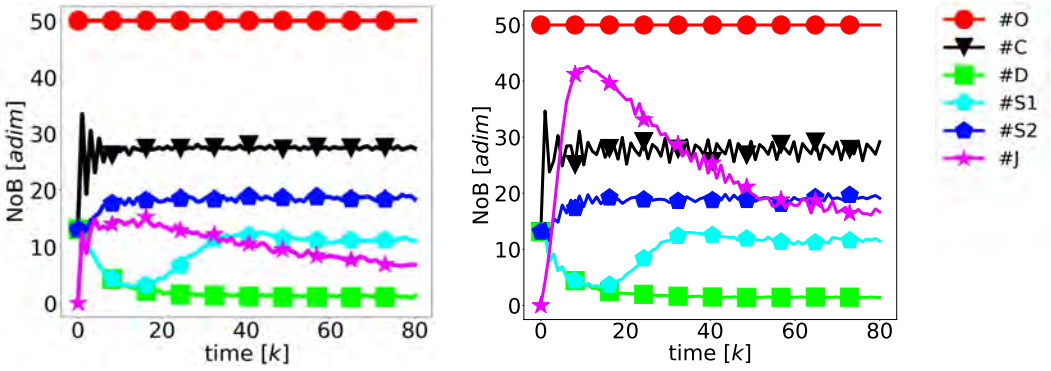


Figure 4.2: Evolution of NoB average over simulation runs with time for the different ET rules: (left) Experiment 1, (right) Experiment 2.

Efficiency in communications is followed by a non-degraded performance. In Fig. 4.3 we depict the MSE of the ET rules in Experiments 1 and 2. In the first instants, those protocols with more communications have the worst MSE (e.g., #O) due to the poor initial estimates. However, afterwards, the tendency is inverted, (e.g., #J). In steady-state, the improvement in communications given by #S1 and #S2 is followed by a slight improvement in MSE. #C has a good trade-off between MSE and NoB. #D is the worst estimator; indeed, #D failed 10 runs in Experiment 1 and 13 runs in Experiment 2. From Fig. 4.1, the MSE of ECO-DKF is the closest to the CKF. In fact, given the high number of positive certifications, ECO-DKF is almost always optimal in the TT version. For general ET cases, ECO-DKF obtains almost the same performance compared to the TT version (Fig. 4.3), so the ET rule does not significantly affect the performance.

Finally, in Table 4.1 we evaluate certifiability and optimality. To do so, those cases where $|\mathcal{N}_i(k)| = 0$ are removed from $\mathbb{C}_i$ and ρ_i since they are trivial to compute, denoted

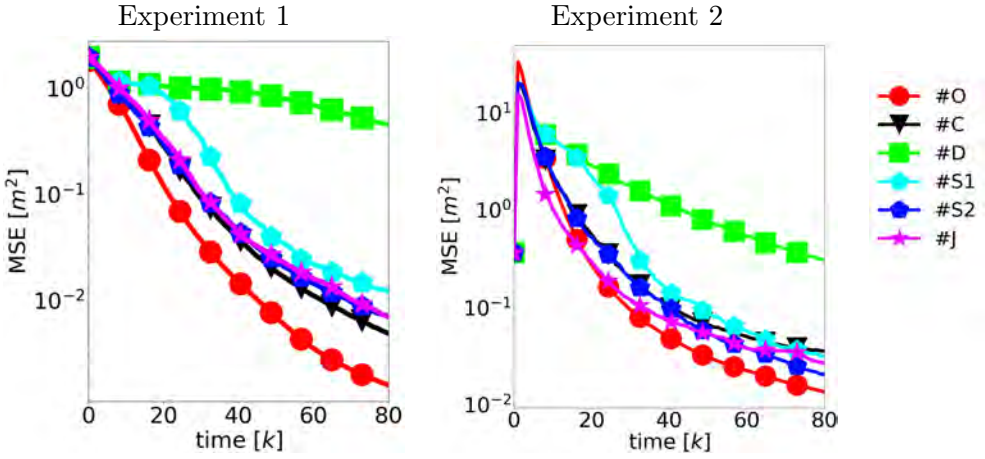


Figure 4.3: MSE for the different estimators in the ET case.

Experiment 1

ET rule	Avg($\mathbb{C}_i^f = \rho_i^f = 1$)
#O	99.33%
#C	79.30%
#D	11.80%
#S1	44.11%
#S2	69.88%
#J	47.69%

Experiment 2

ET rule	Avg($\mathbb{C}_i^f = \rho_i^f = 1$)
#O	95.92%
#C	77.74%
#D	12.11%
#S1	44.14%
#S2	69.16%
#J	67.24%

Table 4.1: Certification results during the ET simulations.

by $\mathbb{C}_i^f$ and ρ_i^f respectively. The results are averaged over the sensors of the network at each instant. Apart from #O, the only ET rule which resists with reasonable certification levels is #C: the disconnected rule hardly ever certifies as optimal; #S1 and #S2 have worse certification results; and #J has a large variability among experiments, so it is hard to predict its certification performance.

4.8 Conclusions

This chapter has presented ECO-DKF, the first event-triggered and certifiable optimal DKF. It has solved two main problems regarding DKFs: optimality in the estimation and reduction of the communication bandwidth. The outer LJ fuses neighbouring estimates with certifiable guarantees of optimality. The output is integrated in an information DKF, achieving a stable filter on heterogeneous sensor networks, with minimal message size and no tuning. We have proved global asymptotic stability of the estimator and optimality under positive certification. Moreover, a novel ET theory has been derived from the outer LJ method output. Under PJC, ECO-DKF preserves the properties of the algorithm. The ET rule is inexpensive to compute, and avoids individual communication links and multiple information exchanges. ECO-DKF surpasses the existing state-of-the-art DKFs while decreasing the communication bandwidth usage. The improved capabilities of ECO-

DKF are specially suited to distributed multi-robot systems, where the stochastic filter is part of a control module that relies on accurate state estimation and the restrictions in communication bandwidth demands efficiency.

Chapter 5

Distributed Outer Approximation of the Intersection of Ellipsoids

The certifiable optimal distributed Kalman filter developed in the previous chapter motivates an important question from a distributed stochastic estimation perspective. The filter successfully applies the outer Löwner-John ellipsoid method locally at each node of the network. However, sensor fusion could be improved if the local outer approximation considered not only the neighboring covariances, but also the rest of the covariances of the network. By doing so, the estimation would be refined, even though cross-correlations are still considered unknown. Besides, beyond a Kalman filter setting, one fundamental problem in control systems is how to approximate system measurements, states, or constraints such that their essential features are preserved, being simple enough to be handled. One of the most popular representations is ellipsoids [Lasserre, 2015], characterized by symmetric and positive semi-definite n-dimensional matrices. Ellipsoids are chosen to describe the conservative shape of an obstacle [Malyuta et al., 2021], obstacle-free regions [Deits and Tedrake, 2015], estimated quantities with their associated uncertainty [Sebastián et al., 2023b], very large multi-agent populations [Saravanos et al., 2023], or patterns and geometrical objects to be identified and clustered [Yang and Carlone, 2022]. All these examples are related to or can be posed as finding the tightest outer ellipsoid approximating a convex set [Vandenberghe and Boyd, 1996]. Therefore, the question is: is it possible to compute the equivalent centralized optimum of the outer Löwner-John ellipsoid method across the network using distributed techniques?

In this chapter we propose two distributed algorithms to compute the solution of the original centralized outer Löwner-John method. Both of them are the first of their kind. We first develop a continuous-time distributed protocol that operates in static settings, where the input ellipsoids at each node remain constant throughout time. The limitations of the first algorithm in terms of applicability to time-varying settings and implementability in a processing unit motivate the second algorithm, a distributed discrete-time dynamic algorithm that deals with both static cases and time-varying cases. The chapter also discusses how to integrate the discrete-time algorithm in sensor fusion applications to enhance distributed stochastic estimation algorithms.

5.1 Related work

The Löwner-John (LJ) methods [Henrion et al., 2001] are a series of ellipsoidal approximations for convex sets. Of particular interest are the convex sets generated by the intersection of n-dimensional ellipsoids, described by symmetric and positive semi-definite n-dimensional matrices. LJ methods have a significant presence in a wide variety of applications [Lasserre, 2015], such as robust control [Henrion and Lasserre, 2011] or statistical analysis [Lu et al., 2018]. Specially important is in the field of data fusion and state estimation [Julier and Uhlmann, 2017], where the ellipsoids represent the measurements or estimates' uncertainty. However, there are no distributed algorithms to compute LJ ellipsoids, despite their potential application in sensor networks where the measurements are scattered over a communication network [Sebastián et al., 2021c]. The outer Löwner-John method has three main characteristics: (i) the objective function is not necessarily separable, so it cannot be expressed as the sum of local functions; (ii) some of the constraints couple all the optimization variables; and (iii) the optimization variables are symmetric positive definite variables. All together, these features lead to a challenging problem from a distributed algorithmic perspective.

Many distributed optimization solutions depart from a centralized optimization formulation and then derive consensus-based algorithms to reconstruct global quantities and converge to the global minimizer [Notarstefano et al., 2019]. When the objective function is separable as a sum of local (strongly) convex smooth functions, dynamic consensus over the optimization variables and the gradients [Carnevale et al., 2023c] can be exploited to design distributed versions of gradient descent [Nedić and Olshevsky, 2014], general first order optimization methods [Van Scoy and Lessard, 2022] or second order Newton-Raphson-like methods [Varagnolo et al., 2015]. Despite proving the success of consensus-based approaches for distributed optimization, these solutions are not suitable when the objective function cannot be decomposed into local smooth convex functions.

To deal with constraints, some works propose extensions of consensus-based algorithms that rely on projection methods [Mai et al., 2023], population dynamics equations [Martinez-Piazuelo et al., 2022], or subgradient methods [Romao et al., 2021]. The most popular alternative to address global coupling constraints is the use of a primal and/or dual proximal stage at each node [Notarnicola and Notarstefano, 2019]. In particular, the alternating direction method of multipliers [Bastianello et al., 2022] and the Douglas-Rachford splitting [Bredies et al., 2022] allow to handle equality coupling constraints among nodes by solving simultaneously the primal and dual optimization problem. Typically, these distributed proximal methods either assume separability of the original objective function [Meng et al., 2015] and/or are restricted to scalar quantities [Parikh and Boyd, 2014]. Therefore, they are not suitable for distributing the outer Löwner-John method. In the presence of constraints and a linear objective function, it is possible to find distributed semi-definite problem reformulations [Li et al., 2021b] that are solved through primal-dual methods. With a similar spirit to our work, [Bürger et al., 2013] proposes a polyhedral outer approximation in a distributed optimization context. Nevertheless, in this case the polyhedral approximation represents a convex constraint set rather than the

optimization objective.

To address the non-separability of the objective function [Yang et al., 2019], some works assume that the optimization variables can be divided in two subsets [Wendell and Hurter Jr, 1976], which is not possible in the outer Löwner-John method since it entails an ellipsoidal object and not scalars. Thus, it is common to develop ad hoc solutions that are based on successive approximations [Scutari and Sun, 2019] or decompositions [Meselhi et al., 2022], again not suitable for symmetric positive semi-definite matrix variables.

5.2 Preliminaries

Consider a network of N agents which communicate according to an undirected graph $\mathcal{G} = (\mathcal{I}, \mathcal{F})$, where $\mathcal{I} = \{1, \dots, \mathsf{N}\}$ is the vertex set and $\mathcal{F} \subset \mathcal{I} \times \mathcal{I}$ is the edge set representing the communication links between agents. The set of neighbors of agent i, including i, is $\mathcal{N}_i = \{j \in \mathcal{I} | (i,j) \in \mathcal{F}\} \cup \{i\}$. Each agent i is described by a $n-$dimensional ellipsoidal set $\mathcal{E}(\mathbf{P}_i) = \{\mathbf{y} \in \mathbb{R}^n : \mathbf{y}^\top \mathbf{P}_i^{-1} \mathbf{y} \leq 1\}$, given $\mathbf{0} \prec \mathbf{P}_i \in \mathbb{R}^{n \times n}$.

The problem addressed in this chapter is considered from a continuous-time and a discrete-time perspective, leading to two different problem formulations.

5.2.1 Continuous-time problem formulation

The goal for the agents is to cooperate to find an ellipsoidal set $\mathcal{E}(\mathbf{P})$ covering the intersection $\check{\mathcal{E}} := \bigcap_{i=1}^{\mathsf{N}} \mathcal{E}(\mathbf{P}_i)$, characterized by some $\mathbf{P} \succ 0$. A wide family of ellipsoidal sets covering $\check{\mathcal{E}}$ is the one parameterized by $\mathbf{P}(\boldsymbol{\lambda})^{-1} := \sum_{i=1}^{\mathsf{N}} \lambda_i \mathbf{P}_i^{-1}$ with $\sum_{i=1}^{\mathsf{N}} \lambda_i = 1$ and $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_{\mathsf{N}}]^\top \in \mathbb{R}^{\mathsf{N}}_{\geq 0}$. It can be verified that $\mathcal{E}(\mathbf{P}(\boldsymbol{\lambda})) \supset \check{\mathcal{E}}$ [Henrion et al., 2001]. Therefore, designing $\mathbf{P}(\boldsymbol{\lambda})^{-1}$ as a convex combination of $\{\mathbf{P}_i^{-1}\}_{i=1}^{\mathsf{N}}$ defines a family of LJ outer ellipsoids for the intersection of $\{\mathcal{E}(\mathbf{P}_i)\}_{i=1}^{\mathsf{N}}$. The weights $\boldsymbol{\lambda}$ can then be optimized by solving the following optimization program:

$$\min_{\boldsymbol{\lambda} \in \mathbb{R}^{\mathsf{N}}_{\geq 0}} f(\mathbf{P}(\boldsymbol{\lambda})^{-1}), \quad \text{such that} \quad \textstyle\sum_{i=1}^{\mathsf{N}} \lambda_i = 1, \tag{5.1}$$

where $f(\bullet)$ is a function that measures the size of $\mathcal{E}(\mathbf{P}(\boldsymbol{\lambda}))$. For instance, $f(\bullet) = \log(\det(\bullet))$ can be used to minimize the volume of $\mathcal{E}(\mathbf{P}(\boldsymbol{\lambda}))$. Other popular choices of $f(\bullet)$ are in Table 5.1. Due to the numerical difficulties found when dealing with equality constraints in practice, we recast (5.1) into

$$\min_{\mathbf{x} \in \mathcal{M}} f(\mathbf{Q}(\mathbf{x})), \quad \mathbf{Q}(\mathbf{x}) := 1/\mathsf{N} \textstyle\sum_{i=1}^{\mathsf{N}} x_i^2 \mathbf{P}_i^{-1}, \tag{5.2}$$

where instead of an equality constraint, the solution is restricted to a wider feasible manifold $\mathcal{M} = \{\mathbf{x} \in \mathbb{R}^{\mathsf{N}} : 1 - \varepsilon \leq s(\mathbf{x}) \leq 1\}$ of $\mathbf{x} = [x_1, \dots, x_{\mathsf{N}}]^\top$, $s(\mathbf{x}) := \|\mathbf{x}\|^2/\mathsf{N}$ and $\varepsilon \in (0,1)$. By using the change of coordinates $\lambda_i = x_i^2/\mathsf{N}$, we ensure $\lambda_i \in \mathbb{R}_{\geq 0}$ and that the unique minimizer of (5.1) follows from the minimizer of (5.2) when $\varepsilon = 0$. For any other $\varepsilon > 0$, we use (5.2) to approximate solutions of (5.1) with arbitrary accuracy dictated by the size of ε. The first goal of the chapter is to provide a distributed algorithm to solve (5.2).

5.2.2 Discrete-time time-varying problem formulation

At each discrete-time step $k \in \{0, 1, \dots\}$, each agent $i \in \mathcal{I}$ is endowed with a positive definitive *input* matrix $\mathbf{P}_i[k] \in \mathbb{S}^n_{++}$. Note that in this case we allow for a time-varying evolution of the input matrices. In practice, the input matrices $\mathbf{P}_i[k]$ cannot have arbitrarily large values, and the variation of $\mathbf{P}_i[k]$ from instant k to $k+1$ cannot be arbitrarily large, so in this work we do the following assumption.

Assumption 2. *There exist $0 < \underline{p} \leq \overline{p}$ such that the input matrices $\mathbf{P}_i[k]$, $\forall i \in \mathcal{I}$ and $\forall k \in \{0, 1, \dots\}$, satisfy*

$$\underline{p}\mathbf{I} \preceq \mathbf{P}_i[k] \preceq \overline{p}\mathbf{I}$$

The previous inequalities provide bounds for the eigenvalues of $\mathbf{P}_i[k]$ for all time.

The goal for all the agents is to agree on a matrix $\mathbf{Q}^*[k]$ that represents the tightest outer approximation of the intersection of the ellipsoids $\mathcal{E}(\mathbf{P}_i[k]^{-1})$. Intuitively, $\mathbf{Q}^*[k]$ represent the information matrix for such covering ellipsoid. Formally, $\mathbf{Q}^*[k]$ is the result of the following optimization problem:

$$\begin{aligned} \mathbf{Q}^*[k] &= \underset{\mathbf{Q} \in \mathcal{C}^*[k]}{\arg\min}\, f(\mathbf{Q}) \\ \mathcal{C}^*[k] &= \left\{ \mathbf{Q} \in \mathbb{S}^n_+ : \exists \lambda_1, \dots, \lambda_{\mathsf{N}} \in [0,1], \sum_{j=1}^{\mathsf{N}} \lambda_j \leq 1, \mathbf{0} \preceq \mathbf{Q} \preceq \sum_{j=1}^{\mathsf{N}} \lambda_j \mathbf{P}_j[k]^{-1} \right\} \end{aligned} \tag{5.3}$$

where we make the following assumption on the function f:

Assumption 3. *The function $f : \mathbb{S}^n_+ \to \mathbb{R}$ satisfies:*

1. *f has continuous derivative over all $\mathbb{S}^n_+$.*
2. *f is strictly convex, meaning that, for any $\lambda_1, \lambda_2 < 0$, $\mathbf{Q}_1, \mathbf{Q}_2 \in \mathbb{S}^n_+$ with $\lambda_1 + \lambda_2 = 1$ it follows that: $f(\lambda_1 \mathbf{Q}_1 + \lambda_2 \mathbf{Q}_2) \leq \lambda_1 f(\mathbf{Q}_1) + \lambda_2 f(\mathbf{Q}_2)$.*
3. *f has no lower bound over $\mathbb{S}^n_+$, meaning that $\inf_{\mathbf{Q} \in \mathbb{S}^n_+} f(\mathbf{Q})$ does not exists.*
4. *f is bounded over any closed bounded convex set $\mathcal{C} \subset \mathbb{S}^n_+$ meaning that $\inf_{\mathbf{Q} \in \mathcal{C}} f(\mathbf{Q}) \in \mathbb{R}$.*

Assumption 3 holds for many popular choices of $f(\bullet)$ related to the volume of ellipsoids or information theoretic measures, such as $\log(\det((\bullet)^{-1}))$ or $\mathrm{tr}((\bullet)^{-1})$. Moreover, Assumption 3-4) is an instrumental property to be used in subsequent analysis. The second goal of the chapter is to design an algorithm that solves (5.3) in a distributed manner.

5.3 Distributed continuous-time ellipsoid computation

To solve problem (5.2) in a distributed fashion, we propose a novel distributed protocol based on Projected Gradient Flow (PGF) [Hauswirth et al., 2016] methods. The idea is to

μ	$f(\mathbf{Q}(\mathbf{x}))$	$g_i(x_i, \mathbf{Q}(\mathbf{x}))$
0	$-\mathrm{tr}(\mathbf{Q}(\mathbf{x}))$	$-2x_i\mathrm{tr}(\mathbf{P}_i^{-1})$
1	$\log(\det(\mathbf{Q}(\mathbf{x})^{-1}))$	$-2x_i\mathrm{tr}(\mathbf{Q}(\mathbf{x})^{-1}\mathbf{P}_i^{-1})$
2	$\mathrm{tr}(\mathbf{Q}(\mathbf{x})^{-1})$	$-2x_i\mathrm{tr}(\mathbf{Q}(\mathbf{x})^{-1}\mathbf{P}_i^{-1}\mathbf{Q}(\mathbf{x})^{-1})$

Table 5.1: Different measure functions $f(\bullet)$ with $\mathbf{Q}(\mathbf{x}) = (1/\mathsf{N})\sum_{i=1}^{\mathsf{N}} x_i^2\mathbf{P}_i^{-1}$ as well as the components of the gradient $\nabla f(\mathbf{Q}(\mathbf{x})) = [g_1(x_1, \mathbf{Q}(\mathbf{x})), \ldots, g_{\mathsf{N}}(x_{\mathsf{N}}, \mathbf{Q}(\mathbf{x}))]^\top/\mathsf{N}$. We label each choice by μ for its reference in Theorem 11.

drive the trajectories of $\mathbf{x}(t)$ towards the feasible manifold where the equality constraint in (5.1) holds. Once there, the trajectories of $\mathbf{x}(t)$ flow towards the optimum while fulfilling the equality constraint. To do so, we set a suitable virtual control action $\dot{x}_i(t) = u_i(t)$ using only local information.

To design a PGF-based protocol, we first need agreement on some global quantities across the nodes of the network. Therefore, the first stage of our algorithm computes local estimates $\{\hat{s}_i(t), \hat{\mathbf{Q}}_i(t)\}_{i=1}^{\mathsf{N}}$ for $s(\mathbf{x}(t)), \mathbf{Q}(\mathbf{x}(t))$ using the following Exact Dynamic Consensus (EDC) protocols:

$$\begin{aligned}
\hat{s}_i(t) &= x_i(t)^2 - v_i(t) \\
\dot{v}_i(t) &= \kappa_s \textstyle\sum_{j\in\mathcal{N}_i} \phi\left(\hat{s}_j(t) - \hat{s}_i(t); \zeta_s; q\right) \\
\hat{\mathbf{Q}}_i(t) &= x_i(t)^2\mathbf{P}_i^{-1} - \mathbf{V}_i(t) \\
\dot{\mathbf{V}}_i(t) &= \kappa_{\mathbf{Q}} \textstyle\sum_{j\in\mathcal{N}_i} \phi\left(\hat{\mathbf{Q}}_j(t) - \hat{\mathbf{Q}}_i(t); \zeta_{\mathbf{Q}}; q\right)
\end{aligned} \tag{5.4}$$

with auxiliary variables $\mathbf{V}_i(t), v_i(t)$ initialized as $\mathbf{V}_i(0) = \mathbf{0}, v_i(0) = 0$. Moreover, the gains $\kappa_{\mathbf{Q}}, \kappa_s, \zeta_{\mathbf{Q}}, \zeta_s > 0, q \in (0,1)$ are design parameters. We define the function $\phi(\bullet; \zeta; q) = (|\bullet|^{1-q} + |\bullet|^{1+q} + \zeta)\mathrm{sign}(\bullet)$ for a scalar parameter $\bullet$, and component-wise for $\bullet$ of any other dimension. After a transient of prescribed duration T_c, these estimates will comply $\hat{\mathbf{Q}}_i(t) \equiv \mathbf{Q}(\mathbf{x}(t)), \hat{s}_i(t) \equiv s(\mathbf{x}(t)), \forall t \geq T_c$ for suitable $\kappa_{\mathbf{Q}}, \kappa_s$. This is possible since $\zeta_{\mathbf{Q}}, \zeta_s$ introduce a discontinuous sliding mode term in ϕ allowing (5.4) to achieve EDC even with time-varying consensus inputs. Then, all agents estimate its local component of the gradient $g_i(x_i(t), \mathbf{Q}(\mathbf{x}(t))) \equiv g_i(x_i(t), \hat{\mathbf{Q}}_i(t)), \forall t \geq T_c$, as in Table 5.1.

During the first consensus stage defined in (5.4), we set a control action $u_i(t) = 0$, for all time in $[0, T_c]$. The second stage of our algorithm consists of taking the arbitrary initial conditions $x_i(0) = x_i(T_c)$ and update $x_i(t)$ towards $\mathcal{M}$. Then, nodes do PGF to find the global optimum of (5.2). This is achieved by a discontinuous controller. For all $t \geq T_c$:

$$u_i(t) = \begin{cases} \kappa_{\mathcal{M}} x_i(t)\mathrm{sign}\left(\left(1-\frac{\varepsilon}{2}\right) - \hat{s}_i(t)\right), & \hat{s}_i(t) \notin [1-\varepsilon, 1] \\ -\kappa_{\mathcal{M}} g_i(x_i(t), \hat{\mathbf{Q}}_i(t)), & \hat{s}_i(t) \in [1-\varepsilon, 1] \end{cases} \tag{5.5}$$

with design parameter $\kappa_{\mathcal{M}} > 0$, which leads to:

$$\dot{\mathbf{x}}(t) = \begin{cases} \kappa_{\mathcal{M}}\mathbf{x}(t)\mathrm{sign}\left(\left(1-\frac{\varepsilon}{2}\right) - s(\mathbf{x}(t))\right), & \mathbf{x}(t) \notin \mathcal{M} \\ -\kappa_{\mathcal{M}}\mathsf{N}\nabla f(\mathbf{Q}(\mathbf{x}(t))), & \mathbf{x}(t) \in \mathcal{M} \end{cases} \tag{5.6}$$

under the synchronization conditions $\hat{\mathbf{Q}}_i(t) \equiv \mathbf{Q}(\mathbf{x}(t))$, $\hat{s}_i(t) \equiv s(\mathbf{x}(t))$. Using the idea of PGF, when $\mathbf{x}(t) \notin \mathcal{M}$, $\mathbf{x}(t)$ flows towards $\mathcal{M}$, in order to fulfill the equality constraint of the outer LJ method in (5.1). On the other hand, for $\mathbf{x}(t) \in \mathcal{M}$, $\mathbf{x}(t)$ flows in the direction opposite to the gradient, towards the optimum of the outer LJ method in (5.1). This results in $\mathbf{x}(t)$ flowing in the direction of the projected gradient of $f(\mathbf{Q}(\mathbf{x}(t)))$ with respect to $\mathcal{M}$ in any case, allowing to maintain feasible trajectories and converge to the optimum. In the following, we state our main result as well as a practical assumption under which it holds.

Assumption 4. *Let $0 < \underline{b} < \overline{b}, \overline{b} > 1$ and $\underline{\sigma}, \overline{\sigma} > 0$. Then, $x_i(0) \in [\underline{b}, \overline{b}]$ and $\underline{\sigma}\mathbf{I} \preceq \mathbf{P}_i^{-1} \preceq \overline{\sigma}\mathbf{I}, \forall i \in \mathcal{I}$, and with p the maximum scalar component among all $\{\mathbf{P}_i\}_{i=1}^{\mathsf{N}}$.*

Theorem 11. *Let $\mathcal{G}$ be a connected undirected graph with N nodes, ℓ edges, algebraic connectivity $\lambda_{\mathcal{G}}$, and consider protocols (5.4). Let $\kappa_{\mathcal{M}} > 0, \varepsilon \in (0,1)$, and Assumption 4 hold. Let $f(\bullet)$ in Table 5.1 be labeled by $\mu \in \{0,1,2\}$ and*

$$h(\mathsf{N}) = \kappa_{\mathcal{M}} \max\{\sqrt{\mathsf{N}}\overline{b}, 2\overline{b}\overline{\sigma}\mathsf{N}^{\mu+1}(\underline{\sigma}\min\{\underline{b}^2, 1-\varepsilon\})^{-\mu}\}.$$

Let $\dot{x}_i(t) = u_i(t)$ with $u_i(t) = 0, \forall t \in [0, T_c]$ and $u_i(t)$ defined as in (5.5) for $t \geq T_c$. For any $\kappa_{\mathcal{M}} > 0$, if

$$\kappa_s, \kappa_{\mathbf{Q}} > \frac{\ell\pi}{q\lambda_{\mathcal{G}} T_c},\ \zeta_s > \frac{4\overline{b}h(\mathsf{N})}{\kappa_s\sqrt{\lambda_{\mathcal{G}}}},\ \zeta_{\mathbf{Q}} > \frac{4p\overline{b}h(\mathsf{N})}{\kappa_{\mathbf{Q}}\sqrt{\lambda_{\mathcal{G}}}}, \tag{5.7}$$

then, there exists $T_\varepsilon > 0$ such that $\mathbf{x}(t) \in \mathcal{M}$, $\forall t \geq T_c + T_\varepsilon$. In addition,

$$\mathcal{E}(\mathbf{Q}(\mathbf{x}(t))^{-1}) \supset \check{\mathcal{E}}, \forall t \geq T_c + T_\varepsilon$$

and $f(\mathbf{Q}(\mathbf{x}(t)))$ converges asymptotically towards the optimum of (5.2).

Remark 2. *After consensus has been reached for $t \geq T_c$, each agent can check the condition $\hat{s}_i(t) = s(\mathbf{x}(t)) \in [1-\varepsilon, 1]$ to verify if $\mathbf{x}(t) \in \mathcal{M}$ and compute T_ε. Hence, the proposed algorithm obtains an outer LJ ellipsoid from $t = T_c + T_\varepsilon$, since $\mathbf{x}(t) \in \mathcal{M}\ \forall t \geq T_c + T_\varepsilon$. Thus, the ellipsoid $\mathcal{E}(\hat{\mathbf{Q}}_i(t)^{-1}) = \mathcal{E}(\mathbf{Q}(\mathbf{x}(t))^{-1})$ is already valid to use. Note that the greater t, the tighter $\mathcal{E}(\mathbf{Q}(\mathbf{x}(t))^{-1})$, so the user can design the algorithm depending on the computing resources available and the desired accuracy for the LJ ellipsoid.*

Remark 3. *The powers $1-q, 1+q$ with $q \in (0,1)$ in ϕ are known to induce a fixed settling time bound, regardless of the initial conditions. This allows all agents to make sure consensus has been reached before $t = T_c$ without having to individually check this condition, avoiding additional synchronization issues. The function ϕ along with q and κ_s, ζ_s (similarly $\kappa_{\mathbf{Q}}, \zeta_{\mathbf{Q}}$) were introduced for static consensus with disturbances in [Aldana-López et al., 2019].*

Before proving Theorem 11, we need some auxiliary results. Systems (5.4) is discontinuous due to the use of the sign$(\bullet)$ function and the discontinuous nature of (5.6) at the boundary of $\mathcal{M}$. Hence, these systems are better understood in the sense of Filippov [Filippov, 1988], devised to study discontinuous dynamics.

We start by analyzing the consensus protocols in (5.4). Note that all these protocols have the structure:

$$\dot{\upsilon}_i(t) = \kappa_z \sum_{j \in \mathcal{N}_i} \phi\left(\hat{z}_j(t) - \hat{z}_i(t); \zeta, q\right), \hat{z}_i(t) = z_i(t) - \upsilon_i(t) \tag{5.8}$$

for suitable input $z_i(t)$, either $x_i(t)^2$ or the elements of $x_i(t)^2\mathbf{P}_i^{-1}$. Now, we show convergence of (5.8) before $t = T_c$.

Proposition 13. *[Aldana-López et al., 2019, Special case of Theorem 6] Let $\mathcal{G}$ be a connected undirected graph with N nodes, ℓ edges, algebraic connectivity $\lambda_{\mathcal{G}}$, and consider*

$$\dot{e}_i(t) = d_i(t) - \kappa_z \textstyle\sum_{j\in\mathcal{N}_i} \phi(e_j(t) - e_i(t); \zeta, q) \tag{5.9}$$

where $|d_i(t)| \leq L', \forall t \geq 0$. Moreover, given $T_c > 0$ let $\kappa_z \geq \ell\pi/(q\lambda_{\mathcal{G}}T_c), \zeta \geq L'/(\kappa_z\sqrt{\lambda_{\mathcal{G}}})$. Then, $e_i(t) \equiv (1/\mathsf{N})\sum_{i=1}^{\mathsf{N}} e_i(0), \forall t \geq T_c$.

Lemma 6. *Let $\mathcal{G}$ be a connected undirected graph with N nodes, ℓ edges, algebraic connectivity $\lambda_{\mathcal{G}}$ and $\sum_{i=1}^{\mathsf{N}} \upsilon_i(0) = 0$. Moreover, let $\bar{z}(t) = (1/\mathsf{N})\sum_{i=1}^{\mathsf{N}} z_i(t)$ and $|\dot{z}_i(t)| \leq L, \forall t \geq 0$. Then, (5.8) satisfies that $\hat{z}_i(t) \equiv \bar{z}(t), \forall t \geq T_c$ and $\forall i$, provided that $\kappa_z \geq \ell\pi/(q\lambda_{\mathcal{G}}T_c)$ and $\zeta \geq 2L/(\kappa_z\sqrt{\lambda_{\mathcal{G}}})$.*

Proof. Let $e_i(t) = \upsilon_i(t) - (z_i(t) - \bar{z}(t))$ such that:

$$\dot{e}_i(t) = (\dot{\bar{z}}(t) - \dot{z}_i(t)) - \kappa_z \sum_{j\in\mathcal{N}_i} \phi\left(e_j(t) - e_i(t); \zeta, q\right)$$

equivalent to (5.9) with $d_i(t) = \dot{\bar{z}}(t) - \dot{z}_i(t)$. By assumption, $|d_i(t)| \leq 2L$. Thus, Proposition 13 is used with $L' = 2L$ to conclude that $e_i(t) \equiv (1/\mathsf{N})\sum_{i=1}^{\mathsf{N}} e_i(0) = 0, \forall t \geq T_c$. Hence, $\upsilon_i(t) = z_i(t) - \bar{z}(t)$ and $\hat{z}_i(t) = \bar{z}(t)$ for $t \geq T_c$. □

Now, we study the ideal PGF trajectories for $\mathbf{x}(t)$, dictated by (5.6). The following result shows that $\mathbf{x}(t)$ converge to the feasible region $\mathcal{M}$, a requirement for a solution of (5.2).

Lemma 7. *Let (5.6) and $\underline{b} > 0$. Then, $\forall \mathbf{x}(T_0) \notin \mathcal{M}$, $x_i(T_0) \geq \underline{b}$ there exist $T_\varepsilon > 0$ such that $\mathbf{x}(t) \in \mathcal{M}, \forall t \geq T_0 + T_\varepsilon$.*

Proof. We split the proof in two cases:

- (a) $(1-\varepsilon/2) - s(\mathbf{x}(T_0)) > 0$: in this case, note that $\dot{x}_i(t) = \kappa_{\mathcal{M}} x_i(t)$ for $t \in [T_0, T_0 + T')$ with $T' = \inf\{t > T_0 : \mathbf{x}(t) \in \mathcal{M}\}$. Thus, $x_i(t)$ is increasing in such interval and $\|\mathbf{x}(t)\| \geq \sqrt{\mathsf{N}\min_{i\in\mathcal{I}} x_i(t)^2} \geq \sqrt{\mathsf{N}}\underline{b}, \quad \forall t \in [T_0, T_0 + T')$. Now, consider a Lyapunov function candidate $V_1(\mathbf{x}(t)) = (1 - \varepsilon) - s(\mathbf{x}(t))$, whose time derivative is

  $$\dot{V}_1(\mathbf{x}(t)) = -(2/\mathsf{N})\mathbf{x}(t)^\top \dot{\mathbf{x}}(t) = -(2/\mathsf{N})\kappa_{\mathcal{M}}\|\mathbf{x}(t)\|^2 \leq -2\kappa_{\mathcal{M}}\underline{b}^2.$$

  Thus, integrating from T_0 to t, $V_1(\mathbf{x}(t)) \leq V_1(\mathbf{x}(T_0)) - 2\kappa_{\mathcal{M}}\underline{b}^2(t - T_0)$ which satisfies $V_1(\mathbf{x}(T_0 + T_\varepsilon)) = 0$ with $T' \leq T_\varepsilon := V_1(\mathbf{x}(T_0))/(2\kappa_{\mathcal{M}}\underline{b}^2)$. Hence, $s(\mathbf{x}(T_0 + T_\varepsilon)) = 1 - \varepsilon$ and thus $\mathbf{x}(T_0 + T_\varepsilon) \in \mathcal{M}$.

- (b) $(1 - \varepsilon/2) - s(\mathbf{x}(T_0)) < 0$: note that $\dot{x}_i(t) = -\kappa_{\mathcal{M}} x_i(t)$ implies $x_i(t)$ is decreasing for $t \in [T_0, T_0 + T_\varepsilon)$ and thus, $\|\mathbf{x}(t)\| = \sqrt{\mathsf{N}s(\mathbf{x}(t))} \geq \sqrt{\mathsf{N}}$ since $s(\mathbf{x}(t)) \geq 1$. The rest of the argument for this case follows as before using $V_2(\mathbf{x}) = s(\mathbf{x}) - 1$. Thus, $\mathbf{x}(t)$ reaches $\mathcal{M}$ in finite time. In addition, $\mathbf{x}(t)$ remain inside $\mathcal{M}$ for $t \geq T_0 + T_\varepsilon$ since $V_1(\mathbf{x}(t))$ (resp. $V_2(\mathbf{x}(t))$) cannot increase at the inner (resp. outer) boundary of $\mathcal{M}$.

□

Now, we recall the definition of the projected gradient with respect to a manifold and then apply the definition to the problem of interest in (5.2).

Definition 9. *[Hauswirth et al., 2016] Let $F : \mathbb{R}^n \to \mathbb{R}$ be differentiable and the constrained manifold $\mathcal{M} = \{\mathbf{x} \in \mathbb{R}^n : \gamma_i(\mathbf{x}) \leq 0, i \in \{1, \dots, m\}\}$ for m differentiable constraint functions $\gamma_i : \mathbb{R}^n \to \mathbb{R}$. Then, the projected gradient of $F(\mathbf{x})$ with respect to $\mathcal{M}$ is given by:*

$$\begin{aligned} \text{proj}_{\mathcal{M}}(\mathbf{x}, -\nabla F(\mathbf{x})) = \underset{\mathbf{w} \in \mathbb{R}^n}{\arg\min} \, \|\mathbf{w} - (-\nabla F(\mathbf{x}))\|^2 \\ \text{such that} \quad \mathbf{w}^\top \nabla \gamma_i(\mathbf{x}) \leq 0, \forall i \in \mathcal{J}(\mathbf{x}) \end{aligned} \tag{5.10}$$

where $\mathcal{J}(\mathbf{x}) = \{i \in \{1, \dots, m\} : \gamma_i(\mathbf{x}) = 0\}$.

Lemma 8. *Let $\mathcal{M}$ with $\gamma_1(\mathbf{x}) = \|\mathbf{x}\|^2/\mathsf{N} - 1$, $\gamma_2(\mathbf{x}) = (1-\varepsilon) - \|\mathbf{x}\|^2/\mathsf{N}$ in Definition 9 with $m = 2$. Then, for any differentiable $F(\bullet)$ and any $\mathbf{x} \in \mathcal{M}$,*

$$\text{proj}_{\mathcal{M}}(\mathbf{x}, -\nabla F(\mathbf{x})) = \begin{cases} -\nabla F(\mathbf{x}) + \left(\frac{\mathbf{x}^\top \nabla F(\mathbf{x})}{\mathsf{N}^2}\right)\mathbf{x} & \gamma_1(\mathbf{x}) = 0, \mathbf{x}^\top \nabla F(\mathbf{x}) \leq 0 \\ -\nabla F(\mathbf{x}) - \left(\frac{\mathbf{x}^\top \nabla F(\mathbf{x})}{(\mathsf{N}(1-\varepsilon))^2}\right)\mathbf{x} & \gamma_2(\mathbf{x}) = 0, \mathbf{x}^\top \nabla F(\mathbf{x}) \geq 0 \\ -\nabla F(\mathbf{x}) & \text{otherwise} \end{cases} \tag{5.11}$$

Proof. First, $\nabla \gamma_1(\mathbf{x}) = 2\mathbf{x}/\mathsf{N}, \nabla \gamma_2(\mathbf{x}) = -2\mathbf{x}/\mathsf{N}$. Consider the following cases:

- (a) $\gamma_1(\mathbf{x}) < 0, \gamma_2(\mathbf{x}) < 0$: since (5.10) is unconstrained, $\mathbf{w} = -\nabla F(\mathbf{x})$.
- (b) For $\gamma_1(\mathbf{x}) = 0, \mathbf{x}^\top \nabla F(\mathbf{x}) \geq 0$ or $\gamma_2(\mathbf{x}) = 0, \mathbf{x}^\top \nabla F(\mathbf{x}) \leq 0$: $\mathbf{w} = -\nabla F(\mathbf{x})$ satisfies the constraint $\mathbf{w}^\top \nabla \gamma_1(\mathbf{x}) \leq 0$ and $\mathbf{w}^\top \nabla \gamma_2(\mathbf{x}) \leq 0$ in (5.10) respectively.
- (c) $\gamma_1(\mathbf{x}) = 0, \mathbf{x}^\top \nabla F(\mathbf{x}) \leq 0$: let $h(\mathbf{w}) = \|\mathbf{w} + \nabla F(\mathbf{x})\|^2$ and define the quantity $\mathbf{w}^* = -\nabla F(\mathbf{x}) + \left(\frac{\mathbf{x}^\top \nabla F(\mathbf{x})}{\mathsf{N}^2}\right)\mathbf{x}$. Note that $(\mathbf{w}^*)^\top \nabla \gamma_1(\mathbf{x}) = 0$ making $\mathbf{w}^*$ is feasible for (5.10) and $\nabla_{\mathbf{w}} h(\mathbf{w}) = 2(\nabla F(\mathbf{x}) + \mathbf{w}) = \lambda \mathbf{x}/\mathsf{N} = (\lambda/2)\nabla_{\mathbf{w}}(\mathbf{w}^\top \nabla \gamma_1(\mathbf{x}))$ is satisfied with $\mathbf{w} = \mathbf{w}^*$ and Lagrange multiplier $\lambda = 2\mathbf{x}^\top \nabla F(\mathbf{x})/\mathsf{N}$ where $\nabla_{\mathbf{w}}$ denotes gradient with respect to $\mathbf{w}$. Hence, $\mathbf{w} = \mathbf{w}^*$ satisfies the optimality conditions for (5.10) and is the unique minimizer [Bertsekas, 1999, Proposition 3.1.1].
- (d) $\gamma_2(\mathbf{x}) = 0, \mathbf{x}^\top \nabla F(\mathbf{x}) \geq 0$: the same reasoning of (c) applies.

□

In the following, we show that the ideal system (5.6) flows in the direction of the previously obtained projected gradient.

Lemma 9. *Let (5.6) with $\mathbf{x}(T_0) \in \mathcal{M}$. Then, under Definition 9,*

$$\dot{\mathbf{x}}(t) = \alpha(\mathbf{x}(t)) \, \text{proj}_{\mathcal{M}}(\mathbf{x}(t), -\nabla f(\mathbf{Q}(\mathbf{x}(t))))$$

for some positive scalar function $\alpha : \mathbb{R}^n \to \mathbb{R}_{>0}$.

Proof. In this proof, we omit time dependence for conciseness. We compare the right hand side of (5.6) with (5.11) in Lemma 8 under $F(\mathbf{x}) = f(\mathbf{Q}(\mathbf{x}))$:

- (a) $\gamma_1(\mathbf{x}) < 0, \gamma_2(\mathbf{x}) < 0$: $\dot{\mathbf{x}} = -\alpha(\mathbf{x})\nabla F(\mathbf{x}) = \alpha(\mathbf{x})\ \mathrm{proj}_{\mathcal{M}}(\mathbf{x}, -\nabla F(\mathbf{x}))$ due to (5.11) with $\alpha(\mathbf{x}) = \kappa_{\mathcal{M}}\mathsf{N}$.
- (b) $\gamma_1(\mathbf{x}) = 0$: we invoke the Filippov interpretation of solutions to write $\dot{\mathbf{x}}$. Call $\mathbf{w}_1 = \kappa_{\mathcal{M}}\mathbf{x}\mathrm{sign}((1-\frac{\varepsilon}{2}) - s(\mathbf{x}))$ and $\mathbf{w}_2 = -\kappa_{\mathcal{M}}\mathsf{N}\nabla f(\mathbf{Q}(\mathbf{x}))$. Filippov solutions in this manifold have $\dot{\mathbf{x}}$ in the convex hull of $\{\mathbf{w}_1, \mathbf{w}_2\}$. Concretely, if $\gamma_1(\mathbf{x}) = 0$, then $\mathbf{x}^\top\nabla F(\mathbf{x}) \geq 0$, then $\mathbf{w}_1, \mathbf{w}_2$ point inside $\mathcal{M}$, and $s(\mathbf{x}(t)) = 0$ is satisfied for an isolated instant t, hence ignored by the trajectory [Cortes, 2008]. In the case of $\gamma_1(\mathbf{x}) = 0$, we have $\mathbf{x}^\top\nabla F(\mathbf{x}) \leq 0$, $\mathbf{w}_1, \mathbf{w}_2$ point in different directions with respect to the inner boundary of $\mathcal{M}$, inducing sliding motion along $\gamma_1(\mathbf{x}) = 0$. Therefore, $\dot{\mathbf{x}}$ is the convex combination of $\mathbf{w}_1, \mathbf{w}_2$, lying in the tangent plane of $\gamma_1(\mathbf{x}) = 0$ [Cortes, 2008]. Note that

$$\mathbf{w} = -\nabla F(\mathbf{x}) + \left(\frac{\mathbf{x}^\top\nabla F(\mathbf{x})}{\mathsf{N}^2}\right)\mathbf{x} = \left(\frac{\mathbf{x}^\top\nabla F(\mathbf{x})}{\kappa_{\mathcal{M}}\mathsf{N}^2\mathrm{sign}((1-\frac{\varepsilon}{2}) - s(\mathbf{x}))}\right)\mathbf{w}_1 + \left(\frac{1}{\kappa_{\mathcal{M}}\mathsf{N}}\right)\mathbf{w}_2$$

  with $\mathbf{w}^\top\mathbf{x} = 0$, lying in such tangent plane. Hence, $\dot{\mathbf{x}}$ is proportional to $\mathbf{w}$ and $\dot{\mathbf{x}} = \alpha(\mathbf{x})\mathbf{w} = \alpha(\mathbf{x})\ \mathrm{proj}_{\mathcal{M}}(\mathbf{x}, -\nabla F(\mathbf{x}))$ for some $\alpha(\mathbf{x}) > 0$.
- (c) $\gamma_2(\mathbf{x}) = 0$: the proof follows from (b). □

Lemma 10. *Consider* (5.6) *under Assumption 4 and* $f(\bullet)$ *in Table 5.1 labeled by* $\mu \in \{0, 1, 2\}$. *For any* $\mathbf{x}(T_0) \notin \mathcal{M}$ *it follows that* $\|\dot{\mathbf{x}}(t)\| \leq h(\mathsf{N}), \forall t \geq T_0$ *with* $h(\bullet)$ *as in Theorem 11.*

Proof. First, we show that $|x_i(t)| \leq \bar{b}, \forall t \geq T_0$ by contradiction. Assume there exists $T_2 \geq T_0$ such that $|x_i(T_2)| > \bar{b}$. By continuity of the solution, there must have existed $T_1 = \sup\{t \in [T_0, T_2] : |x_i(t)| = \bar{b}\}$ such that $a_i(t) = |x_i(t)|$ is increasing for $t \in [T_1, T_2]$. However, in this interval, $\mathbf{x}(t) \notin \mathcal{M}$ since $s(\mathbf{x}(t)) = \|\mathbf{x}(t)\|^2/\mathsf{N} \geq \min_{i\in\mathcal{I}}|x_i(t)|^2/\mathsf{N} \geq \bar{b}^2 > 1$. Therefore, due to (5.6), $\dot{a}_i(t) = -\kappa_{\mathcal{M}}a_i(t)$ which means that $a_i(t)$ is decreasing in $t \in [T_1, T_2]$ leading to a contradiction.

Now, we show $\mathrm{tr}(\mathbf{Q}(\mathbf{x}(t))^{-1}) \leq \mathsf{N}(\underline{\sigma}\min\{\underline{b}^2, 1-\varepsilon\})^{-1}, \forall t \geq T_0$. From Lemma 7's proof, $s(\mathbf{x}(t)) = \|x_i(t)\|^2/\mathsf{N} \geq \underline{b}^2, \forall t \in [T_0, T_0 + T_\varepsilon]$ and $s(\mathbf{x}(t)) \geq 1-\varepsilon, \forall t \geq T_0 + T_\varepsilon$. Henceforth, $\mathbf{Q}(\mathbf{x}(t)) \succeq (1/\mathsf{N})\sum_{i=1}^{\mathsf{N}} x_i(t)^2\underline{\sigma}\mathbf{I} \succeq \underline{\sigma}\min\{\underline{b}^2, 1-\varepsilon\}\mathbf{I}$ for $t \geq T_0$. Then, it follows that we have $\mathbf{Q}(\mathbf{x}(t))^{-1} \preceq (\underline{\sigma}\min\{\underline{b}^2, 1-\varepsilon\})^{-1}\mathbf{I}$ from which $\mathrm{tr}(\mathbf{Q}(\mathbf{x}(t))^{-1}) \leq \mathsf{N}(\underline{\sigma}\min\{\underline{b}^2, 1-\varepsilon\})^{-1}$ is obtained. From the previous bounds, and the gradient in Table 5.1:

$$\|\nabla f(\mathbf{Q}(\mathbf{x}(t)))\| \leq \max_{i\in\mathcal{I}} 2|x_i|\mathrm{tr}(\mathbf{P}_i^{-1})\mathrm{tr}(\mathbf{Q}(\mathbf{x}(t))^{-1})^\mu \leq 2\bar{b}\bar{\sigma}\mathsf{N}^\mu(\underline{\sigma}\min\{\underline{b}^2, 1-\varepsilon\})^{-\mu},$$

and

$$\|\dot{\mathbf{x}}(t)\| \leq \kappa_{\mathcal{M}}\max\{\|\mathbf{x}(t)\|, \mathsf{N}\|\nabla f(\mathbf{Q}(\mathbf{x}(t)))\|\} \leq \kappa_{\mathcal{M}}\max\{\sqrt{\mathsf{N}}\bar{b}, 2\bar{b}\bar{\sigma}\mathsf{N}^{\mu+1}(\underline{\sigma}\min\{\underline{b}^2, 1-\varepsilon\})^{-\mu}\}.$$

□

Recalling from Lemma 7 that (5.6) flows in the direction of the projected gradient, the following result characterizes convergence of such flow towards the optimum of (5.2) (the result follows from [Hauswirth et al., 2016, Theorem 3].).

Proposition 14. *Let $F(\bullet), \gamma_i(\bullet), \mathcal{M}$ defined as in Definition 9 and $\alpha : \mathbb{R}^n \to \mathbb{R}_{>0}$ be a positive scalar function such that $\dot{\mathbf{x}}(t) = \alpha(\mathbf{x}(t))\ \mathrm{proj}_{\mathcal{M}}(\mathbf{x}(t), -\nabla f(\mathbf{x}(t)))$ has unique forward solution for $t \geq T_0$ and any $\mathbf{x}(T_0) \in \mathcal{M}$. Then, $\mathbf{x}(t)$ converges asymptotically towards $\arg\min_{\mathbf{x}\in\mathcal{M}} F(\mathbf{x})$.*

Proof of Theorem 11: We start by analyzing (5.4) for $t \in [0, T_c]$, in which $x_i(t)^2\mathbf{P}_i^{-1}$ and $x_i(t)^2$ remain constant because $u_i(t) = 0$. The result from Lemma 6 is valid obtaining $\hat{s}_i(t) \equiv s(\mathbf{x}(t)), \hat{\mathbf{Q}}_i(t) \equiv \mathbf{Q}(\mathbf{x}(t))$ at $t = T_c$. Assume that the condition $\hat{s}_i(t) \equiv s(\mathbf{x}(t))$ is maintained for some open interval $\mathcal{T} \subset [T_c, \infty)$. Then, $\mathbf{x}(t)$ is dictated by (5.6) for any $t \in \mathcal{T}$ and thus $z_i(t) = x_i(t)^2$ comply $|\dot{z}_i(t)| \leq 2|x_i(t)|\|\dot{\mathbf{x}}(t)\| \leq 2\bar{b}h(\mathsf{N})$ due to Lemma 10. Hence, the conditions of Lemma 6 are fulfilled with $L = 2\bar{b}h(\mathsf{N})$ due to the choice of κ_s in Theorem 11 for such interval $\mathcal{T}$.

Now, we show that $\hat{s}_i(t) \equiv s(\mathbf{x}(t))$ is maintained $\forall t \in [T_c, \infty)$. We verify this by contradiction. Assume that there exists $T_2 = \inf\{t \geq T_c : \|\dot{\mathbf{x}}(t)\| > 2\bar{b}h(\mathsf{N})\}$. Then, there must have existed some $T_1 = \sup\{t < T_2 : \hat{s}_i(t) \neq s(\mathbf{x}(t))$ or $\hat{\mathbf{Q}}_i(t) \neq \mathbf{Q}(\mathbf{x}(t))\}$ since the system could not have been the ideal one in (5.6). Hence, the system is not synchronized for some time $t \in [T_1, T_2]$ for which $\|\mathbf{x}(t)\| \leq 2\bar{b}h(\mathsf{N})$, which is impossible since this means that the inputs for the consensus protocols have bounded derivative, and Lemma 6 ensure synchronization. Hence, $\hat{s}_i(t) \equiv s(\mathbf{x}(t)), \forall t \geq T_c$. The same reasoning applies to the synchronization condition $\hat{\mathbf{Q}}_i(t) \equiv \mathbf{Q}(\mathbf{x}(t))$.

Now, use Lemma 7 with $T_0 = T_c$ to obtain $\mathbf{x}(t) \in \mathcal{M}, \forall t \geq T_c + T_\varepsilon$. We can verify $\mathcal{E}(\mathbf{Q}(\mathbf{x}(t))^{-1}) \supset \check{\mathcal{E}}, \forall t \geq T_c + T_\varepsilon$ by taking any $\mathbf{y} \in \check{\mathcal{E}}$ which satisfy

$$\mathbf{y}^\top \mathbf{Q}(\mathbf{x}(t))\mathbf{y} = \sum_{i=1}^{\mathsf{N}} (x_i(t)^2/\mathsf{N})\mathbf{y}^\top \mathbf{P}_i^{-1}\mathbf{y} \leq \sum_{i=1}^{\mathsf{N}} (x_i(t)^2/\mathsf{N}) = s(\mathbf{x}(t)) \leq 1.$$

Hence, $\mathbf{y} \in \mathcal{E}(\mathbf{Q}(\mathbf{x}(t))^{-1})$. Use Lemma 9 with $T_0 = T_c + T_\varepsilon$ to write (5.6) as $\dot{\mathbf{x}}(t) = \alpha(\mathbf{x}(t))\ \mathrm{proj}_{\mathcal{M}}(\mathbf{x}(t), -\nabla f(\mathbf{Q}(\mathbf{x}(t))))$, implying convergence towards the optimal weights due to Proposition 14. □

5.4 Distributed discrete-time outer ellipse computation

To compute the global optimum of problem (5.3), we propose that, at each instant k, each agent i solves the semi-definite program:

$$\mathbf{Q}_i[k] = \underset{\mathbf{Q}\in\mathcal{C}_i[k]}{\arg\min}\ f(\mathbf{Q})$$

$$\mathcal{C}_i[k] = \left\{\mathbf{Q}\in\mathbb{S}_+^n : \exists \lambda_j^i, \lambda_{\mathbf{P}}^i \in [0,1], j\in\mathcal{N}_i, \lambda_{\mathbf{P}}^i + \sum_{j\in\mathcal{N}_i}^{\mathsf{N}} \lambda_j^i \leq 1, \mathbf{0} \preceq \mathbf{Q} \preceq \lambda_{\mathbf{P}}^i \mathbf{P}_i[k]^{-1} + \frac{1}{\bar{\theta}} \sum_{j\in\mathcal{N}_i} \lambda_j^i \mathbf{Q}_j[k-1]\right\} \tag{5.12}$$

for $k \geq 1$, with initial conditions $\mathbf{Q}_i[0] = \mathbf{P}_i[0]^{-1}$ and a parameter $\bar{\theta}$ to be specified subsequently. Note that each node only requires the solution of (5.12) at each neighbor in the previous instant. Therefore, (5.12) defines a distributed algorithm, described in Algorithm 9. Its properties are summarized in Theorem 12.

Algorithm 9 Distributed estimation of (5.3) at node i

1: Initialization: $\lambda_j^i, \lambda_{\mathbf{P}}^i = 0 \quad \forall j \in \mathcal{N}_i$, $\bar{\theta}_i = \bar{\theta} \geq 1$ and $\mathbf{Q}_i[0] = \mathbf{P}_i[0]^{-1}$
2: **for** $k = 1, 2, \dots$ **do**
3: Information exchange:

Send $\mathbf{Q}_i[k-1]$ to neighbors $j \in \mathcal{N}_i$.

Receive $\mathbf{Q}_j[k-1]$ from neighbors $j \in \mathcal{N}_i$.

4: Solve local reformulated outer Löwner-John program:

$\mathbf{Q}_i[k] = \arg\min_{\mathbf{Q} \in \mathcal{C}_i[k]} f(\mathbf{Q})$, with $\mathcal{C}_i[k]$ given in (5.12).

5: **end for**

Theorem 12. *Let Assumptions 2 and 3 hold and $\mathcal{G}$ be connected. Then, for a network that executes Algorithm 9, the following statements are true:*

1. *(Robustness) $\mathcal{C}_i[k] \subseteq \mathcal{C}^*[k]$ for all $k \in \{1, 2 \dots\}, i \in \mathcal{I}$.*
2. *(Boundedness) $\mathbf{Q}_i[k] \preceq (1/\underline{p})\mathbf{I}$ for all $k \in \{1, 2 \dots\}, i \in \mathcal{I}$.*
3. *(Convergence) Given any choice of norm $\|\bullet\|$ in $\mathbb{S}_+^n$ and any $\delta > 0$, there exists $K \in \{1, 2, \dots\}, 0 \leq \underline{\theta} < 1 < \bar{\theta}$ such that if $\underline{\theta}\mathbf{P}_i[k-1] \preceq \mathbf{P}_i[k] \preceq \bar{\theta}\mathbf{P}_i[k-1], \forall k = 1, 2, \dots$ then,*

$$\|\mathbf{Q}_i[k] - \mathbf{Q}^*[k]\| \leq \delta \text{ for any } k \geq K, i \in \mathcal{I}.$$

In addition, if $\underline{\theta} = \bar{\theta} = 1$, then $\mathbf{Q}_i[k] = \mathbf{Q}^*[k]$.

First, the intuition behind *robustness* is that, given arbitrary input matrices $\{\mathbf{P}_i[k]\}_{i=1}^{\mathsf{N}}$ under Assumption 2, the feasible set of the local optimization problem at each node is always contained inside the feasible set of the global optimization problem; this is important because, if *robustness* did not hold, then the optimum of the local optimization problem at node i might be such that $\mathbf{Q}_i[k] \notin \mathcal{C}^*[k]$ and, therefore, it would not be possible to converge arbitrarily close to the global optimum. Second, *boundedness* implies that, irrespective of the changes in the input ellipsoids, the estimate of the global optimum at each node never escapes to infinity. Third, *convergence* means that our algorithm converges in finite time K to a region around the optimum of the global problem, where the size of the region depends on how fast the input ellipsoids vary with time: in particular, if the input ellipsoids are constant, then $\delta = 0$ and the global optimum is perfectly recovered; besides, the estimates remain inside that region for $k > K$.

Remark 4. *In many applications, the inputs $\{\mathbf{P}_i[k]\}_{i=1}^{\mathsf{N}}$ come from the discretization of the continuous-time dynamics of $\{\mathbf{P}_i(t)\}_{i=1}^{\mathsf{N}}$. In this case, the designer can choose the sampling*

step sufficiently small, such that Assumption 2 holds for $\underline{\theta} \simeq \bar{\theta} \simeq 1$ *and a desired accuracy* δ *is achieved.*

Remark 5. *The tracking error depends on the rate of change of the input ellipsoids. This suggests an adaptive scheme over parameter* $\bar{\theta}$*. One can follow similar procedures to those found in [Lippi et al., 2022, Deplano et al., 2022] but adapted to positive definite matrices. The rate of change of local input matrices at instant* k *is approximately given by* $\mathbf{P}_i[k-1]\mathbf{P}_i[k]^{-1}$*. Then, in practice, one can choose* $\bar{\theta}_i[k] = \kappa \mathbf{P}_i[k-1]\mathbf{P}_i[k]^{-1}$ *with* $\kappa > 0$ *in place of* $\bar{\theta}$ *in* (5.12)*.*

After discussing the proposed algorithm and its main properties, the next section is devoted to prove Theorem 12.

5.5 Convergence analysis

To prove Theorem 12, we first present some auxiliary results (Sec. 5.5.1) helpful for the main proofs. After that, we prove the theorem for the static case, i.e., $\mathbf{P}_i[k] = \mathbf{P}_i \ \forall i, k$ (Sec. 5.5.2). This intermediate step provides the tools and insights to prove Theorem 12 in Section 5.5.3.

5.5.1 Auxiliary results

First, we characterize some properties of general closed convex sets that are useful for the subsequent study of the properties of the feasible sets $\mathcal{C}_i[k]$ and $\mathcal{C}^*[k]$.

Lemma 11. *Let* $\mathcal{C}_1, \mathcal{C}_2$ *two closed and convex sets such that* $\mathcal{C}_1 \subseteq \mathcal{C}_2 \subseteq \mathbb{S}^n_+$*. Moreover, denote by*

$$\mathbf{Q}_1 = \underset{\mathbf{Q} \in \mathcal{C}_1}{\arg\min}\, f(\mathbf{Q}) \qquad \textit{and} \qquad \mathbf{Q}_2 = \underset{\mathbf{Q} \in \mathcal{C}_2}{\arg\min}\, f(\mathbf{Q}).$$

Then, the following holds:

1. *If* $\mathbf{Q}_2 \in \mathcal{C}_1$ *then* $\mathbf{Q}_1 = \mathbf{Q}_2$*.*
2. *If* $\mathbf{Q}_2 \notin \mathcal{C}_1$ *then* $\mathbf{Q}_1 \in \mathsf{rebdr}(\mathcal{C}_1)$*.*

Proof. For the first item, since $f(\bullet)$ is strictly convex and $\mathcal{C}_1, \mathcal{C}_2$ are convex, then $\mathbf{Q}_1, \mathbf{Q}_2$ must be unique. Hence, if $\mathbf{Q}_2 \in \mathcal{C}_1$, then $\mathbf{Q}_1 = \mathbf{Q}_2$ must follow. For the second item, $\mathcal{C}_1$ is closed, so the relative boundary $\mathsf{rebdr}(\mathcal{C}_1)$ exists and contains all its relative boundary points. The next steps of the proof follow by contradiction. Assume that $\mathbf{Q}_1 \in \mathsf{relint}(\mathcal{C}_1)$, where $\mathsf{relint}(\bullet)$ represents the relative interior operator. Then, there exist a ball $\mathcal{B} \in \mathsf{relint}(\mathcal{C}_1)$ centered at $\mathbf{Q}_1$ such that $f(\mathbf{Q}_1) \leq f(\mathbf{Q}), \forall \mathbf{Q} \in \mathcal{B}$. Thus, $\mathbf{Q}_1$ is a local optimum of $f(\bullet)$, and, as a result of convexity of $f(\bullet)$ and $\mathcal{C}_1, \mathcal{C}_2$, it is the global optimum of $f(\bullet)$ in any case. By assumption of the lemma, $\mathbf{Q}_1 \in \mathcal{C}_1 \subseteq \mathcal{C}_2$ and thus $\mathbf{Q}_1 = \mathbf{Q}_2$. However, by assumption of the second item, $\mathbf{Q}_2 \notin \mathcal{C}_1$, which leads to the contradiction $\mathbf{Q}_1 \notin \mathcal{C}_1$. □

The next lemma demonstrates properties of the feasible set $\mathcal{C}_i[k]$ and $\mathcal{C}^*[k]$ that are important to study the relationships between the centralized and distributed optimization problems (5.3) and (5.12).

Lemma 12. *Let Assumptions 2 and 3 hold. Then, the following statements are true:*

1. *The sets $\mathcal{C}_i[k], \mathcal{C}^*[k]$ are closed and convex for all $k \in \{1, 2, \dots\}$ and $i \in \mathcal{I}$.*
2. *Denote $\mathbf{Q}^*[k] = \min_{\mathbf{Q} \in \mathcal{C}^*[k]} f(\mathbf{Q})$ for arbitrary fixed $k \in \{0, 1, \dots, \}$. Then, we have $\mathbf{Q}^*[k] \in \mathsf{co}\{\mathbf{P}_i[k]^{-1}\}_{i=1}^{\mathsf{N}}$, where $\mathsf{co}(\bullet)$ denotes the convex hull of a set.*
3. *$\mathcal{C}_i[k] \subseteq \mathcal{C}^*[k]$ for all $k \in \{1, 2, \dots\}, i \in \mathcal{I}$.*

Proof. For item 1), the result is straightforward since the sets come from the standard semi-definite programs defined in (5.3)-(5.12). For item 2), let $\mathcal{C}_1 = \mathcal{C}^*[k], \mathcal{C}_2 = \mathbb{S}^n_+$. Hence, item 3) of Assumption 3 implies that $\mathbf{Q}_2$ from Lemma 11 does not exist and as a result $\mathbf{Q}_2 \notin \mathcal{C}_1$ and $\mathbf{Q}^*[k] = \mathbf{Q}_1 \in \mathsf{rebdr}(\mathcal{C}_1)$. Note that points in the relative boundary of $\mathcal{C}_1$ are convex combinations of $\{\mathbf{P}_i[k]^{-1}\}_{i=1}^{\mathsf{N}}$ from which the result follows.

For item 3) we proceed by induction. As induction base, set an arbitrary agent $i \in \mathcal{I}$ and define $\mathcal{C}_i[0]$ only for $k = 0$ as $\mathcal{C}_i[0] := \{\mathbf{P}_i[0]^{-1}\}$, so that the initial condition comply $\mathbf{Q}_i[0] = \mathbf{P}_i^{-1}[0] \in \mathcal{C}_i[0]$. Now, referring to the centralized optimization problem (5.3), let $\lambda_j = 1$ if $j = i$ and $\lambda_j = 0$ otherwise. Hence, $\mathbf{Q}_i[0] = \sum_{j=1}^{\mathsf{N}} \lambda_j \mathbf{P}_j[0]^{-1} \in \mathcal{C}^*[0]$. Henceforth, $\mathcal{C}_i[0] \subseteq \mathcal{C}^*[0]$.

Now, assume that $\mathcal{C}_i[k-1] \subseteq \mathcal{C}^*[k-1]$ for some $k \in \{1, 2, \dots\}$. Then, it follows $\mathbf{Q}_i[k-1] \in \mathcal{C}^*[k-1]$, meaning that there exists $\beta_1^j, \dots, \beta_{\mathsf{N}}^j \in [0, 1]$ with $\sum_{\ell=1}^{\mathsf{N}} \beta_\ell^j \leq 1$ and

$$\mathbf{0} \preceq \mathbf{Q}_j[k-1] \preceq \sum_{\ell=1}^{\mathsf{N}} \beta_\ell^j \mathbf{P}_\ell[k-1]^{-1} \preceq \bar{\theta} \sum_{\ell=1}^{\mathsf{N}} \beta_\ell^j \mathbf{P}_\ell[k]^{-1}$$

For an arbitrary $\mathbf{Q} \in \mathcal{C}_i[k], k = 1, 2, \dots$ it follows that

$$\mathbf{Q} \preceq \lambda_{\mathbf{P}}^i \mathbf{P}_i[k]^{-1} + \frac{1}{\bar{\theta}} \sum_{j \in \mathcal{N}_i} \lambda_j^i \mathbf{Q}_j[k-1] \preceq \lambda_{\mathbf{P}}^i \mathbf{P}_i[k]^{-1} + \sum_{j \in \mathcal{N}_i} \lambda_j^i \sum_{\ell=1}^{\mathsf{N}} \beta_\ell^j \mathbf{P}_\ell[k]^{-1}$$

Now, consider

$$\lambda_\ell = \begin{cases} \lambda_{\mathbf{P}}^i + \sum_{j \in \mathcal{N}_i} \lambda_j^i \beta_l^j & \text{if } \ell = i \\ \sum_{j \in \mathcal{N}_i} \lambda_j^i \beta_\ell^j & \text{otherwise} \end{cases}$$

Then, it follows that

$$\sum_{\ell=1}^{\mathsf{N}} \lambda_\ell = \lambda_{\mathbf{P}}^i + \sum_{j \in \mathcal{N}_i} \lambda_j^i \beta_i^j + \sum_{l \neq i} \sum_{j \in \mathcal{N}_i} \lambda_j^i \beta_\ell^j = \lambda_{\mathbf{P}}^i + \sum_{j \in \mathcal{N}_i} \lambda_j^i \sum_{\ell=1}^{\mathsf{N}} \beta_\ell^j \leq \lambda_{\mathbf{P}}^i + \sum_{j \in \mathcal{N}_i} \lambda_j^i \leq 1$$

and

$$\sum_{\ell=1}^{\mathsf{N}} \lambda_\ell \mathbf{P}_\ell[k]^{-1} = \left(\lambda_{\mathbf{P}}^i + \sum_{j \in \mathcal{N}_i} \lambda_j^i \beta_i^j \right) \mathbf{P}_i[k]^{-1} + \sum_{l \neq i} \left(\sum_{j \in \mathcal{N}_i} \lambda_j^i \beta_\ell^j \right) \mathbf{P}_\ell[k]^{-1} =$$

$$\lambda_{\mathbf{P}}^i \mathbf{P}_i[k] + \sum_{\ell=1}^{\mathsf{N}} \left(\sum_{j \in \mathcal{N}_i} \lambda_j^i \beta_\ell^j \right) \mathbf{P}_\ell[k]^{-1} \succeq \mathbf{Q}$$

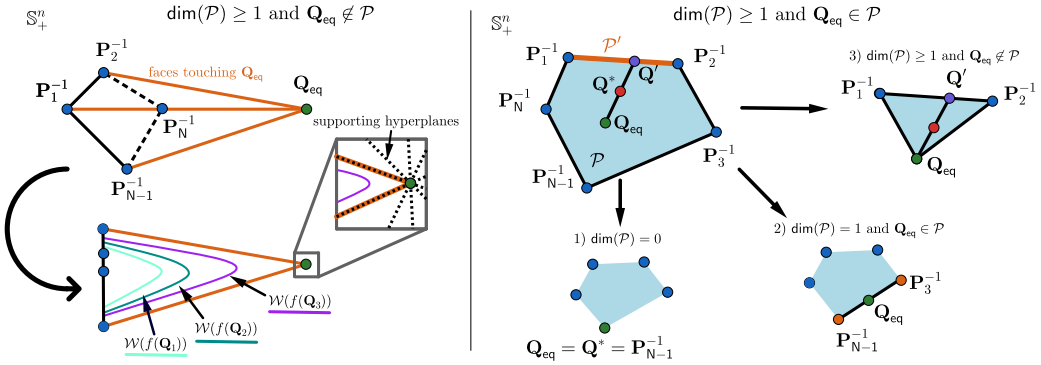


Figure 5.1: Illustrative visualization of the cases 3) and 4) of the proof of Lemma 14. The figure leverages the representation of ellipsoids in $\mathbb{S}^n_+$ projected to $\mathbb{R}^3$ and $\mathbb{R}^2$.

Therefore, the feasible solution set for agent i of problem (5.12) is a particular case of the feasible solution set of problem (5.3), $\mathbf{Q} \in \mathcal{C}^*[k]$ and, as a result, $\mathcal{C}_i[k] \subseteq \mathcal{C}^*[k]$. □

5.5.2 Constant inputs

In this section we focus on the static case, i.e., the input ellipsoids do not change with time. Henceforth, it is assumed that Assumption 2 holds with $\underline{\theta} = \overline{\theta} = 1$, meaning that the inputs $\mathbf{P}_i[k]$ are constant.

Lemma 13. *Let Assumptions 2 and 3 hold and $\underline{\theta} = \overline{\theta} = 1$. Then,*

$$f(\mathbf{Q}_i[k]) \leq f(\mathbf{Q}_i[k-1]). \tag{5.13}$$

for all $i \in \mathcal{I}, k = 1, 2, \ldots$. Moreover, there exists a sequence $f_1, \ldots, f_N \in \mathbb{R}$ such that it holds that $\lim_{k\to\infty} f(\mathbf{Q}_i[k]) = f_i$.

Proof. Let $\lambda_j^i = 1$ if $j = i$ and $\lambda_j^i = 0$ otherwise, and $\lambda_{\mathbf{P}}^i = 0$. Therefore,

$$\lambda_{\mathbf{P}}^i \mathbf{P}_i[k]^{-1} + \sum_{j \in \mathcal{N}_i} \lambda_j^i \mathbf{Q}_j[k-1] = \mathbf{Q}_i[k-1]$$

and $\lambda_{\mathbf{P}}^i + \sum_{j\in\mathcal{N}_i} \lambda_j^i = 1$. Hence, $\mathbf{Q}_i[k-1] \in \mathcal{C}_i[k]$. Therefore, (5.13) follows by noting that both $\mathbf{Q}_i[k], \mathbf{Q}_i[k-1] \in \mathcal{C}_i[k]$ but that $\mathbf{Q}_i[k]$ is the minimizer of $f(\bullet)$ over $\mathcal{C}_i[k]$.

For the last part of the lemma, note that $\mathcal{C}^*$ is closed and convex due to Lemma 12-1) such that item 4) of Assumption 3 implies f is lower bounded over $\mathcal{C}^*$. Moreover, since $\mathcal{C}_i[k] \subseteq \mathcal{C}^*$ then f attains the same lower bound over $\mathcal{C}_i[k]$. Furthermore, combine (5.13) with such lower bound to conclude that $f_i := \lim_{k\to\infty} f(\mathbf{Q}_i[k])$ must exist due to monotonicity in (5.13). □

Lemma 14. *Let Assumptions 2 and 3 hold, and $\underline{\theta} = \overline{\theta} = 1$. Moreover, let $\mathcal{G}$ be connected. Then, $\lim_{k\to\infty} \mathbf{Q}_i[k] = \mathbf{Q}^*$ for all $i \in \mathcal{I}$ where $\mathbf{Q}^*$ is the (constant) optimum of (5.3).*

Proof. First, we study the equilibrium of (5.12) as follows. Let $\ell = \arg\min_{i\in\mathcal{I}} f_i^*$ with f_i^* taken from Lemma 13. Moreover, let $\mathcal{N}_\ell^r = \{i \in \mathcal{I} : i \in \mathcal{N}_j, j \in \mathcal{N}_\ell^{r-1}\}$ and $\mathcal{N}_\ell^0 = \mathcal{N}_\ell$. This is, $\mathcal{N}_\ell^{r-1}$ contains the set of neighbors of neighbors of $\ell \in \mathcal{I}$ after r hops. We now show that $f_\ell = f_i$ for all $i \in \mathcal{N}_\ell^r$ and all $r \in \{0, 1, \dots\}$ by induction, for some k in which equilibrium is attained. For the induction base, note that there exists $\mathbf{Q}'_\ell \in \mathbb{S}_+^n$ such that $f(\mathbf{Q}'_\ell) = f_\ell \leq f_i \leq f(\mathbf{Q})$ for all $\mathbf{Q} \in \mathcal{C}_i[k]$ with $i \in \mathcal{N}_\ell$. Thus, (5.12) implies $f_i = f(\mathbf{Q}'_\ell)$ since $\mathbf{Q}'_\ell \in \mathcal{C}_i[k]$. The induction step follows in the same way. Finally, since $\mathcal{G}$ is connected, then $\mathcal{N}_\ell^d = \mathcal{I}$ where d is the diameter of $\mathcal{G}$. Therefore, $f_i = f_\ell$ for all $i \in \mathcal{I}$. Finally, note that (5.12) being a strictly convex program implies that $f(\mathbf{Q}) = f_\ell$ is attained for a single $\mathbf{Q} \in \mathbb{S}_+^n$ which we denote as $\mathbf{Q}_{\mathsf{eq}}$.

Moreover, item 3) of Lemma 12 implies that in equilibrium, $\mathcal{C}_i \subseteq \mathcal{C}^*$, which are constant $\mathcal{C}_i = \mathcal{C}_i[k], \mathcal{C}^* = \mathcal{C}^*[k]$ for $k \geq 1$. Now, assume $\mathbf{Q}^* \notin \mathcal{C}_i$ for all i for a contradiction. Item 2) of Lemma 11 implies $\mathbf{Q}_{\mathsf{eq}} \in \mathsf{rebdr}(\mathcal{C}_i)$. In equilibrium, it follows that

$$\mathsf{rebdr}(\mathcal{C}_i) = \{\mathbf{Q} \in \mathbb{S}_+^n : \lambda_{\mathbf{P}} \mathbf{P}_i^{-1} + \lambda_i^{\mathsf{eq}} \mathbf{Q}_{\mathsf{eq}}, \lambda_{\mathbf{P}} + \lambda_i^{\mathsf{eq}} = 1\}$$

since $\mathbf{Q}_i[k] = \mathbf{Q}_{\mathsf{eq}}$ and with constant $\mathbf{P}_i = \mathbf{P}_i[k]$. This is, $\mathbf{Q}_{\mathsf{eq}}$ is at the intersection of the N lines $\{\mathsf{rebdr}(\mathcal{C}_i)\}_{i\in\mathcal{I}}$. Denote the star $\mathcal{S} = \bigcup_{i\in\mathcal{I}}\{\mathsf{rebdr}(\mathcal{C}_i)\}$ for the rest of the proof, and note that $\mathbf{Q}_{\mathsf{eq}} = \arg\min_{\mathbf{Q}\in\mathcal{S}} f(\mathbf{Q})$. On the other hand, note that, item 2) of Lemma 12 implies $\mathbf{Q}^* \in \mathcal{P} := \mathsf{co}\{\mathbf{P}_i^{-1}\}_{i=1}^{\mathsf{N}}$. Denote with $\mathsf{dim}(\mathcal{P})$ the dimension of the manifold $\mathcal{P} \subset \mathbb{S}_+^n$. Using these properties, we distinguish the following cases:

1) $\mathsf{dim}(\mathcal{P}) = 0$, with $\mathcal{P}$ consisting of a single point $\mathbf{Q}^* = \mathbf{P}_1^{-1} = \cdots = \mathbf{P}_{\mathsf{N}}^{-1}$. However, note that from (5.12) it follows that $\mathbf{Q}_i[k] = \mathbf{Q}_{\mathsf{eq}} = \mathbf{P}_i^{-1} = \mathbf{Q}^*$ and thus $\mathbf{Q}^* \in \mathcal{C}_i$, which is a contradiction.

2) $\mathsf{dim}(\mathcal{P}) = 1$ and $\mathbf{Q}_{\mathsf{eq}} \in \mathcal{P}$. In this case, $\mathcal{P}$ necessarily consists of a line segment from $\mathbf{P}_i^{-1}$ and $\mathbf{P}_j^{-1}$ with $i \neq j$. Note that in this case, the line segment must comply $\mathcal{P} = \mathcal{S}$ for the star $\mathcal{S}$. However, this imply $f(\mathbf{Q}_{\mathsf{eq}}) \leq f(\mathbf{Q}^*)$. Due to strong convexity of f, either $f(\mathbf{Q}_{\mathsf{eq}}) < f(\mathbf{Q}^*)$ which is impossible, or $\mathbf{Q}^* = \mathbf{Q}_{\mathsf{eq}} = \mathbf{Q}_i[k]$ which leads to a contradiction similar to case 1).

3) $\mathsf{dim}(\mathcal{P}) \geq 1$ and $\mathbf{Q}_{\mathsf{eq}} \notin \mathcal{P}$. In this case, denote with $\mathcal{H} = \mathsf{co}(\mathcal{P} \cup \{\mathbf{Q}_{\mathsf{eq}}\})$. Note that $\mathcal{H}$ is a polytope over the (possibly) lower dimensional space $\mathsf{span}\{\mathbf{P}_1^{-1}, \dots, \mathbf{P}_{\mathsf{N}}^{-1}, \mathbf{Q}_{\mathsf{eq}}\} \subset \mathbb{S}_+^n$, with $\mathbf{Q}_{\mathsf{eq}}$ as one corner, with at least two faces touching it due to the dimension of $\mathcal{P}$. Let $\mathcal{W}(\alpha) := \{\mathbf{Q} \in \mathbb{S}_+^n : f(\mathbf{Q}) \leq \alpha\}$ the level sets of f, which are convex for any $\alpha \in \mathbb{R}$. Moreover, note that $\mathcal{W}(f(\mathbf{Q}_{\mathsf{eq}})) \subseteq \mathcal{C}^* \cap \mathcal{H}$. Otherwise, there would exist smaller $\alpha < f(\mathbf{Q}_{\mathsf{eq}})$ such that $\mathcal{W}(f(\mathbf{Q}_{\mathsf{eq}})) \cap \mathcal{S} \neq \varnothing$, which contradicts the fact that $f(\mathbf{Q}_{\mathsf{eq}}) \leq f(\mathbf{Q})$ for all $\mathbf{Q} \in \mathcal{S}$. Using the previous fact in combination with $\mathbf{Q}_{\mathsf{eq}} \in \mathcal{W}(f(\mathbf{Q}_{\mathsf{eq}}))$ hence, $\mathcal{W}(f(\mathbf{Q}_{\mathsf{eq}}))$ has at least two supporting hyper-planes at $\mathbf{Q}_{\mathsf{eq}}$, namely the faces of $\mathcal{H}$ meeting at that point. However, f is smooth by Assumption 3, which imply that $\mathcal{W}(f(\mathbf{Q}_{\mathsf{eq}}))$ must have a single supporting hyperplane at all points (see [Rockafellar, 1970, Theorems 23.3 and 25.1]), contradicting the previous fact that $\mathcal{W}(f(\mathbf{Q}_{\mathsf{eq}}))$ has two supporting hyper-planes.

4) $\mathsf{dim}(\mathcal{P}) > 1$ and $\mathbf{Q}_{\mathsf{eq}} \in \mathcal{P}$. Set $\mathbf{Q}' \in \mathsf{rebdr}(\mathcal{P})$ be the point such that $\mathbf{Q}^* \in \mathcal{P}$ is contained in the line segment between $\mathbf{Q}'$ and $\mathbf{Q}_{\mathsf{eq}}$. Pick an arbitrary face $\mathcal{P}' \subseteq \mathsf{rebdr}(\mathcal{P})$ of the polytope $\mathcal{P}$ with $\mathbf{Q}' \in \mathcal{P}'$, which is of lower dimension than $\mathcal{P}$. Hence, distinguish the

same cases 1), 2), 3) and 4), with $\mathcal{P}$ replaced by the face $\mathcal{P}'$, $\mathcal{H}$ with $\mathcal{H}' = \mathsf{co}(\mathcal{P}' \cup \{\mathbf{Q}_{\mathsf{eq}}\})$ and noting that $\mathbf{Q}^* \in \mathcal{H}'$ allowing to follow the same reasoning, reaching a contradiction in cases 1), 2) and 3) directly. Case 4) is used recursively until other cases are reached, which always happens since dimension of $\mathcal{P}$ is decreased every recursion step.

Henceforth, a contradiction is reached in any case implying that $\mathbf{Q}^* \in \mathcal{C}_i$ for some $i \in \mathcal{I}$. Then, item 1) of Lemma 12 allows the usage of item 1) of Lemma 11, which implies $\mathbf{Q}_i^* = \arg\min_{\mathbf{Q} \in \mathcal{C}_i} f(\mathbf{Q}) = \mathbf{Q}^*$, being $\mathbf{Q}_i^* = \mathbf{Q}_{\mathsf{eq}}$ the unique equilibrium of (5.12). □

5.5.3 Dynamic inputs

In this section, we provide a **proof for Theorem 12** in the general case with dynamic inputs.

Item 1): The result follows from item 3) of Lemma 12.

Item 2): We proceed by induction. For the induction base, note that $\mathbf{Q}_i[0] = \mathbf{P}_i[0]^{-1}$ from which the bound follows directly by Assumption 2. Now, assume the bound for arbitrary time $k-1 \in \{0, 1, \dots\}$. Item 3) of Assumption 3 and item 2) of Lemma 11 imply that $\mathbf{Q}_i[k] \in \mathsf{rebdr}(\mathcal{C}_i[k])$. As a result

$$\mathbf{Q}_i[k] = \lambda^i_{\mathbf{P}} \mathbf{P}_i[k]^{-1} + \frac{1}{\overline{\theta}} \sum_{j \in \mathcal{N}_i} \lambda^i_j \mathbf{Q}_j[k-1] \preceq \frac{1}{\underline{p}} \left(\lambda^i_{\mathbf{P}} + \frac{1}{\overline{\theta}} \sum_{j \in \mathcal{N}_i} \lambda^i_j \right) \mathbf{I} \preceq \frac{1}{\underline{p}} \mathbf{I}.$$

Item 3): First, let $\varepsilon, \underline{\theta}, \overline{\theta}$ be such that $[\underline{\theta}, \overline{\theta}] \subseteq [1-\varepsilon, 1+\varepsilon]$. Note that if $\varepsilon = 0$, then $\overline{\theta} = \underline{\theta} = 1$ which corresponds to the static case. Hence, we expect that small $\delta > 0$ will lead to a small $\varepsilon > 0$, preventing it to be arbitrarily big. To quantify this, we now show the existence of $K \geq 1, \varepsilon > 0$ such that $\|\mathbf{Q}_i[K] - \mathbf{Q}^*[K]\| < \delta$ independently of the initial conditions, by repeated use of a continuity argument. We denote the trajectories of a nominal version of the algorithm (5.12) with constant inputs $\mathbf{P}_i[k] = \mathbf{P}_i[0]$ as $\mathbf{Q}_i^{\varepsilon=0}[k], k \geq 1$. Due to stability and optimally of the nominal system established in Lemma 14, for any $\delta_1 > 0$ there exist $K' \geq 0$ such that $\|\mathbf{Q}_i^{\varepsilon=0}[K'] - \mathbf{Q}^{*,\varepsilon=0}[K']\| < \delta_1$. Moreover, note that $(1-\varepsilon)^{K'} \mathbf{P}_i[0] \preceq \mathbf{P}_i[K'] \preceq (1+\varepsilon)^{K'} \mathbf{P}_i[0]$ by Assumption 2. Henceforth, for any $\delta_2 > 0$, there exists $\varepsilon > 0$ sufficiently small to make $\mathbf{P}_i[K']$ as close as desired to $\mathbf{P}_i[0]$ to make $d_H(\mathcal{C}^*[K'], \mathcal{C}^{*,\varepsilon=0}[K']) < \delta_2$, where d_H denotes the Hausdorff distance and $\mathcal{C}^{*,\varepsilon=0}[K]$ the feasible set in (5.3) in the nominal case. Similarly, for any $\delta_3 > 0$ there exists $\varepsilon > 0$ such that $d_H(\mathcal{C}_i[K'], \mathcal{C}_i^{\varepsilon=0}[K']) < \delta_3$. Henceforth, by picking $\delta_1, \delta_2, \delta_3$ appropriately, $\varepsilon > 0$ exist to make $\mathbf{Q}_i[K']$ as close as desired to $\mathbf{Q}_i^{\varepsilon=0}[K']$ (with error related to δ_3), to $\mathbf{Q}^{*,\varepsilon=0}[K']$ (with error related to δ_1) and to $\mathbf{Q}^*[K']$ (with error related to δ_2) in that order, allowing $\|\mathbf{Q}_i[K'] - \mathbf{Q}^*[K']\| < \delta$ for some δ.

Note that such ε, K' depend on the initial conditions $\mathbf{Q}_i[0]$ since they might be different between trajectories. Make this dependence explicit in $K' = K'(\mathbf{Q}_1[0], \dots, \mathbf{Q}_{\mathsf{N}}[0])$ and note that $\overline{K}(\delta) := \sup K'(\mathbf{Q}_1[0], \dots, \mathbf{Q}_{\mathsf{N}}[0])$ exists since $\mathbf{Q}_1[0], \dots, \mathbf{Q}_{\mathsf{N}}[0]$ lie in a compact set defined by $\mathbf{Q}_i[0] = \mathbf{P}_i[0]^{-1} \preceq \underline{p}\mathbf{I}$. Similarly, $\overline{\varepsilon}(\delta) = \inf \varepsilon(\mathbf{Q}_1[0], \dots, \mathbf{Q}_{\mathsf{N}}[0]) > 0$ exists. Henceforth, for any $0 < \varepsilon < \overline{\varepsilon}(\delta)$ it follows that $\|\mathbf{Q}_i[\overline{K}(\delta)] - \mathbf{Q}^*[\overline{K}(\delta)]\| < \delta$ regardless of

the initial conditions.

Assume that $\|\mathbf{Q}_i[k] - \mathbf{Q}^*[k]\| < \alpha$ holds for some arbitrary $k \geq 0$ and $\alpha < \delta$. Now, we show $\|\mathbf{Q}_i[k+1] - \mathbf{Q}^*[k+1]\| < \alpha$. The property follows by making ε and $\alpha > 0$ sufficiently small by taking $\mathbf{Q}^*[k]$ as close as desired to $\mathbf{Q}^*[k+1]$ and $\mathbf{Q}_i[k]$ close to $\mathbf{Q}_i[k+1]$, using a similar reasoning as before. Hence, there exists $\tilde{\varepsilon} = \inf \varepsilon(\mathbf{Q}_1[k], \dots, \mathbf{Q}_{\mathsf{N}}[k]) > 0$ and $\tilde{\alpha} = \inf \alpha(\mathbf{Q}_1[k], \dots, \mathbf{Q}_{\mathsf{N}}[k])$ if the infimum is taken over the set of possible $\mathbf{Q}_i[k]$, which is compact by the boundedness property established in item 2) of Theorem 12.

Henceforth, we take $\varepsilon < \min(\tilde{\varepsilon}, \overline{\varepsilon}(\tilde{\alpha}))$ and $K = \overline{K}(\tilde{\alpha})$. This allows $\|\mathbf{Q}_i[K] - \mathbf{Q}^*[K]\| < \tilde{\alpha}$, maintaining such property for all subsequent steps. The result follows by noting that $\tilde{\alpha} < \delta$ and that, if $\varepsilon = 0$, the static case is recovered, optimal from Lemma 14.

5.6 Application: distributed Kalman filter

To further motivate the distributed discrete-time dynamic outer approximation of the intersection of ellipsoids, we exemplify how to exploit Algorithm 9 to improve the mean square error performance of distributed Kalman filtering. To achieve this, we first need to ensure consistency, i.e., the fusion of predicted covariance matrices is such that the updated covariance matrices are still (tight) outer-approximations of the true covariance we would have if all the nodes tracked all the cross-correlations among nodes (see [Julier and Uhlmann, 2017] for further details). The next proposition proves consistency in the fusion of predicted covariances using Algorithm 9.

Proposition 15. *Let Assumptions 2-3 hold. Moreover, let $\{\mathbf{x}[k]\}_{k\geq 0}$ be a Gaussian stochastic process and $\bar{\mathbf{x}}_i[k]$, $i \in \mathcal{I}$ be Gaussian distributed correlated unbiased estimates for $\mathbf{x}[k]$ with covariance matrices given by $\mathbf{P}_i[k]$ and unknown correlations. Let $\mathbf{Q}^*[k]$ the solution to (5.3) with its corresponding weights $\{\lambda_i^*\}_{i\in\mathcal{I}}$ and*

$$\hat{\mathbf{x}}[k] = \mathbf{Q}^*[k]^{-1} \sum_{j\in\mathcal{I}} \lambda_j^* \mathbf{P}_j[k]^{-1} \bar{\mathbf{x}}_j[k]. \tag{5.14}$$

Then, the covariance matrix $\mathsf{cov}\{\hat{\mathbf{x}}[k] - \mathbf{x}[k]\} \preceq \mathbf{Q}^*[k]^{-1}$ always.

Proof. Assumption 3 implies $\mathbf{Q}^*[k] = \sum_{j\in\mathcal{I}} \lambda_j^* \mathbf{P}_j[k]^{-1}$ due to $\mathbf{Q}^*[k] \in \mathsf{rebdr}(\mathcal{C}^*[k])$. This means that (5.14) is the standard covariance intersection fusion rule, whose consistency was proven in [Niehsen, 2002]. □

Proposition 15 exploits consistency to guarantee that the estimates at each node can be fused when cross-correlations are unknown, either for static of dynamic input matrices. The previous result motivates estimating (5.14) in a distributed way:

$$\mathbf{Q}_i[k]\hat{\mathbf{x}}_i[k] = \frac{1}{\mathsf{N}} \sum_{j\in\mathcal{I}} \left(\frac{\lambda_{\mathbf{P}}^j[k]}{1 - \sum_{l\in\mathcal{N}_j} \lambda_l^j[k]} \right) \mathbf{P}_j[k]^{-1} \bar{\mathbf{x}}_j[k], \tag{5.15}$$

where the right hand side of (5.15) can be computed using standard dynamic consensus tools with local inputs $\left(\frac{\lambda_{\mathbf{P}}^j[k]}{1-\sum_{l\in\mathcal{N}_j} \lambda_l^j[k]} \right) \mathbf{P}_j[k]^{-1} \bar{\mathbf{x}}_j[k]$.

Proposition 16. *Consider the same setting as in Proposition 15. Moreover, let* $\bar{\theta} = 1$. *Then, the update rule in eq.* (5.14) *leads to* $\lim_{k\to\infty} \|\hat{\mathbf{x}}_i[k] - \hat{\mathbf{x}}[k]\| = 0$.

Proof. First, let $\lambda_i^* = \frac{1}{\mathsf{N}} \left(\frac{\lambda_{\mathbf{P}}^i}{1-\sum_{j\in\mathcal{N}_i} \lambda_j^i} \right)$, where $\lambda_{\mathbf{P}}^i$ and λ_j^i come from (5.12) in equilibrium at agent $i \in \mathcal{I}$. Then,

$$\mathbf{Q}[k]^* \preceq \sum_{j=1}^{\mathsf{N}} \lambda_j^* \mathbf{P}_j[k]^{-1}. \tag{5.16}$$

and $\sum_{j=1}^{\mathsf{N}} \lambda_j^* \leq 1$ with $\lambda_j^* \geq 0$. Moreover, if Assumption 3 is satisfied, (5.16) is complied with equality. Now, note that Lemma 14 implies that in equilibrium all agents are in consensus at $\mathbf{Q}_i[k] = \mathbf{Q}^*$ with constant $\mathbf{Q}^* = \mathbf{Q}^*[K]$. Moreover, due to the constraints in (5.12): $\mathbf{Q}_i[k] = \mathbf{Q}^* \preceq \lambda_{\mathbf{P}}^i \mathbf{P}_i[k]^{-1} + \sum_{j\in\mathcal{N}_i} \lambda_j^i \mathbf{Q}^*$ implies $\mathbf{Q}^* \Big(- \sum_{j\in\mathcal{N}_i} \lambda_j^i \Big) \preceq \lambda_{\mathbf{P}}^i \mathbf{P}_i[k]^{-1}$. Now, dividing by $1 - \sum_{j\in\mathcal{N}_i} \lambda_j^i$ and applying a summation over all agents in the previous relation: $\mathsf{N}\mathbf{Q}^* \preceq \sum_{i=1}^{\mathsf{N}} \left(\frac{\lambda_{\mathbf{P}}^i}{1-\sum_{j\in\mathcal{N}_i} \lambda_j^i} \right) \mathbf{P}_i[k]^{-1}$ which reduces to (5.16) by the definition of λ_i^*. Now, note that $0 \leq \lambda_{\mathbf{P}}^i \leq 1 - \sum_{j\in\mathcal{N}_i} \lambda_j^i$ implying $\lambda_i^* \geq 0$ and as a result that $\lambda_i^* \leq 1/\mathsf{N}$. Hence, $\sum_{j=1}^{\mathsf{N}} \lambda_j^* \leq 1$. In addition, if Assumption 3 is satisfied, the all previous inequalities are interchanged with equalities.

As a result, the proof of the proposition follows by noting that (5.16) is satisfied with equality as well as Theorem 12 ensuring $\lim_{k\to\infty} \|\mathbf{Q}_i[k] - \mathbf{Q}^*[k]\| = 0$. □

The results of Proposition 15 and 16 lead to a novel distributed Kalman filter with the following four steps: at instant k and given $\hat{\mathbf{x}}_i[k-1], \hat{\mathbf{P}}_i[k-1]$, each node (i) predicts $\bar{\mathbf{x}}_i[k], \bar{\mathbf{P}}_i[k]$ using the linear stochastic dynamics $\mathbf{x}[k] = \mathbf{A}\mathbf{x}[k-1] + \mathbf{w}[k]$, with $\mathbf{w}[k]$ a zero-mean Gaussian noise with covariance $\mathbf{W}$; (ii) exchanges of $\bar{\mathbf{x}}_i[k], \bar{\mathbf{P}}_i[k], \mathbf{Q}_i[k-1]$ with neighbors $i \in \mathcal{N}_i$; (iii) uses Algorithm 9 to obtain $\mathbf{Q}_i[k]$, which is employed as the new covariance matrix to be updated; (iv) updates the predictions using (5.14) and the Kalman filter update equations using the measurements $\mathbf{y}_i[k] = \mathbf{H}_i \mathbf{x}_i[k-1] + \mathbf{v}_i[k]$, with $\mathbf{v}_i[k]$ a zero-mean Gaussian noise with covariance $\mathbf{V}_i$, obtaining $\hat{\mathbf{x}}_i[k], \hat{\mathbf{P}}_i[k]$. We refer to [Olfati-Saber, 2007b, Sebastián et al., 2021c] for the explicit expressions for the prediction and correction steps of distributed Kalman filtering. It is interesting to remark that, even though Eq. (5.14) is similar to a batch version of CI, it is not since it incorporates information from all nodes through $\mathbf{Q}^*[k], \lambda_j^*$, which are computed using Algorithm 9. Hence, while existing methods are limited to local CI, our method can calculate a global CI once Algorithm 9 has converged.

5.7 Illustrative examples

In this section, we simulate the proposed methods to illustrate their properties.

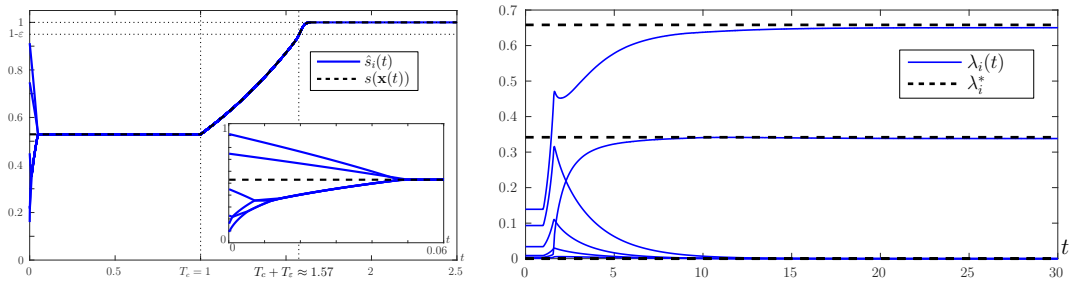


Figure 5.2: (left) Convergence of the estimates $\hat{s}_i(t)$ which reach consensus towards $s(\mathbf{x}(t))$ at $t \approx 0.05$, with zoom in $t \in [0, 0.06]$. Moreover, the evolution of the surface $s(\mathbf{x}(t))$ towards achieving $\mathbf{x}(t) \in \mathcal{C}$ is shown. (right) Evolution of the trajectories $\lambda_i(t) = x_i(t)^2/\mathsf{N}$ which converge towards the global optimal values of (5.1) asymptotically, up to error less than $\varepsilon = 0.05$.

5.7.1 Continuous-time algorithm

We assume that each agent reads a noisy measurement from a sensor with covariance matrix $\{\mathbf{P}_i\}_{i=1}^{\mathsf{N}}$. For the sake of generality, we set $\mathbf{P}_i = \mathbf{M}_i^\top \mathbf{M}_i$ where $\mathbf{M}_i$ was drawn with uniformly distributed components. The purpose of each agent i is to compute $\mathsf{N}\lambda_i, \mathbf{P}(\boldsymbol{\lambda})$ from $x_i(t)^2, \hat{\mathbf{Q}}_i(t)^{-1}$. We choose $f(\mathbf{Q}(\mathbf{x})) = \mathrm{tr}(\mathbf{Q}(\mathbf{x})^{-1})$ as a performance index.

All the differential equations, namely, (5.4) and $\dot{x}_i(t) = u_i(t)$ under (5.5), where simulated using the forward Euler method with time step $\Delta t = 10^{-4}$. In the sake of interpretability, we set $n = 2$ and a graph $\mathcal{G}$ of $\mathsf{N} = 6$ agents with edges

$$\mathcal{F} = \{(1,4), (1,5), (2,4), (2,6), (3,6), (5,6)\}.$$

In addition, we set $\varepsilon = 0.05$ as well as $T_c = 1, q = 1/2, \underline{b} = 0.1, \overline{b} = 1.1, \kappa_{\mathcal{C}} = 0.1, \kappa_s = \kappa_{\mathbf{Q}} = 10, \zeta_{\mathbf{Q}} = \zeta_s = 1, \zeta_g = 4$. Figure 5.2a shows the local estimates $\hat{s}_i(t)$ which reach consensus towards the global value of $s(\mathbf{x}(t))$ at $t \approx 0.05$, occurring before $t = T_c = 1$, which is a conservative prescribed convergence time bound. After $t = T_c$, $s(\mathbf{x}(t))$ increases until it reaches $s(\mathbf{x}(t)) = 1 - \varepsilon$, which means that $\mathbf{x}(t) \in \mathcal{C}$ from $t \geq T_c + T_\varepsilon$ with $T_\varepsilon \approx 0.57$. Figure 5.2b shows that, once $\mathbf{x}(t) \in \mathcal{C}$, then, the trajectories of $\lambda_i(t) = x_i(t)^2/\mathsf{N}$ converge towards the global minimizer of (5.2) which corresponds to $\lambda_1^* = \lambda_2^* = \lambda_3^* = \lambda_4^* = 0, \lambda_5^* = 0.6582, \lambda_6^* = 0.3418$ in this example, up to some error less than $\varepsilon = 0.05$. Finally, Figure 5.3 show the ellipses $\{\mathcal{E}(\mathbf{P}_i)\}_{i=1}^{\mathsf{N}}$ as well as $\mathcal{E}(\mathbf{Q}(\mathbf{x}(t))^{-1})$ with $t = T_c + T_\varepsilon$ and $t = 30$, which shows that $\mathcal{E}(\mathbf{Q}(\mathbf{x}(t))^{-1}) \supset \check{\mathcal{E}}$ in both cases, but the latter case leads to a tighter outer LJ ellipse.

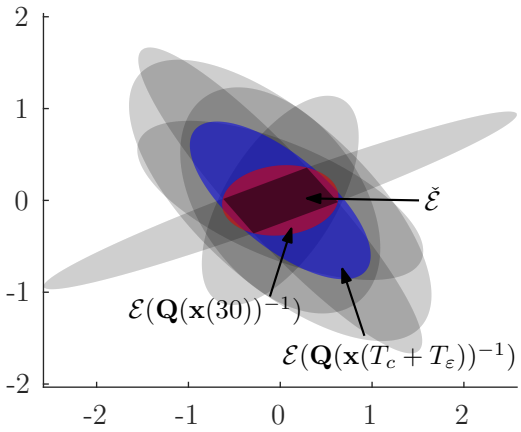


Figure 5.3: Ellipses $\{\mathcal{E}(\mathbf{P}_i)\}_{i=1}^{\mathsf{N}}$ (shown in grey) as well as $\mathcal{E}(\mathbf{Q}(\mathbf{x}(t))^{-1})$ with $t = T_c + T_\varepsilon$ (blue) and $t = 30$ (red).

Moreover, we evaluate the time it takes for the agents to reach consensus T_{cons} and T_ε by repeating the same experiment as before with a circular graph $\mathcal{G}$ of $\mathsf{N} = 10$ and

20 agents and with $n = 2$ and 4. We set $\kappa_s, \kappa_{\mathsf{Q}}, \zeta_s, \kappa_{\mathsf{Q}}$ the same in all cases, adjusted accordingly to account for $\ell, \lambda_{\mathcal{G}}$ in both networks as required in Theorem 12. The results are summarized in Table 5.2 where it is observed that T_{cons} increases with N as expected from [Aldana-López et al., 2019]. On the other hand, T_{cons} is similar as before when n is increased. The reason is that (5.4) is executed in parallel for each component of $\hat{\mathbf{Q}}_i(t)$ such that there is no influence of n in the protocol convergence itself. Note that T_ε is similar in all cases since once consensus has been reached, the system is equivalent to (5.6) which does not depend on N, n for $\mathbf{x}(t) \notin \mathcal{C}$.

N	ℓ	$\lambda_{\mathcal{G}}$	n	T_{cons} [s]	T_ε [s]
10	9	0.3819	2	0.0371	0.169
			4	0.0368	0.172
20	19	0.0978	2	0.085	0.170
			4	0.088	0.181

Table 5.2: Values for the consensus and feasible region reaching times $T_{\mathsf{cons}}, T_\varepsilon$, for a circular graph with N nodes.

5.7.2 Discrete-time with constant inputs

First, we evaluate Algorithm 9 in the static case, i.e., $\mathbf{P}_i[k] = \mathbf{P}_i \ \forall i \in \mathcal{I}$. We generate a random connected graph $\mathcal{G}$ of $\mathsf{N} = 6$ nodes, leading to the following edge set: $\mathcal{F} = \{(1,2),(1,3),(1,5),(2,4),(2,5),(2,6)\}$. Input matrices are of dimension $n = 2$, and they are initialized as $\mathbf{P}_i = \mathbf{L}_i^\top \mathbf{L}_i$, where the elements of each $\mathbf{L}_i$ are randomly generated using a uniform distribution between -1 and 1. The resulting ellipses are characterized by the following matrices:

$$\mathbf{P}_1^{-1}=\begin{pmatrix} 4.6 & -3.8 \\ -3.8 & 4.2 \end{pmatrix} \mathbf{P}_2^{-1}=\begin{pmatrix} 1.5 & -0.2 \\ -0.2 & 2.0 \end{pmatrix} \mathbf{P}_3^{-1}=\begin{pmatrix} 9.5 & 0.4 \\ 0.4 & 2.3 \end{pmatrix}$$

$$\mathbf{P}_4^{-1}=\begin{pmatrix} 2.8 & -2.2 \\ -2.2 & 4.5 \end{pmatrix} \mathbf{P}_5^{-1}=\begin{pmatrix} 11.0 & 7.9 \\ 7.9 & 6.7 \end{pmatrix} \quad \mathbf{P}_6^{-1}=\begin{pmatrix} 11.5 & -3.9 \\ -3.9 & 3.1 \end{pmatrix}$$

Since the input ellipsoids are constant, we set $\bar{\theta} = 1$. Also, let $q_1^i[k] \leq q_2^i[k] \leq \ldots \leq q_n^i[k]$ the sorted eigenvalues of $\mathbf{Q}_i[k]$ and $q_1^*[k] \leq q_2^*[k] \leq \ldots \leq q_n^*[k]$ the sorted eigenvalues of $\mathbf{Q}^*[k]$.

Fig. 5.4 shows the evolution of the estimates at each node with time. To measure the difference between $\mathbf{Q}_i[k]$ and $\mathbf{Q}^*$ we compute the absolute error between their sorted eigenvalues as $|q_j^i[k] - q_j^*| \ \forall j \in \{1,\ldots,n\}$ and $\forall i \in \mathcal{I}$. We can observe how the estimates converge in finite time to the global optimum of the centralized outer Löwner-John method, with $K = 3$. By focusing on node $i = 1$, we can see how the estimates are refined at each time step, reaching the global optimum at $k = 2$. The value of K depends on the topology and how the input ellipsoids are across the network. However, for any case, the global minimum is perfectly attained in finite time.

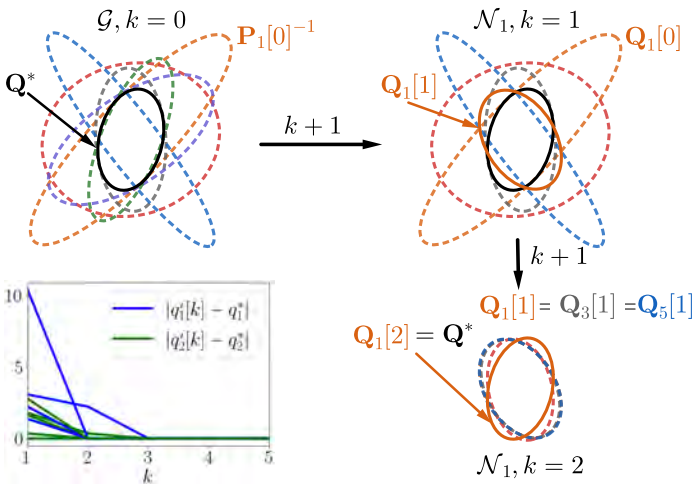


Figure 5.4: Illustrative example of Algorithm 9 applied to a static problem. At each time step, the error between the eigenvalues of the global optimum matrix and the local estimates $|q_j^i[k] - q_j^*|$ decreases until the error becomes zero in finite time (in this case, $K = 3$). The top left pannel shows the initial ellipsoids at each node and the optimal solution from the original centralized Löwner-John method. The top right pannel shows, for node 1, its estimate after using Algorithm 9 at $k = 1$ and the ellipsoids exchanged with its neighbors. The bottom right pannel depicts the estimate at node 1 and $k = 2$ after using again Algorithm 9, depicting that node 1 already recovers the desired $\mathbf{Q}^*$. The bottom left pannel depicts the evolution of the error with time for all the nodes.

5.7.3 Discrete-time with dynamic inputs

For the dynamic case where the input matrices evolve with time, we use the same graph from Subsection 5.7.2. We use the same initial ellipsoids of Section 5.7.2 and apply a different oscillatory rotation to each of them at each time step. In particular, we set $\mathbf{P}_i[k] = (1+\phi_i[k])\mathbf{R}_i[k]^\top \mathbf{P}_i[0]\mathbf{R}_i[k]$ where $\mathbf{R}_i[k]$ is a 2D rotation matrix with rotation angle $\psi_i[k] = A\sin(\omega_i k)$, and $\phi_i[k] = B\sin(\omega_i k)$. We choose $A = \frac{\pi}{25}$, $B = \frac{1}{200}$ and $\omega_i \in [1, 2]$. It can be shown that $\bar{\theta} = \frac{1}{0.98}$ complies Assumption 2.

Fig. 5.5 depicts the evolution of the error between the eigenvalues of the global optimum matrix and the average of the local estimates $|\tilde{q}_j[k] - q_j^*|$ with time, where it holds that $\tilde{q}_j[k] = \frac{1}{\mathsf{N}}\sum_{i=1}^{\mathsf{N}} q_j^i[k]$. It is observed the error trajectories converge in finite-time to some bounded region around the global optimum after a transient, which is maintained invariant for the rest of the experiment, consistent with the sub-optimality result in Theorem 12.

5.7.4 Application example: distributed Kalman filtering

We compare the our novel distributed Kalman filter developed in Section 5.6 with the consensus distributed Kalman filter in [Olfati-Saber, 2007b] (CDKF) and the equivalent centralized Kalman filter that we obtain by collecting all the measurements acquired by each node at a central server, fusing them and using a standard Kalman filter [Kalman, 1960]. The former serves as a comparison with an establish distributed Kalman filter whereas the latter serves as a baseline. We run 100 simulations with randomly generated connected graphs of $\mathsf{N} = 50$ nodes. The target system is the same in [Sebastián et al., 2021c], and the

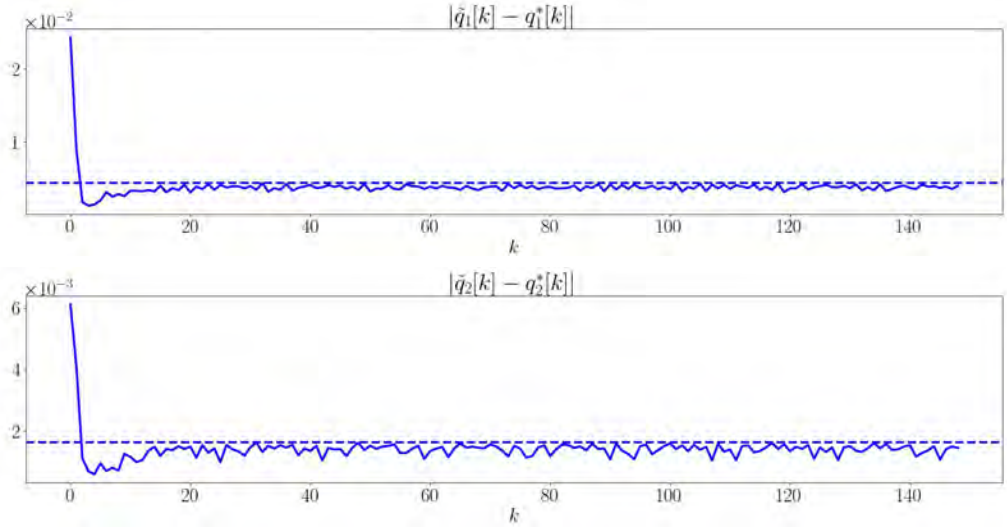


Figure 5.5: Illustrative example of Algorithm 9 applied to a dynamic problem. The plot depicts the time evolution of the error between the eigenvalues of the global optimum matrix and the average of the local estimates $|\tilde{q}_j[k] - q_j^*|$, for $j \in \{1, 2\}$.

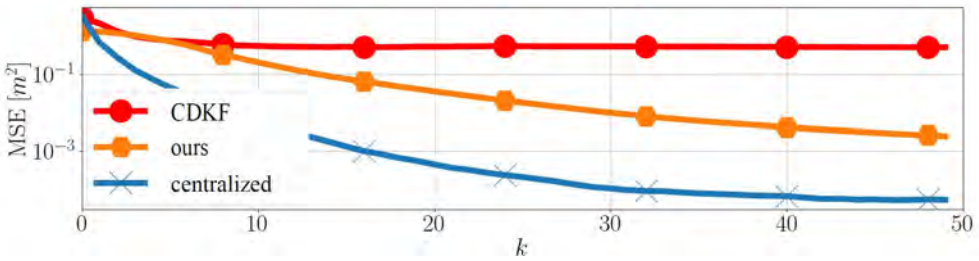


Figure 5.6: Evolution of the averaged mean square error with time for our novel distributed Kalman filter, the CDKF and the equivalent centralized Kalman filter.

ellipsoids are of $n = 4$. We replicate the experimental setting reported in [Sebastián et al., 2021c], choosing $\bar{\mathbf{x}}_i[0], \bar{\mathbf{P}}_i[0]$ randomly. We set $\mathbf{W} = 2 \times 10^{-8}\mathbf{I}$ and $\mathbf{V}_i = \mu_i \mathbf{I}$ with μ_i drawn from a uniform distribution between 0.03 and 0.05.

Fig. 5.6 reports the average of the mean square error across nodes and simulation runs for the two distributed Kalman filters: $\text{MSE}[k] = \frac{1}{100}\sum_{s=1}^{100}\frac{1}{\mathsf{N}}\sum_{i=1}^{\mathsf{N}}||\bar{\mathbf{x}}_i^s[k] - \mathbf{x}^s[k]||^2$. Our proposed algorithm surpasses CO-DKF by exploiting a better approximation of the true fused covariance matrix under known correlations: while our approach tracks the global minimum across all the nodes, CO-DKF only computes the outer approximation of the intersection of the neighboring ellipsoids. The difference between both approximations is greater for larger and sparser networks.

5.8 Conclusions

In this chapter, we have focused on a fundamental question: Is it possible to compute the equivalent centralized optimum of the outer Löwner-John ellipsoid method across the

network using distributed techniques? We have found that the answer is yes. To address this challenge, we have developed two novel distributed algorithms.

The first is a continuous-time algorithm designed for static scenarios where the ellipsoids at each node remain constant over time. This algorithm provides a theoretical foundation for understanding the problem and its potential solutions, exploiting fundamentals on project gradient flow methods and exact dynamic consensus. However, it has limitations in terms of applicability to dynamic, real-world situations, where the computation units work at discrete steps.

To overcome these limitations, we have also developed a second algorithm: a discrete-time algorithm capable of handling both static and time-varying ellipsoids. This algorithm is particularly well-suited for sensor fusion applications, as demonstrated in the chapter. It allows robots to estimate global properties of the system, such as the shape of a herd (Chapter 2) or the collective uncertainty of a sensor network, by exchanging information only with their neighbors. The discrete-time algorithm achieves this by reformulating the centralized outer Löwner-John method into a local semi-definite program. Robots then exchange and scale their estimates of the global optimum with their neighbors. In static problems, this process leads to perfect recovery of the global optimum in finite time. In dynamic problems, the algorithm tracks the global optimum in finite time with bounded accuracy.

We have provided a rigorous theoretical analysis of both algorithms, proving their convergence properties. Additionally, we have conducted numerical experiments to illustrate their effectiveness in practice. The results confirm that our distributed algorithms can achieve the same level of accuracy as the centralized method, while also offering the advantages of scalability and robustness.

Part III

Physics-Informed Learning of Distributed Interactions

Chapter 6

Learning Distributed Multi-robot Interactions exploiting Physics

The methods derived in Part I and II offer improved performance in distributed multi-robot problems. However, something that we have not discussed so far is how to obtain the dynamical models subsequently exploited by the distributed estimation and control techniques to achieve the desired goals. For instance, the Implicit Control presented in Chapter 2 requires an analytical model of the behavior of the evaders. In general, it turns our that the complexity of describing mathematically the objective and constraints in many of multi-robot problems makes the design of analytical controllers a challenging task. To address this issue, there exist two fundamental approaches:

- Multi-agent reinforcement learning [Matarić, 1997, Matignon et al., 2007, Matignon et al., 2012, Munikoti et al., 2023, Serra-Gómez et al., 2023] addresses this issue by only requiring a high-level mathematical specification of the task (the reward function), which is commonly available.
- Imitation learning [Jiahao et al., 2022, Bloembergen et al., 2015, Khan et al., 2020, Tolstaya et al., 2020, Tolstaya et al., 2021, Yang and Matni, 2021, Gama et al., 2022, Furieri et al., 2022b, Han et al., 2020, Shi et al., 2020, Long et al., 2018, Semnani et al., 2020, Dasari et al., 2020, Bogert and Doshi, 2018, Zhu et al., 2021, Zhou et al., 2019] addresses this issue by exploiting information from expert demonstrations of the task.

A fundamental limitation of existing approaches in both formulations is the poor scalability they offer against increasing and time-varying numbers of robots. Centralized control policies do not scale whereas independent control policies neglect the information that other robots can offer. How to design and train control policies based on neural networks that are distributed and use all the available information is still an open problem [Lo et al., 2023, Qu et al., 2022, Munikoti et al., 2023].

The main contribution of this chapter is a novel distributed and scalable by design neural network architecture to describe multi-robot control policies. This is achieved by

combining a physics-informed port-Hamiltonian description of the multi-robot system with self-attention. The former naturally encodes the distributed nature of the policy and respects the energy conservation laws of the individual dynamics of the robots. The latter handles the information coming from communication or perception in time-varying neighborhoods.

6.1 Related work

The learning of distributed interactions for multi-robot control deals with two challenges.

The first challenge is the mathematical formulation of the problem, where we must consider the task and the constraints inherited from the cooperative-competitive nature of the multi-robot team. To address this, recent works exploit machine learning and focus on learning control policies for optimal control or reinforcement learning problems [Bloembergen et al., 2015, Long et al., 2018, Semnani et al., 2020]. When demonstrated data from an expert are available, inverse reinforcement learning [Ng et al., 2000] can be used to learn centralized [Dasari et al., 2020, Bogert and Doshi, 2018] or distributed [Han et al., 2020, Gharbi et al., 2023] policies from task demonstrations. However, finding experts for multi-robot applications is difficult. It is also possible to apply supervised learning approaches to learn multi-robot control policies [Zhu et al., 2021, Zhou et al., 2019], but, again, collecting labeled trajectories is hard.

The second challenge is that the learning and execution of control policies for multi-robot systems should scale favorably with an increasing numbers of robots. Learning a joint policy function is challenging due to the exponential growth of the state and action space [Qu et al., 2020]. Attention mechanisms are widely used in multi-agent reinforcement learning problems [Zambaldi et al., 2018, Iqbal and Sha, 2019, Li et al., 2020a, Parnika et al., 2021] to enhance the performance of centralized training settings where the agents are isolated from each other and do not consider communication during deployment. Graph neural networks [Munikoti et al., 2023] have been utilized as a stable [Marino et al., 2023], scalable and communication-aware policy representation in path planning, coverage, exploration, and flocking problems [Li et al., 2021a, Khan et al., 2020, Tolstaya et al., 2020, Tolstaya et al., 2021, Yang and Matni, 2021, Gama et al., 2022]. These approaches assume discrete robot dynamics, fixed or known communication topology, or prior knowledge on the formulation of the control policy.

6.2 Problem formulation

Consider a team of robots, indexed by $\mathcal{V} = \{1, \dots, n\}$. The robot team motion is governed by *known* continuous-time control-affine stochastic dynamics:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) + \mathbf{L}\omega(\mathbf{t}), \tag{6.1}$$

where $\mathbf{x}(t) = [(\mathbf{x}^1(t))^\top, \dots, (\mathbf{x}^n(t))^\top] \in \mathcal{X} \subseteq \mathbb{R}^{n \times n_x}$ is the joint state of the robot team at time $t \geq 0$, with $\mathbf{x}^i(t)$ the state of robot i at time t. On the other hand, we define the joint input as $\mathbf{u} = [(\mathbf{u}^1(t))^\top, \dots, (\mathbf{u}^n(t))^\top] \in \mathcal{U} \subseteq \mathbb{R}^{n \times n_u}$, with $\mathbf{u}^i(t)$ the input of robot i at time t. The term $\boldsymbol{\omega}(t)$ is white noise modeling the uncertainty in the robot sensors and actuators and $\Xi = \mathbf{L}\mathbf{L}^\top$ is the noise diffusion matrix. Let $\{t_\tau\}_{\tau=0}^{\infty}$ be a sequence of discrete time instants such that $t_{\tau+1} - t_\tau = T > 0$ and assume zero-order hold inputs such that $\mathbf{u}(t) = \mathbf{u}(t_\tau)$, $\forall t \in [t_\tau, t_{\tau+1})$. Then, from Eq. (6.1), we can obtain an Euler discretization of the dynamics with discrete-time state $\mathbf{s}_\tau = \mathbf{x}(t_\tau)$, action $\mathbf{a}_\tau = \mathbf{u}(t_\tau)$, and dynamics:

$$\mathbf{s}_{\tau+1} = \mathbf{s}_\tau + T\mathbf{f}(\mathbf{s}_\tau, \mathbf{a}_\tau) + \mathbf{n}_\tau, \tag{6.2}$$

where $\mathbf{n}_\tau$ is zero-mean Gaussian noise with covariance $T\Xi$. We will formulate the dynamics $\mathbf{f}$ using port-Hamiltonian mechanics as its modularity in terms of energy effectively describes the networked interactions in a robot team.

The multi-robot task and the interactions among the robots are modeled as a Markov Decision Process (MDP), defined as a tuple $(\mathcal{X}, \mathcal{U}, p, r, \gamma)$. In the tuple, $p : \mathcal{X} \times \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ is the probability density of the next joint state $\mathbf{s}_{\tau+1}$ conditioned on the current joint state $\mathbf{s}_\tau$ and joint action $\mathbf{a}_\tau$, $r : \mathcal{X} \times \mathcal{U} \to [r_{\min}, r_{\max}]$ is a reward function encoding the objectives of the multi-robot task, and $\gamma \in (0, 1)$ is the discount factor. Whereas the robot dynamics are formulated in continuous time, the MDP is formulated in discrete time based on the zero-order hold discretization in Eq. (6.2). Accordingly, $p(\mathbf{s}_{\tau+1}|\mathbf{s}_\tau, \mathbf{a}_\tau)$ is a Gaussian density with mean $\mathbf{s}_\tau + T\mathbf{f}(\mathbf{s}_\tau, \mathbf{a}_\tau)$ and covariance $T\Xi$.

The robots interact in a distributed manner, described by a time-varying undirected graph $\mathcal{G}_\tau = (\mathcal{V}, \mathcal{E}_\tau)$, where $\mathcal{E}_\tau \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. An edge $(i, j) \in \mathcal{E}_\tau$ exists when robots i and j interact at time τ. Let $\mathbf{A}_\tau \in \{0, 1\}^{n \times n}$ be the adjacency matrix associated to $\mathcal{G}_\tau$, such that $[\mathbf{A}_\tau]_{ij} = 1$ if and only if $(i, j) \in \mathcal{E}_\tau$, and 0 otherwise. The set of k-hop neighbors of robot i at time τ is $\mathcal{N}_\tau^{i,k} = \{j \in \mathcal{V} \mid [\mathbf{A}_\tau^k]_{ij} \neq 0\}$, where $\mathbf{A}_\tau^k$ is the k-th power of matrix $\mathbf{A}_\tau$. We remark that $\mathcal{N}_\tau^{i,k}$ includes robot i.

The goal is to learn distributed control policies that solve a given multi-robot task, such that they respect the networked structure of the multi-robot team and the MDP model. We represent the policies as stochastic Markov control policies that depend on the k-hop neighbors of robot i:

$$\mathbf{a}_\tau^i \sim \boldsymbol{\pi}_{\boldsymbol{\theta}} \left(\mathbf{a}_\tau^i | \mathbf{s}^i_{\mathcal{N}_\tau^{i,k}} \right). \tag{6.3}$$

Here, $\mathbf{s}^i_{\mathcal{N}_\tau^{i,k}} = \{\mathbf{s}_\tau^j | j \in \mathcal{N}_\tau^{i,k}\}$ denotes the states of robot i and its k-hop neighbors, and $\boldsymbol{\theta}$ denotes the control policy parameters. The use of a stochastic Markov policy is not only motivated by the MDP model but also by the fact that distributed control policies are prone to uncertainty from the communication-perception modules, control goals, and interaction with the environment. We assume that the policy in Eq. (6.3) is the same for all the robots. The joint control policy of all the robots is denoted by

$$\boldsymbol{\Pi}_{\boldsymbol{\theta}} = [\boldsymbol{\pi}_{\boldsymbol{\theta}}^\top \left(\mathbf{a}_\tau^1 | \mathbf{s}^1_{\mathcal{N}_\tau^{1,k}} \right), \dots, \boldsymbol{\pi}_{\boldsymbol{\theta}}^\top \left(\mathbf{a}_\tau^n | \mathbf{s}^n_{\mathcal{N}_\tau^{n,k}} \right)]^\top.$$

Learning a distributed control policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ that solves a certain multi-robot task is equivalently posed as learning the parameters $\boldsymbol{\theta}$ such that $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ maximizes the expected sum of

rewards over time, i.e.,

$$\max_{\boldsymbol{\theta}} \; Q_{\boldsymbol{\Pi}_{\boldsymbol{\theta}}}(\mathbf{s}, \mathbf{a}) = \tag{6.4}$$
$$\max_{\boldsymbol{\theta}} \; \mathbb{E}_{\mathbf{s}_\tau \sim p} \mathbb{E}_{\mathbf{a}_\tau \sim \boldsymbol{\Pi}_{\boldsymbol{\theta}}(\cdot|\mathbf{s}_\tau)} \left[\sum_{\tau=0}^{\infty} \gamma^\tau r(\mathbf{s}_\tau, \mathbf{a}_\tau) | \mathbf{s}_0 = \mathbf{s}, \mathbf{a}_0 = \mathbf{a} \right].$$

In Eq. (6.4), the function $Q_{\boldsymbol{\Pi}_{\boldsymbol{\theta}}}(\mathbf{s}, \mathbf{a})$ is known as the action-value (Q) function associated with the policy $\boldsymbol{\Pi}_{\boldsymbol{\theta}}$ [Buşoniu et al., 2010].

We do not make any assumptions on the reward function or action-value function, such as a specific factorization [Qu et al., 2020, Qu et al., 2022]. The purpose of this work is to design a control policy that maximizes $Q_{\boldsymbol{\Pi}_{\boldsymbol{\theta}}}(\mathbf{s}, \mathbf{a})$ and enforces the distributed factorization expressed in Eq. (6.3).

6.3 Physics-informed multi-agent reinforcement learning

In this section, we present a novel physics-informed multi-agent reinforcement learning approach to find distributed control policies that solve multi-robot tasks as defined in Eq. (6.4), under the restrictions on the robot dynamics and available information defined in Eq. (6.3). Our formulation is done in continuous time, following the continuous-time definition of the robot dynamics in Eq. (6.1). Later, zero-order hold control is used to utilize a discrete-time version of the control policies in the reinforcement learning algorithm. To simplify the notation, we omit the time dependence of the states $\mathbf{x}$ and controls $\mathbf{u}$.

6.3.1 Port-Hamiltonian dynamics for multi-robot energy conservation

Port-Hamiltonian mechanics are a general yet interpretable modeling approach for learning and control. On the one hand, many physical networked systems can be described as a port-Hamiltonian system [Furieri et al., 2022b] using the same formulation and with a modular and distributed interpretation. Meanwhile, the port-Hamiltonian description allows to derive general energy-based controllers with closed-loop stability guarantees. Since robots are physical systems that satisfy Hamiltonian mechanics, we model each robot as a port-Hamiltonian system [Van Der Schaft et al., 2014]:

$$\dot{\mathbf{x}}^i = \left(\mathbf{J}^i(\mathbf{x}^i) - \mathbf{R}^i(\mathbf{x}^i) \right) \frac{\partial H^i(\mathbf{x}^i)}{\partial \mathbf{x}^i} + \mathbf{F}^i(\mathbf{x}^i)\mathbf{u}^i, \tag{6.5}$$

where the skew-symmetric interconnection matrix $\mathbf{J}^i(\mathbf{x}^i)$ represents energy exchange within a robot, the positive semi-definite dissipation matrix $\mathbf{R}^i(\mathbf{x}^i)$ represents energy dissipation, the Hamiltonian $H^i(\mathbf{x}^i)$ represents the total energy, and the matrix $\mathbf{F}^i(\mathbf{x}_i)$ is the input gain.

The interconnection of port-Hamiltonian systems leads to another port-Hamiltonian system [van der Schaft, 2004]. Therefore, if the control and state of each robot are considered as input and output energy ports, then, due to the modularity of port-Hamiltonian

dynamics, the multi-robot system with joint state $\mathbf{x}$ also follows port-Hamiltonian dynamics:

$$\dot{\mathbf{x}} = (\mathbf{J}(\mathbf{x}) - \mathbf{R}(\mathbf{x})) \frac{\partial H(\mathbf{x})}{\partial \mathbf{x}} + \mathbf{F}(\mathbf{x})\mathbf{u}, \tag{6.6}$$

where $\mathbf{J}(\mathbf{x})$, $\mathbf{R}(\mathbf{x})$, and $\mathbf{F}(\mathbf{x})$ are block-diagonal:

$$\begin{aligned} \mathbf{J}(\mathbf{x}) &= \operatorname{diag}\left(\mathbf{J}^1(\mathbf{x}^1), \dots, \mathbf{J}^n(\mathbf{x}^n)\right), \\ \mathbf{R}(\mathbf{x}) &= \operatorname{diag}\left(\mathbf{R}^1(\mathbf{x}^1), \dots, \mathbf{R}^n(\mathbf{x}^n)\right), \\ \mathbf{F}(\mathbf{x}) &= \operatorname{diag}\left(\mathbf{F}^1(\mathbf{x}^1), \dots, \mathbf{F}^n(\mathbf{x}^n)\right), \end{aligned} \tag{6.7}$$

and $H(\mathbf{x}) = \sum_{i=1}^{n} H^i(\mathbf{x}^i)$. It is noteworthy that the expression in Eq. (6.6) is control-affine and follows Eq. (6.1), with $\mathbf{f}(\mathbf{x}, \mathbf{u}) = \mathbf{h}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}$, $\mathbf{h}(\mathbf{x}) = (\mathbf{J}(\mathbf{x}) - \mathbf{R}(\mathbf{x})) \frac{\partial H(\mathbf{x})}{\partial \mathbf{x}}$, $\mathbf{g}(\mathbf{x}) = \mathbf{F}(\mathbf{x})$, and $\mathbf{L} = \mathbf{0}$ since we are considering a deterministic setting for now.

Without control, the trajectories of the open-loop system in (6.6) would follow the dynamics of the robots in the absence of interactions with the environment or other robots. The dynamics need to be controlled by a policy in order to accomplish the desired task. We propose to design a control policy $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x})$ and, then, obtain the desired policy in Eq. (6.3). Policy $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x})$ is designed using an Interconnection and Damping Assignment Passivity-Based Control (IDA-PBC) approach [Van Der Schaft et al., 2014], which injects additional energy in the system through the control input $\mathbf{u}$ to obtain closed-loop dynamics that achieve the desired task:

$$\dot{\mathbf{x}} = (\mathbf{J}_{\boldsymbol{\theta}}(\mathbf{x}) - \mathbf{R}_{\boldsymbol{\theta}}(\mathbf{x})) \frac{\partial H_{\boldsymbol{\theta}}(\mathbf{x})}{\partial \mathbf{x}}, \tag{6.8}$$

with Hamiltonian $H_{\boldsymbol{\theta}}(\mathbf{x})$, skew-symmetric interconnection matrix $\mathbf{J}_{\boldsymbol{\theta}}(\mathbf{x})$, and positive semi definite dissipation matrix $\mathbf{R}_{\boldsymbol{\theta}}(\mathbf{x})$, which depend on the control policy $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x})$. By matching the terms in (6.6) and (6.8), one obtains the joint policy:

$$\begin{aligned} &\mathbf{u} = \boldsymbol{\Pi}_{\boldsymbol{\theta}}(\mathbf{x}) = \\ &\mathbf{F}^{\dagger}(\mathbf{x}) \left((\mathbf{J}_{\boldsymbol{\theta}}(\mathbf{x}) - \mathbf{R}_{\boldsymbol{\theta}}(\mathbf{x})) \frac{\partial H_{\boldsymbol{\theta}}(\mathbf{x})}{\partial \mathbf{x}} - (\mathbf{J}(\mathbf{x}) - \mathbf{R}(\mathbf{x})) \frac{\partial H(\mathbf{x})}{\partial \mathbf{x}} \right), \end{aligned} \tag{6.9}$$

where $\mathbf{F}^{\dagger}(\mathbf{x}) = \left(\mathbf{F}^{\top}(\mathbf{x})\mathbf{F}(\mathbf{x})\right)^{-1} \mathbf{F}^{\top}(\mathbf{x})$ is the pseudo-inverse of $\mathbf{F}(\mathbf{x})$.

If the robots are fully-actuated, i.e., $\mathbf{F}(\mathbf{x})$ is full-rank, then the input $\mathbf{u}$ in (6.9) exactly transforms the open loop system in (6.6) to the closed-loop system in (6.8). For under-actuated systems, the transformation may not be exact [Blankenstein et al., 2002]. Being able to maximize $Q_{\boldsymbol{\Pi}_{\boldsymbol{\theta}}}$ is, hence, related to whether the robot configurations that solve the task are realizable by the class of control policies in (6.9). Even if goals of the task are not realizable, the policy parameters $\boldsymbol{\theta}$ may still be optimized to achieve a behavior as good as possible to solve the task.

Let $[\mathbf{J}_{\boldsymbol{\theta}}(\mathbf{x})]_{ij}$ and $[\mathbf{R}_{\boldsymbol{\theta}}(\mathbf{x})]_{ij}$ denote the $n_x \times n_x$ blocks with index (i, j), representing the energy exchange between robot i and j and the energy dissipation of robot i caused by robot j, respectively. Since the input gain $\mathbf{F}(\mathbf{x})$ in (6.7) is block-diagonal, the individual control policy of robot i is:

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}) = (\mathbf{F}^i)^{\dagger}(\mathbf{x}^i) \left(\sum_{j \in \mathcal{V}} \left([\mathbf{J}_{\boldsymbol{\theta}}(\mathbf{x})]_{ij} - [\mathbf{R}_{\boldsymbol{\theta}}(\mathbf{x})]_{ij} \right) \frac{\partial H_{\boldsymbol{\theta}}(\mathbf{x})}{\partial \mathbf{x}_j} - \left(\mathbf{J}^i(\mathbf{x}^i) - \mathbf{R}^i(\mathbf{x}^i) \right) \frac{\partial H^i(\mathbf{x})}{\partial \mathbf{x}^i} \right). \tag{6.10}$$

Note that the individual control policy $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x})$ in (6.10) does not necessarily respect the hops in the communication network as desired in (6.3) because it depends on the structure of $\mathbf{J}_{\boldsymbol{\theta}}(\mathbf{x})$, $\mathbf{R}_{\boldsymbol{\theta}}(\mathbf{x})$, and $H_{\boldsymbol{\theta}}(\mathbf{x})$. In Sec. 6.3.2, we impose conditions on these terms to ensure that they respect the communication topology and are skew-symmetric, positive semi-definite and positive respectively, as required for a valid port-Hamiltonian system and to find a policy $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ that only depends on the k-hop neighbors.

6.3.2 Self-attention parameterization to enforce communication patterns

We seek to learn distributed control policies that follow the structure of Eqs. (6.9)-(6.10) and (i) scale with the number of robots, (ii) handle time-varying communication and (iii) guarantee the port-Hamiltonian constraints. To do so, we first derive conditions on the port-Hamiltonian terms of the controller, $\mathbf{J}_{\boldsymbol{\theta}}(\mathbf{x})$, $\mathbf{R}_{\boldsymbol{\theta}}(\mathbf{x})$ and $H_{\boldsymbol{\theta}}(\mathbf{x})$, which are the terms to be learned from the reinforcement learning experience. Then, we develop a novel architecture based on self-attention to ensure that the learned control policies guarantee the desired requirements. We summarize the overall neural network architecture in Fig. 6.1.

To respect the robot team topology defined by the graph $\mathcal{G}$, we first impose $\mathbf{J}_{\boldsymbol{\theta}}(\mathbf{x})$ and $\mathbf{R}_{\boldsymbol{\theta}}(\mathbf{x})$ to be block-sparse,

$$[\mathbf{J}_{\boldsymbol{\theta}}(\mathbf{x})]_{ij} = [\mathbf{R}_{\boldsymbol{\theta}}(\mathbf{x})]_{ij} = \mathbf{0}, \quad \forall j \notin \mathcal{N}^{i,k}. \tag{6.11}$$

From the perspective of robot i, this means that the controller only considers information from its k-hop neighbors. Moreover, we require that the desired Hamiltonian factorizes over the k-hop neighborhoods:

$$H_{\boldsymbol{\theta}}(\mathbf{x}) = \sum_{i=0}^{n} H^i_{\boldsymbol{\theta}}(\mathbf{x}^i_{\mathcal{N}^{i,k}}), \tag{6.12}$$

with $\mathbf{x}^i_{\mathcal{N}^{i,k}} = \{\mathbf{x}^j | j \in \mathcal{N}^{i,k}\}$. The factorization in (6.12) ensures that each robot i can calculate

$$\frac{\partial H_{\boldsymbol{\theta}}(\mathbf{x})}{\partial \mathbf{x}^i} = \sum_{j \in \mathcal{N}^{i,k}} \frac{\partial H^j_{\boldsymbol{\theta}}(\mathbf{x}^j_{\mathcal{N}^{j,k}})}{\partial \mathbf{x}^i} \tag{6.13}$$

by gathering $\partial H^j_{\boldsymbol{\theta}}(\mathbf{x}^j_{\mathcal{N}^{j,k}})/\partial \mathbf{x}^i$ from its k-hop neighbors. Then, the control policy $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ of robot i becomes:

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}) = (\mathbf{F}^i(\mathbf{x}^i))^{\dagger}\Big(\sum_{j \in \mathcal{N}^{i,k}} ([\mathbf{J}_{\boldsymbol{\theta}}(\mathbf{x})]_{ij} - [\mathbf{R}_{\boldsymbol{\theta}}(\mathbf{x})]_{ij}) \frac{\partial H_{\boldsymbol{\theta}}(\mathbf{x})}{\partial \mathbf{x}^j} - \big(\mathbf{J}^i(\mathbf{x}^i) - \mathbf{R}^i(\mathbf{x}^i)\big) \frac{\partial H^i(\mathbf{x})}{\partial \mathbf{x}^i} \Big). \tag{6.14}$$

Imposing the requirements in (6.11)-(6.12) is a first step towards making the control policy in (6.14) distributed. Note that the terms $[\mathbf{J}_{\boldsymbol{\theta}}(\mathbf{x})]_{ij}$ and $[\mathbf{R}_{\boldsymbol{\theta}}(\mathbf{x})]_{ij}$ might still depend on the joint state vector $\mathbf{x}$ even though the sum runs over the k-hop neighbors in $\mathcal{N}^{i,k}$. Next, we discuss how to remove this dependence and achieve a similar factorization as (6.12).

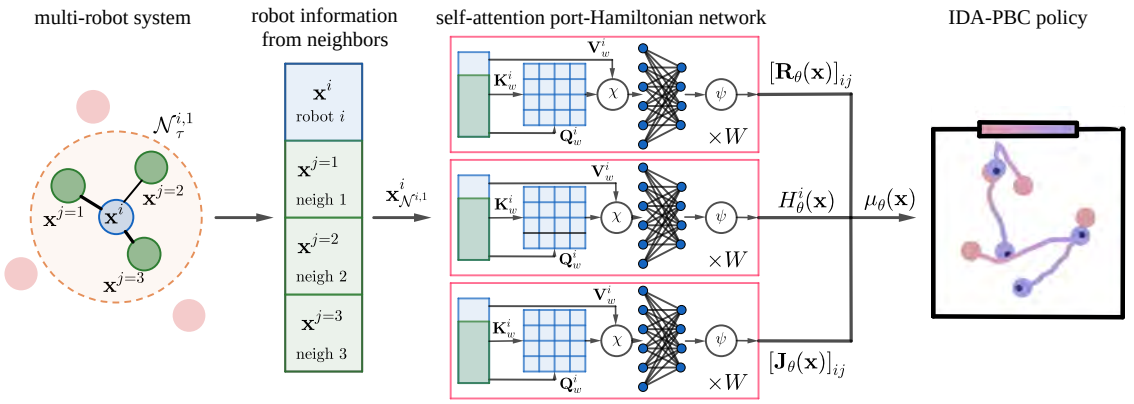


Figure 6.1: Physics-informed policy parameterization. At each instant, robot i receives state information from its neighbors, typically associated to a perception or communication radius. Then, three self-attention-based modules, each one associated to a component of the desired closed-loop port-Hamiltonian dynamics ($[\mathbf{R}_\theta(\mathbf{x})]_{ij}$, $[\mathbf{J}_\theta(\mathbf{x})]_{ij}$, $H^i_\theta(\mathbf{x})$), uses the information from the neighbors to compute the parameters of the IDA-PBC policy, which is then used to execute the desired control action.

First, we model $[\mathbf{J}_{\boldsymbol{\theta}}(\mathbf{x})]_{ij}, [\mathbf{R}_{\boldsymbol{\theta}}(\mathbf{x})]_{ij}$, and $H^i_{\boldsymbol{\theta}}(\mathbf{x})$ in Eq. (6.14) with the parameters $\boldsymbol{\theta}$ shared across the robots, so that the team can handle time-varying communication graphs. Specifically, we propose a novel architecture based on self-attention [Vaswani et al., 2017]. Self-attention layers extract the relationships among the inputs of a sequence by calculating the importance associated to each input using an attention map. The length of the sequences can vary as the number of parameters of the self-attention is constant with the number of inputs. Our key idea is to consider the self and neighboring states as the sequence, where each neighbor's state is an input. We now detail how to model each of the port-Hamiltonian terms.

To learn $[\mathbf{R}_{\boldsymbol{\theta}}(\mathbf{x})]_{ij}$, robot i will use, at instant t, the state $\mathbf{x}^j$ from all k-hop neighbors $j \in \mathcal{N}^{i,k}$. The sequence of states is given by $\mathbf{x}^i_{\mathcal{N}^{i,k}}$. The proposed architecture is composed by a sequence of layers, indexed using the subscript $w = 1, \ldots, W$. Following a self-attention mechanism, each layer computes the operations presented below:

$$
\begin{aligned}
&\mathbf{Q}^i_1 = \mathbf{A}^{\mathbf{R}}_1 \mathbf{x}^i_{\mathcal{N}^{i,k}},\ \mathbf{K}^i_1 = \mathbf{B}^{\mathbf{R}}_1 \mathbf{x}^i_{\mathcal{N}^{i,k}},\ \mathbf{V}^i_1 = \mathbf{C}^{\mathbf{R}}_1 \mathbf{x}^i_{\mathcal{N}^{i,k}} \\
&\mathbf{Y}^i_1 = \chi\left(\mathsf{softmax}\left(\frac{\beta(\mathbf{Q}^i_1)\beta((\mathbf{K}^i_1)^\top)}{\sqrt{|\mathcal{N}^{i,k}|}}\right)\beta(\mathbf{V}^i_1)\right), \\
&\mathbf{X}^i_1 = \psi(\mathbf{D}^{\mathbf{R}}_1 \mathbf{Y}^i_1), \\
&\vdots \\
&\mathbf{Q}^i_w = \mathbf{A}^{\mathbf{R}}_w \mathbf{X}^i_{w-1},\ \mathbf{K}^i_w = \mathbf{B}^{\mathbf{R}}_w \mathbf{X}^i_{w-1},\ \mathbf{V}^i_w = \mathbf{C}^{\mathbf{R}}_w \mathbf{X}^i_{w-1} \\
&\mathbf{Y}^i_w = \chi\left(\mathsf{softmax}\left(\frac{\beta(\mathbf{Q}^i_w)\beta((\mathbf{K}^i_w)^\top)}{\sqrt{|\mathcal{N}^{i,k}|}}\right)\beta(\mathbf{V}^i_w)\right), \\
&\mathbf{X}^i_w = \psi(\mathbf{D}^{\mathbf{R}}_w \mathbf{Y}^i_w),
\end{aligned}
\tag{6.15}
$$

where $\beta(\cdot)$, $\chi(\cdot)$, and $\psi(\cdot)$ are nonlinear activation functions. In the aforementioned operations, $\mathbf{A}^{\mathbf{R}}_w, \mathbf{B}^{\mathbf{R}}_w, \mathbf{C}^{\mathbf{R}}_w \in \mathbb{R}^{r_w \times h_w}$ and $\mathbf{D}^{\mathbf{R}}_w \in \mathbb{R}^{n_x \times r_w}$ for $w = 1, \ldots, W$ are matrices to be

learned and shared across robots; and $h_w, r_w, d_w > 0$, with $d_W = n_x$ and $h_1 = n_x$ for valid matrix multiplications. Matrices $\mathbf{A}_w^{\mathbf{R}}, \mathbf{B}_w^{\mathbf{R}}, \mathbf{C}_w^{\mathbf{R}}$ are of fixed size, so they are independent of the number of robots and neighbors. Thus, robot i can deal with time-varying neighbors. In particular, $\mathbf{A}_1^{\mathbf{R}}, \mathbf{B}_1^{\mathbf{R}}, \mathbf{C}_1^{\mathbf{R}}$ transforms the states to features encoded in the query $\mathbf{Q}_1^i$, key $\mathbf{K}_1^i$ and value $\mathbf{V}_1^i$ matrices. Then, $[\mathbf{R}_{\boldsymbol{\theta}}(\mathbf{x})]_{ij}$ is constructed as a weighted matrix that models the interactions of robot i with its k-hop neighbors, and a diagonal positive semi-definite matrix that accounts for the self-interactions:

$$\begin{aligned} \mathbf{Z}_{ij}^{\mathbf{R}} &= \mathsf{diag}(\mathbf{x}_W^{i,j}), \\ [\mathbf{R}_{\boldsymbol{\theta}}(\mathbf{x})]_{ij} &= -(\mathbf{Z}_{ij}^{\mathbf{R}} + \mathbf{Z}_{ji}^{\mathbf{R}}), \quad \forall j \in \mathcal{N}^{i,k}, \\ [\mathbf{R}_{\boldsymbol{\theta}}(\mathbf{x})]_{ii} &= \mathbf{Z}_{ii}^{\mathbf{R}} + \sum_{j \in \mathcal{N}^{i,k}} (\mathbf{Z}_{ij}^{\mathbf{R}} + \mathbf{Z}_{ji}^{\mathbf{R}}), \end{aligned} \tag{6.16}$$

where $\mathbf{x}_W^{i,j}$ is the column that corresponds to neighbor j in $\mathbf{X}_W^i$, and $\mathsf{diag}(\cdot)$ is the operator that reshapes the $n_x \times 1$ vector to a $n_x \times n_x$ diagonal matrix. If the nonlinear activation function $\psi(\cdot)$ is designed such that the elements of the output $\mathbf{X}_W^i$ are positive, then $\mathbf{Z}_{ij}^{\mathbf{R}}$ is a diagonal positive semi-definite matrix and $\mathbf{R}_{\boldsymbol{\theta}}(\mathbf{x})$ is a diagonally dominant matrix. This way, $\mathbf{R}_{\boldsymbol{\theta}}(\mathbf{x})$ is positive semi-definite by design. In fact, note the similarities between (6.16) and a weighted Laplacian matrix. More importantly, each element $[\mathbf{R}_{\boldsymbol{\theta}}(\mathbf{x})]_{ij}$ depends only on the information from robot i and its neighbor j, so the computation is distributed.

To learn $[\mathbf{J}_{\boldsymbol{\theta}}(\mathbf{x})]_{ij}$, we follow the same steps in (6.15), with parameters $\mathbf{A}_w^{\mathbf{J}}, \mathbf{B}_w^{\mathbf{J}}, \mathbf{C}_w^{\mathbf{J}}, \mathbf{D}_w^{\mathbf{J}}$ instead of $\mathbf{A}_w^{\mathbf{R}}, \mathbf{B}_w^{\mathbf{R}}, \mathbf{C}_w^{\mathbf{R}}, \mathbf{D}_w^{\mathbf{R}}$, to obtain encodings $\mathbf{Z}_{ij}^{\mathbf{J}}$ instead of $\mathbf{Z}_{ij}^{\mathbf{R}}$. Due to the reciprocal communication between robots i and j, we enforce the skew-symmetry of $\mathbf{J}_{\boldsymbol{\theta}}(\mathbf{x})$ by:

$$[\mathbf{J}_{\boldsymbol{\theta}}(\mathbf{x})]_{ij} = \mathbf{Z}_{ij}^{\mathbf{J}} - \mathbf{Z}_{ji}^{\mathbf{J}} \quad \forall j \in \mathcal{N}^{i,k}. \tag{6.17}$$

Since $[\mathbf{J}_{\boldsymbol{\theta}}(\mathbf{x})]_{ii} = \mathbf{0}$, the interconnection matrix is such that $\mathbf{J}_{\boldsymbol{\theta}}(\mathbf{x}) + \mathbf{J}_{\boldsymbol{\theta}}^{\top}(\mathbf{x}) = \mathbf{0}$ and, thus, is skew-symmetric by design. Again, each element $[\mathbf{J}_{\boldsymbol{\theta}}(\mathbf{x})]_{ij}$ depends only on the information from robot i and its neighbor j, so the computation is distributed.

Finally, to learn $H_{\boldsymbol{\theta}}^i(\mathbf{x}_{\mathcal{N}^{i,k}}^i)$, we represent it as follows:

$$H_{\boldsymbol{\theta}}^i(\mathbf{x}_{\mathcal{N}^{i,k}}^i) = \mathsf{vec}(\mathbf{x}_{\mathcal{N}^{i,k}}^i)^{\top} \mathbf{M}_{\boldsymbol{\theta}}^i(\mathbf{x}_{\mathcal{N}^{i,k}}^i) \mathsf{vec}(\mathbf{x}_{\mathcal{N}^{i,k}}^i) + U_{\boldsymbol{\theta}}^i(\mathbf{x}_{\mathcal{N}^{i,k}}^i). \tag{6.18}$$

The first term in the right-hand side of Eq. (6.18) is a kinetic-like energy function and the second terms is a potential energy function with

$$\mathbf{M}_{\boldsymbol{\theta}}^i(\mathbf{x}_{\mathcal{N}^{i,k}}^i) = \mathrm{diag}(\mathbf{1}^{\top} \mathbf{Z}_i^{\mathbf{M}}) \text{ and } U_{\boldsymbol{\theta}}^i(\mathbf{x}_{\mathcal{N}^{i,k}}^i) = \mathbf{1}^{\top} \mathbf{Z}_i^U \mathbf{1}.$$

The encodings $\mathbf{Z}_i^{\mathbf{M}}$ and $\mathbf{Z}_i^U$ are calculated using the same steps in (6.15), with parameters $\mathbf{A}_w^{\mathbf{M}}$, $\mathbf{B}_w^{\mathbf{M}}$, $\mathbf{C}_w^{\mathbf{M}}$, $\mathbf{D}_w^{\mathbf{M}}$ and $\mathbf{A}_w^U$, $\mathbf{B}_w^U$, $\mathbf{C}_w^U$, $\mathbf{D}_w^U$, respectively. Once we have $H_{\boldsymbol{\theta}}^i(\mathbf{x}_{\mathcal{N}^{i,k}}^i)$, we obtain $\partial H_{\boldsymbol{\theta}}^j(\mathbf{x}_{\mathcal{N}^{j,k}}^j)/\partial \mathbf{x}^i$ from all the neighboring robots and compute Eq. (6.13). Therefore, since all the operations are only dependent on the available information in robot i and its neighbor j, the computations involving the Hamiltonian function are distributed by design.

To deploy the control policy (6.14), we design a message $\mathbf{m}_{\tau}^{i,j}$, encoding information that robot i needs from robot j at time τ to calculate $[\mathbf{J}_{\boldsymbol{\theta}}(\mathbf{x})]_{ij}$, $[\mathbf{R}_{\boldsymbol{\theta}}(\mathbf{x})]_{ij}$, and $H_{\boldsymbol{\theta}}^i(\mathbf{x})$. When there is no communication among robots, i.e., $k = 0$, no message is needed. For $k \geq 1$, robot i uses the following communication protocol:

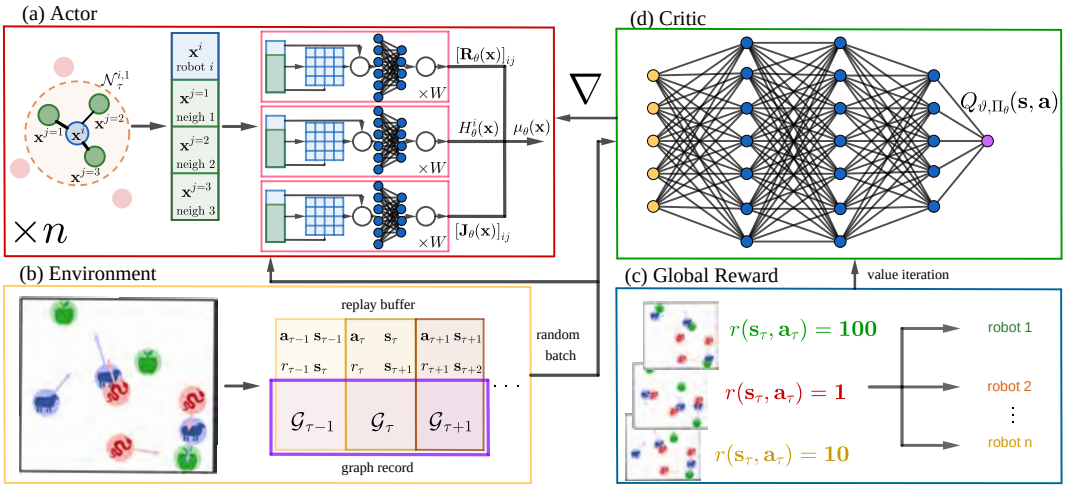


Figure 6.2: Overview of our physics-informed soft actor-critic multi-agent reinforcement learning approach. The main differences with respect of other actor-critic methods are the following: (a) the actor is the multi-robot network modeled as a port-Hamiltonian system and with a policy given by a self-attention-based IDA-PBC; (b) the replay buffer not only stores action, states and reward, but also the graph structure of the multi-robot network to enforce the desired distributed structure; (c) the reward is global because the actor is the whole multi-robot team and the physics-informed parameterization already conditions the policy on the graph structure of the team; (d) the output of the critic is shared across robots and the policy parameters are the same for all robots, so for the same critic gradient step, n policy gradient steps are taken.

1. Robot i receives messages $\mathbf{m}^{i,j}_{\tau,1} = \mathbf{x}_j$ from its k-hop neighbors in $\mathcal{N}^{i,k}$. Then, robot i computes $\mathbf{Z}^{\mathbf{R}}_{ij}$, $\mathbf{Z}^{\mathbf{J}}_{ij}$, $H^i_{\boldsymbol{\theta}}$, and $\partial H^i_{\boldsymbol{\theta}}/\partial \mathbf{x}_j$.

2. Robot i receives messages $\mathbf{m}^{i,j}_{\tau,2} = \{\partial H^j_{\boldsymbol{\theta}}/\partial \mathbf{x}_i, \mathbf{Z}^{\mathbf{R}}_{ji}, \mathbf{Z}^{\mathbf{J}}_{ij}\}$ from its k-hop neighbors in $\mathcal{N}^{i,k}$ and computes $\partial H_{\boldsymbol{\theta}}/\partial \mathbf{x}_i$, $[\mathbf{J}_{\boldsymbol{\theta}}]_{ij}, [\mathbf{R}_{\boldsymbol{\theta}}]_{ij}$.

3. Robot i receives messages $\mathbf{m}^{i,j}_{\tau,3} = \partial H_{\boldsymbol{\theta}}/\partial \mathbf{x}_j$ from its k-hop neighbors in $\mathcal{N}^{i,k}$ and computes the control input $\mathbf{a}_i$.

In summary, each robot i receives a message $\mathbf{m}^{i,j}_{\tau} = [\mathbf{m}^{i,j}_{\tau,1}, \mathbf{m}^{i,j}_{\tau,2}, \mathbf{m}^{i,j}_{\tau,3}]$ in 3 communication rounds from its neighboring robot j. We assume negligible delays between communication rounds. If the delay is large, Wang et al. [Wang et al., 2022] suggest to learn a function that predicts quantities such as $\partial H_{\boldsymbol{\theta}}(\mathbf{x})/\partial \mathbf{x}_j$, $\mathbf{Z}^{\mathbf{J}}_{ji}$, $\mathbf{Z}^{\mathbf{R}}_{ji}$, leading to one communication round. We leave this for future work.

6.4 Physics-informed multi-robot soft actor-critic

The previous section describes how to parameterize the distributed control policies to be learned. In this section, we present a soft actor-critic algorithm [Haarnoja et al., 2018b] to train the self-attention port-Hamiltonian neural network, detailing the main features that allow the integration of our physics-informed policy representation. The key components are presented in Fig. 6.2, namely: (a) actor, (b) environment, (c) reward, and (d) critic.

6.4.1 The actor

First, we model the actor with the port-Hamiltonian system detailed in Sec. 6.3.1 and we use the IDA-PBC policy in Eq. (6.14) parameterized by the self-attention architecture proposed in Sec. 6.3.2. The known robot dynamics are typically provided by the simulation environments or the hardware specifications of the robots [Bettini et al., 2022]. Nevertheless, two aspects must be adapted to match the soft actor-critic formulation: (i) the MDP formulation and the stochastic Markov policy are in discrete time while the robot dynamics and the IDA-PBC are in continuous-time, and (ii) the actions in the soft actor-critic method are stochastic while the port-Hamiltonian formulation is deterministic.

To address the continuous versus discrete time mismatch, we use the zero-order hold control approach described in Sec. 6.2. More specifically, $\mathbf{u}(t) = \mathbf{a}_\tau = \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}(t_\tau))$ for $t \in [t_\tau, t_{\tau+1})$. This means that we can write the control policy as dependent on the discrete-time state $\mathbf{s}_\tau$ rather than the continuous-time state $\mathbf{x}(t_\tau)$, leading to $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{s}_\tau)$ with the same expression in Eq. (6.14).

To address stochasticity and maintain the desired distributed structure already derived in Sec. 6.3.1 and 6.3.2, we model the distributed control policies as (squashed) Gaussian distributions, whose mean is given by the IDA-PBC controller in (6.14), therefore respecting the distributed policy factorization. Meanwhile, the variance of the policy distribution is provided by a neural network that is learned during training. Overall, we obtain a distributed and stochastic Markov control policy of the form:

$$\mathbf{a}^i_\tau \sim \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{a}^i_\tau | \mathbf{s}^{i,k}_{\mathcal{N}^i_\tau}) = \tanh(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{s}^{i,k}_{\mathcal{N}^i_\tau}) + \boldsymbol{\sigma}_{\varrho}(\mathbf{a}^i_\tau, \mathbf{s}^{i,k}_{\mathcal{N}^i_\tau})\xi), \tag{6.19}$$

where $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{s}^{i,k}_{\mathcal{N}^i_\tau})$ is the IDA-PBC policy parameterized by self-attention neural networks under the restrictions imposed by the networked structure of the robot team.

On the other hand, $\boldsymbol{\sigma}_{\varrho}(\mathbf{a}^i_\tau, \mathbf{s}^{i,k}_{\mathcal{N}^i_\tau})$ is a vector of standard deviations given by a neural network that approximates the variance of the squashed Gaussian distribution with parameters ϱ. Besides, $\xi \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, so the control policy is a Gaussian distribution with mean $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{s}^{i,k}_{\mathcal{N}^i_\tau})$ and diagonal covariance matrix whose diagonal is equal to $(\boldsymbol{\sigma}_{\varrho}(\mathbf{a}^i_\tau, \mathbf{s}^{i,k}_{\mathcal{N}^i_\tau}))^2$. The control is constrained to $\mathbf{a}^i_\tau \in [-1, 1]^{n_u}$ by means of a tanh function [Haarnoja et al., 2018a, Haarnoja et al., 2018c, Haarnoja et al., 2018b], leading to a squashed Gaussian policy. In practice, the control input can be constrained to $\mathbf{a}^i_\tau \in [a_{\min}, a_{\max}]^{n_u}$ with $-\infty < a_{\min} \leq a_{\max} < \infty$ by scaling the output of the tanh function. It is also important to remark that $\boldsymbol{\sigma}_{\varrho}(\mathbf{a}^i_\tau, \mathbf{s}^{i,k}_{\mathcal{N}^i_\tau})$ only depends on the available information at each robot, such that the desired distributed factorization of the control policy is preserved.

During training, the control inputs are sampled from the Gaussian distribution, as stated in Eq. (6.3). After training, the mean control input is chosen, i.e., $\mathbf{a}^i_\tau = \tanh(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{s}^{i,k}_{\mathcal{N}^i_\tau}))$. In the case that an additional layer of robust, adaptive or active control is desired, then $\boldsymbol{\sigma}_{\varrho}(\mathbf{a}^i_\tau, \mathbf{s}^{i,k}_{\mathcal{N}^i_\tau})$ can be used as a proxy of the uncertainty in the control policy, since it only depends on the neighboring information at robot i. The design of $\boldsymbol{\sigma}_{\varrho}(\mathbf{a}^i_\tau, \mathbf{s}^{i,k}_{\mathcal{N}^i_\tau})$ is free to choose, but in this work we opt for the same architecture in Eq. (6.15).

6.4.2 The environment

The second modification over the soft actor-critic algorithm is in the collection of experiences from the environment. The main difference with other multi-agent reinforcement learning works is that, to build the replay buffer that stores the trial and error experience used for training the control policies, we take into account all the interactions among the robots. Differently from centralized-training decentralized-deployment approaches, where the experience of each agent is recorded independently of the other agents, we record for each experience all the robot states, actions, and interaction graph together. Thanks to that, we keep track of the correlation among robots during training. This also allows us to condition the trained control policies on the available information at each robot, therefore only providing the information robots will have access during deployment.

6.4.3 The reward

The reward function can be shared across robots and include global terms because the actor is the whole multi-robot network. This is important because, from the perspective of the soft actor-critic algorithm during training, the whole robot network is a single centralized agent. Other state-of-the-art multi-agent reinforcement learning approaches (see Sec. 6.1) require a factorization of the reward function because each agent is considered as an isolated learning unit which does not exploit neighboring information during the execution of the control policy. In contrast, in our approach the shared distributed control policy is simultaneously learned at all the robots used in training, and since the experience records the underlying communication graph, we can seamlessly associate global rewards with distributed cooperative control policies through communication. It is worth to note that, as in any classical soft actor-critic approach, the maximization objective is changed to include an entropy term $\mathcal{H}(\boldsymbol{\Pi}_{\boldsymbol{\theta}}(\mathbf{a}|\mathbf{s}))$ that measures the entropy of the control policy. This entropy term, weighted by a temperature parameter $\alpha > 0$, trades off exploration (high value) and exploitation (low value). The value of α changes over time according to a gradient descend law which manages automatically the exploration/exploitation dichotomy.. The details of this automatic temperature adjustment rule can be found in [Haarnoja et al., 2018b].

6.4.4 The critic

Finally, regarding the critic, its main purpose is to learn the action-value function of the environment, encoded in $Q_{\boldsymbol{\Pi}_{\boldsymbol{\theta}}}(\mathbf{s}, \mathbf{a})$. This approximation steers the training of the control policy towards the maximization objective. Nevertheless and importantly, the critic is only used during training. Thus, it is not necessary to design it to be distributed and an existing centralized neural network architecture can be used. In this work, we opt for a multi-layer perceptron with parameters $\boldsymbol{\vartheta}$ to learn $Q_{\boldsymbol{\vartheta},\boldsymbol{\Pi}_{\boldsymbol{\theta}}}(\mathbf{s}, \mathbf{a})$. As with the reward, since, from the perspective of the soft actor-critic algorithm, the single agent is the whole robot network described by the port-Hamiltonian dynamics, a single centralized critic can learn the appropriate action-value function to condition the training of the distributed control

policy.

A shared centralized action value function implies that each optimization step considers, simultaneously, all the policies gathered in the joint policy $\boldsymbol{\Pi}_{\boldsymbol{\theta}}$. Since the policies are homogeneous, each optimization step is simultaneously updating the policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ n times. This is achieved without any particular specification of the gradients nor factorization of the action value function. This is one of the key properties and advantages of our proposed soft actor-critic algorithm compared to other state-of-the-art algorithms. Typical multi-agent reinforcement learning approaches, by factorizing the policy and/or the value functions, arrive to a different set of policy parameters, one per agent used in training. In cooperative tasks where the robots share the same goal, there should be a single policy that resolves the task independently on the configuration of the robot. In our case, by appropriately integrating the modular port-Hamiltonian description of the multi-robot system with the soft actor-critic algorithm, we train a single distributed control policy which takes into account the current available information at each robot. Using the soft actor-critic update rules [Haarnoja et al., 2018c, Haarnoja et al., 2018b, Haarnoja et al., 2018a], the gradients with respect to $\boldsymbol{\theta}$ are, simultaneously for all the robots, already conditioned on the multi-robot communication topology.

In conclusion, in contrast with other solutions, ours explicitly considers the exchanges of information among robots by modeling the multi-robot system as a graph. This allows to use a reinforcement learning algorithm for single-robot problems, where the single agent is the multi-robot network. From the perspective of the reinforcement learning algorithm, the actor is centralized. However, by means of a physics-informed self-attention parameterization of the dynamics and control of the robots, the learned policies are distributed by design, modeling the robot team as a modular port-Hamiltonian system.

6.5 Results

To assess our physics-informed multi-agent reinforcement learning approach, we present seven multi-robot scenarios. The first three are extracted from the VMAS simulator [Bettini et al., 2022]. The next three are adaptations from the MPE simulator [Mordatch and Abbeel, 2018, Lowe et al., 2017] that can be found in [Wang et al., 2022]. The last scenario comes from the Multi-Agent MuJoCo benchmark [Peng et al., 2021]. The overview of the scenarios are detailed in the following:

a. **Reverse transport**: the robots are randomly spawned inside a box that they must push towards a desired landmark in the arena. The initial position of the box and the landmark is random. Compared to [Bettini et al., 2022], we decrease the mass of the box to 1 kg to ensure that the box can be moved even with a small number of robots. Also, when the box is in the landmark, the reward is set to 1.

b. **Sampling**: the robot team is randomly spawned in an arena with an underlying Gaussian density function composed of 3 modes. The field is discretized to a grid. Robots must collect samples of the field such that once a robot visits a cell its sample

is collected without replacement and given as reward to the team. Robots use a LiDAR to sense each other, and they observe the samples in the 3×3 grid around it.

c. **Navigation**: each robot has a landmark to reach. The initial position of the robots and landmarks are randomly spawned in a 2×2m square arena. Robots must navigate to reach their corresponding landmarks while avoiding collisions with other robots. Compared to [Bettini et al., 2022], we encourage collision avoidance by changing the collision penalty from -1 to -5. Besides, each robot only observes its desired landmark instead of all the landmarks.

d. **Food collection**: it is a version of the simple spread scenario [Lowe et al., 2017] where a team of robots and food landmarks is randomly spawned in an arena. There are as many landmarks as robots. Robots must cooperate to cover as many food landmarks as possible. Each time a robot covers a new landmark, the whole team is rewarded.

e. **Grassland**: A team of robots must collect food resources that are randomly spawned in an arena, while evading multiple predators. There are as many robots as predators, and the former move twice faster than the latter. The predator team is positively rewarded when some member captures a robot, whereas the robot team is negatively rewarded and the robot is deactivated. On the other hand, the robots receive a positive reward if they reach a food landmark, which is then spawned again in a new position.

f. **Adversarial**: two robotic teams compete for the same food landmarks. Both teams have the same number of robots. When a member of a team reaches a food landmark, the landmark is randomly spawned and the team is positively rewarded. When two members of a team collide with one member of the other team, then the first team is positively rewarded, while the second team is penalized and the robot deactivated.

g. **6x1-Half Cheetah**: a robotic cheetah with two legs and 6 joints, where each joint is a different agent. The multi-agent team is distributed in the sense that each agent can communicate only with the adjacent joints, enforced by a ring graph topology. The task is to make the robot run as fast as possible. The reward is global and is composed by a first term that favours forward movements and a second term that penalizes too large actions.

The first three scenarios are used for ablation studies, where the main goal is to evaluate how our physics-informed policy parameterization improves upon existing standard policy parameterizations. The second three are used to compare our method with other state-of-the-art multi-agent reinforcement learning approaches that are not physics-informed. The last scenario is used to validate our approach in a realistic robotic platform.

All robots have a communication radius $r_{comm} > 0$ which allows to exchange information with 1-hop neighbors. The communication radius and other hyperparameters of the soft actor-critic algorithm are specified in Section 6.9. Each robot observes its position, velocity, position and velocity of the landmarks or objects of interest (e.g., the box in reverse transport). The particularities of the observation space of each robot can be

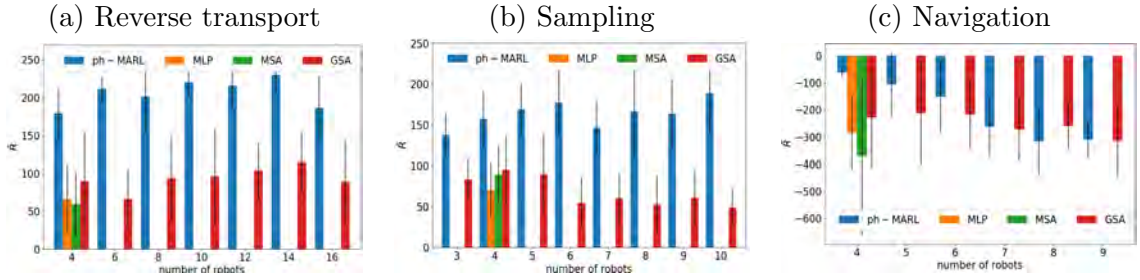


Figure 6.3: Comparison of the performance of the ablated control policies when we scale the number of robots in deployment. In all the scenarios, our proposed combination of a port-Hamiltonian modeling and self-attention-based neural networks achieves the best cumulative reward without further training the control policy. Each bar displays the mean and standard deviation of $\bar{R}$ over 10 evaluation episodes.

Table 6.1: Averaged training convergence cumulative reward for the four ablated methods. In all the cases, $n = 4$ robots are used.

Method	Mean and std of $\bar{R}$ over 10 evaluation episodes		
	Reverse transport	Sampling	Navigation
pH-MARL	$\mathbf{213 \pm 21}$	$\mathbf{161 \pm 41}$	$\mathbf{-53 \pm 101}$
MLP	64 ± 38	73 ± 25	-280 ± 98
MSA	57 ± 43	82 ± 32	-353 ± 99
GSA	90 ± 49	89 ± 38	-204 ± 87

found in [Bettini et al., 2022, Lowe et al., 2017, Wang et al., 2022]. In some scenarios, the robot observation vector changes its dimension depending on the number of robots. Specifically, in the food collection, grassland and adversarial scenarios the number of landmarks changes with the number of robots. Therefore, to accommodate the proposed self-attention port-Hamiltonian neural network with an observation vector which may change its size, we use an additional neural network to pre-process the observation vector. In particular, we concatenate a self-attention layer with a dense layer which receives, as input, the food landmarks' positions and outputs a feature vector of constant dimension that is used as part of the state vector to build $\mathbf{S}_t^i$. The details can be found in Section 6.8.

6.5.1 Ablation results

We conduct ablation studies using the reverse transport, sampling and navigation scenarios. The goal is to evaluate how our proposed physics-informed policy parameterization improves upon existing standard policy parameterizations. We compare the proposed self-attention port-Hamiltonian neural network (pH-MARL) with a classical multi-layer perceptron (MLP), a modular self-attention-based neural network (MSA) and a graph- and attention-based neural network (GSA). The implementation details can be found in Section 6.8.

The MLP network is an unstructured neural network that receives, as input, the current

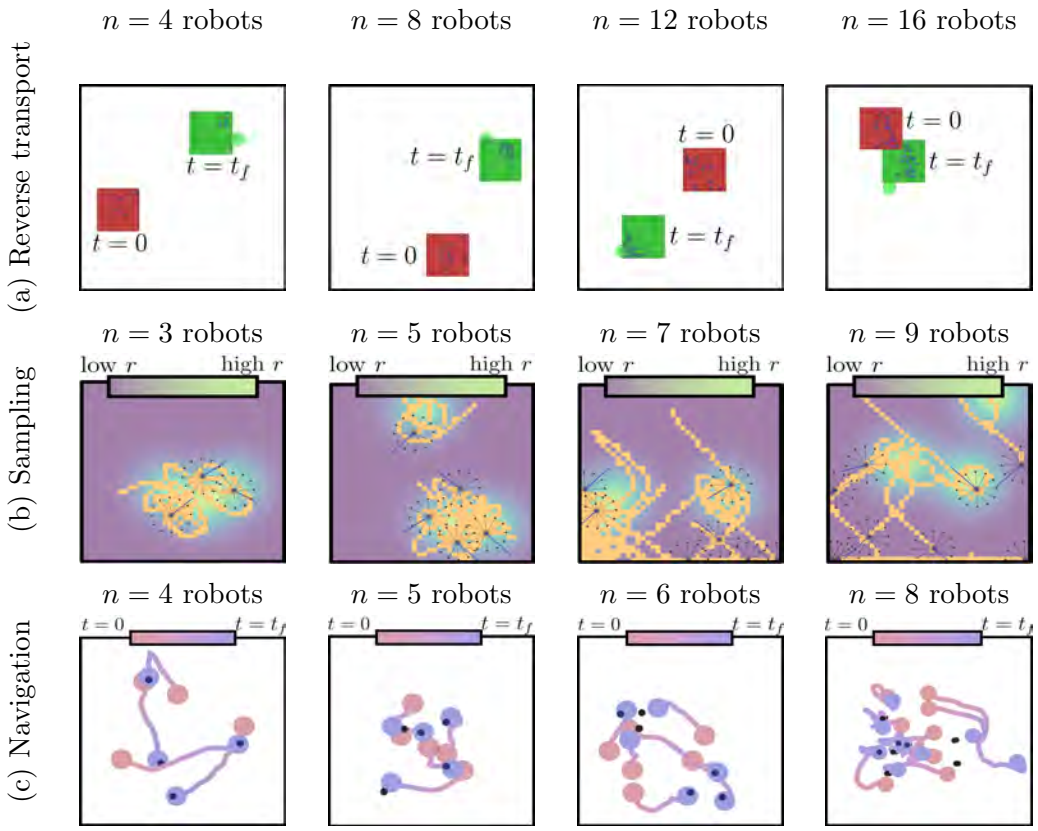


Figure 6.4: Examples of multi-robot scenarios for different initial conditions and number of robots. In the sampling scenario, LiDAR measurements are depicted as points forming a star pattern centered at the position of the robot, and regions already explored are painted in orange. It is interesting to see that some aspects of the environment do not scale with the team size, e.g., the size and weight of the box in the reverse transport scenario, the number of hot spots in the sampling scenario or the size of the arena in the navigation scenario.

states and actions of the robots and outputs the action vector. The MSA network replaces the dense layers of the MLP network with an architecture based on self-attention, but has the same input and output. The GSA network is similar to the MSA network but has the adjacency matrix as additional input. The three neural networks constitute a sequential improvement from a standard neural network to our port-Hamiltonian formulation. It is important to note that, with the current formulation, it is not possible to try a port-Hamiltonian neural network which is not based on self-attention because each element of the processed input sequence is employed to compute the port-Hamiltonian terms, that are also a sequence spanned from $i \in \mathcal{N}_t^{i,k}$. This is key for the scalability of the policy and is a feature that is not provided by other architectures based on, e.g., multi-layer perceptrons or convolutions. Finally, we employ the same architecture used in pH-MARL to learn the standard deviation for all the options, along with the same critic and soft actor-critic algorithm (Section 6.8).

Table 6.1 shows the averaged cumulative reward $\bar{R} = \frac{1}{n}R = \frac{1}{n}\sum_{\tau=0}^{\tau_{\max}-1} r(\mathbf{s}_\tau, \mathbf{a}_\tau)$ after convergence of the training, where $\tau_{\max}$ is the maximum number of steps per episode. For the three scenarios and the four neural network architectures, we use $n = 4$ robots. The table reports the mean of $\bar{R}$ over 10 evaluation episodes. In all the cases, pH-MARL

surpasses the other three architectures, with a particularly significant difference in the reverse transport and navigation scenarios. Given the same number of parameters in the four architectures, the use of a physics-informed formulation of the neural network leads to a structured learning with efficient sampling in training. The difference in the reverse transport scenario is due to the fact that the other networks sometimes fail in reaching the landmark, despite arriving to a very close position. In the sampling scenario, pH-MARL is the fastest in inspecting the environment, therefore covering more informative cells. It is followed, in decreasing order, by GSA, MSA and MLP, which is reasonable since the three neural networks are in decreasing order of architecture complexity in terms of neural network modules and input information. The navigation scenario is the simplest among the three. Nevertheless, pH-MARL is still the best because it better learns the collision avoidance constraint.

To assess the scalability of the different architectures, in Fig. 6.3 each trained neural network is evaluated with a different number of robots, showing the mean and standard deviation of $\bar{R}$ over 10 episodes. The first conclusion is that pH-MARL and GSA scale well with the number of robots, achieving the same cumulative reward per robot for all team sizes. In contrast, MSA and MLP do not scale because their architectures considers directly the global observation and action vectors, so their modules are not ready to process an input vector of different dimension. On the other hand, pH-MARL achieves much better performance than the other networks in all the scenarios and team sizes, thus confirming the importance of using physics-informed priors to ease the learning. The single exception is in the navigation example, where the performance deteriorates when the number of robots increases.

Fig. 6.4 provides some qualitative examples for the three scenarios and helps to understand some limitations of our approach. Beginning with the navigation scenario, we can see that the size of the arena does not scale with the number of robots, it is always 2×2m. Hence, taking into account that the robots have a radius of 15cm, the space is very small to navigate towards the landmarks without colliding. According to the reward, the robots search for safe motions, so they prefer to avoid collisions rather than moving to their landmarks (Fig. 6.4, navigation with 8 robots). There are other features that do not scale in the reverse transport and sampling scenarios: size of the box, weight of the box, size of the arena and number of Gaussian modes. Nonetheless, pH-MARL is capable of generalizing, scaling and circumventing these issues and achieve the desired tasks. In the reverse transport scenario, even with 16 robots that are highly packed inside the box, the robots manage to push the box and avoid erratic movements caused by the low weight of the box. In the sampling scenario, robots learn to spread and coordinate to cover more cells when there are more robots. Interestingly, when there are a lot of robots ($n = 9$), some robots move to the corners because they annoy the others by increasing the potential number of collisions.

Summing up, the combination of a port-Hamiltonian formulation and self-attention mechanisms leads to superior performance and the desired scalability with the number of robots, learning control policies that are distributed by design and fully exploit the graph structure of the multi-robot system.

6.5.2 Comparative results

We compare the performance of our proposed physics-informed multi-agent reinforcement learning approach with other state-of-the-art multi-agent reinforcement learning approaches that are not physics-informed. In particular, we compare with: Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [Lowe et al., 2017], Mean Field Actor Critic (MFAC) [Yang et al., 2018], Evolutionary Population Curriculum (EPC) [Long et al., 2019], Distributed multi-Agent Reinforcement Learning with One-hop Neighbors (DARL1N) [Wang et al., 2022] and Multi-Agent Proximal Policy Optimization (MAPPO) [Yu et al., 2022]. The comparison results of the first four methods are extracted from [Wang et al., 2022].

We use the food collection, grassland and adversarial scenarios for the comparison. Following the same procedure in [Wang et al., 2022], for the scenarios with an adversarial team (grassland and adversarial), we first use MADDPG to learn the control policies of both teams. Then, the adversarial team control policy is frozen and the policy of the other team is learned using any of the aforementioned approaches.

Next, we assess the scalability of our proposed approach in this second series of scenarios. We train the other state-of-the-art approaches for each specific number of robots, ranging from 3 to 48 robots. In contrast, we train pH-MARL $n = 4$ robots and evaluate the trained control policies with the other number of robots, without further training. Fig. 6.5 demonstrates that pH-MARL achieves better or similar results than the other approaches without further training the control policies. When the team size is close to the one used during training ($n = 3, 6, 12$) pH-MARL surpasses the other methods in the three scenarios, proving the accuracy of combining a physics-informed description of the multi-robot team with self-attention mechanisms. In this sense, MAPPO in the grassland scenario with $n = 6$ robots is the only case that obtains better results than pH-MARL. When the team size is much greater than the one used during training ($24, 48$), pH-MARL outperforms (MADDPG, MFAC) or achieves similar results (EPC, DARL1N, MAPPO) as the state-of-the-art. Notably, pH-MARL outperforms all the state-of-the-art methods in the adversarial scenario even when $n = 48$, which is significant considering that they are all trained with $n = 48$ robots except pH-MARL, which is trained with $n = 4$ robots. Regarding MAPPO, despite its strong performance in some cases (e.g., food collection with $n = 3$ robots, grassland with $n = 6$ robots or grassland with $n = 48$ robots), it generally exhibits poor results when the number of robots is larger than 10 robots. Besides, MAPPO is unable to learn successful policies that avoid collision and find the food resources for any of the teams in the adversarial scenario.

The reason why pH-MARL exhibits a slight drop in performance when $n = 24, 48$ robots is the same discussed in the ablation experiments. There are some features of the scenarios, like the size of the arena, that do not scale with the number of robots. Robots are not trained to move in configurations that far from those experienced during training. How to design the control policy to be invariant to environmental changes is part of the future work.

Regarding the computational cost, it is worth-noting that, given the same scenario,

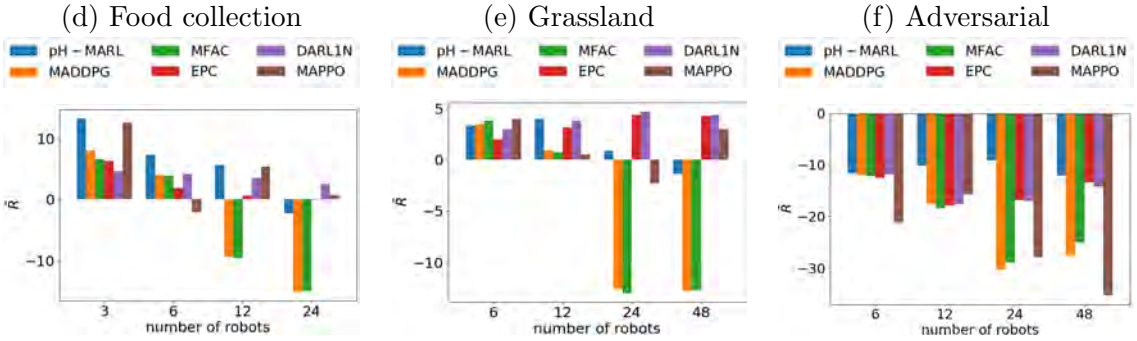


Figure 6.5: Comparison of our proposed physics-informed multi-agent reinforcement learning approach with other state-of-the-art approaches. pH-MARL is only trained with $n = 4$ robots and deployed with different team sizes, while the state-of-the-art control policies are trained for each specific number of robots.

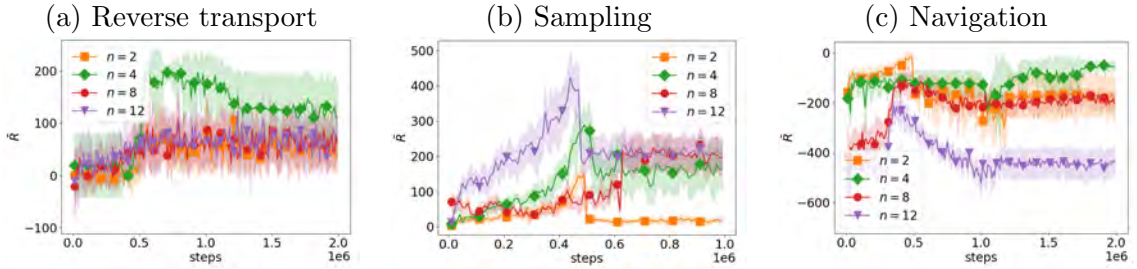


Figure 6.6: Averaged cumulative reward obtained by pH-MARL for different numbers of robots at training.

optimizer and replay buffer hyperparameters, the main differences among methods come from the parameterization of the actor. In this sense, our approach involves greater training times *per sample and robot* compared to other non-physics-informed methods. Specifically, the increase in computational burden comes from the differentiation of the Hamiltonian $\partial H_{\boldsymbol{\theta}}(\mathbf{x})/\partial \mathbf{x}_j$ at each time step. However, by restricting the space of admissible policies, we have better sample efficiency during training compared to non-physics-informed parameterizations. Therefore, the increase of computational burden per sample and robot is compensated by the sample efficiency of the method, leading to similar overall computational times in robot tasks during training. Besides, as it has been observed in the scalability results, our approach needs a very small number of robots to reproduce or even outperform the other methods when they are trained with larger numbers of robots, saving additional computation. Importantly, the computational cost during deployment scales with the number of neighbors rather than the network size, therefore preserving the desirable properties coming from a distributed multi-robot setting.

6.5.3 Sensitivity analysis

Regarding the evaluation of the training scalability of our approach, we have conducted a series of experiments using the same scenarios of Section 6.5.1 and Section 6.5.2. Given those scenarios, we have trained our ph-MARL with an increasing number of robots, namely, $n = \{2, 4, 8, 12\}$ (see Fig. 6.6). The first interesting conclusion is that the increase in number of robots does not necessarily lead to better performance because the interactions

among robots always happen at the robot neighborhood level. In the reverse transport scenario, more robots inside the cage implies a greater number of robot contacts with the cage and, therefore, a drop in averaged cumulative reward. In the sampling scenario, a greater number of robots leads to an improvement in performance because, for the same time horizon, robots can explore the arena faster and localize the informative regions faster. In the case of the navigation scenario, due to space constraints, an increase in the number of robots during training leads to a significant drop in averaged cumulative reward because the space available to each robot is smaller, so more collisions happen (specially with $n = 12$). In all the scenarios, $n = 2$ is a too small multi-robot team size to capture the richness of distributed multi-robot interactions, so the training process is unable to learn a successful policy in terms of averaged cumulative reward.

Therefore, in general, the optimal number of training robots depends on constraints of the scenario and the task. Nonetheless, as a rule of thumb, increasing the number of training robots towards a large multi-robot team only leads to greater sample complexity, whereas decreasing the number of training robots does not allow to learn sufficiently varied behaviors to scale with the number of robots. Henceforth, we conclude that our approach is best suited for training with a small number of training robots (relative to the task/scenario) but still sufficiently large to capture all possible neighborhood interactions.

From a behavioral analysis perspective, the results in Fig. 6.6 suggest that the learned distributed multi-robot policies benefit from small neighborhoods, irrespective of the network size. In scenarios with a fixed arena size, an increase in the number of robots indirectly leads to an increase in neighborhood sizes, with a subsequent drop in averaged cumulative reward both in training and evaluation (see Fig. 6.5). The appropriate number of training robots depends on the scenario, e.g., in the reverse transport scenario the best performance is achieved with $n = 4$ while in the sampling scenario this is obtained with $n = 8$. In any case, as concluded above, our approach requires a sufficiently large team size to capture a variety of local interactions.

6.5.4 Validation on a realistic robot platform

We further validate pH-MARL in a realistic robot setting using MuJoCo [Todorov et al., 2012]. MuJoCo is a general purpose physics engine that replicates multi-joint dynamics with contacts, allowing for fast yet realistic physical behaviors. Specifically, we use the Multi-agent MuJoCo benchmark [Peng et al., 2021], where the task is to learn coordination policies for multi-joint robots that make them walk along the environment. Our approach has been validated using the 6x1-Half Cheetah environment, consisting of a robotic cheetah with two legs and 6 joints, where each joint is a different agent and distribution is achieved by enabling agent communication only with the adjacent joints, enforced by a ring graph topology. Since our fundamental contribution is the physics-informed distributed and scalable policy parameterization, we have integrated our proposed policy design in a Trust-Region Policy Optimization reinforcement learning algorithm [Schulman et al., 2015]. This allowed us to test the flexibility of the physics-informed policy parameterization in the sense that it can be integrated in different reinforcement learning algorithms, although it is better suited

to a soft actor-critic approach. Besides, the benchmark includes other state-of-the-art multi-agent reinforcement learning approaches for comparison. These are Heterogeneous Agent Trust Region Policy Optimization (HATRPO) and Heterogeneous-Agent Proximal Policy Optimization (HAPPO) [Kuba et al., 2021], MAPPO, IPPO [De Witt et al., 2020], and MADDPG. Importantly, in all these state-of-the-art methods, the policy of each agent/joint depends on the global observation vector, whereas our policy only depends on the neighboring information.

As it can be observed in Fig. 6.7, our proposed approach is not only able to surpass other methods in multi-robot tasks with simple robot motion dynamics, but also in multi-agent coordination tasks that involve complex robotic platforms. Our method, although less informed than the others due to the networked constraints, improves upon the existing state-of-the-art methods in average episode reward. Besides, by fully exploiting the physical dynamics of the robots, our method presents better sample efficiency than MADDPG, IPPO and MAPPO, and similar sample efficiency than HAPPO and HATRPO. In this sense, our method leverages the fact that the acceleration of the joints is available, so in the open-loop feed forward term $(\mathbf{J}(\mathbf{x}) - \mathbf{R}(\mathbf{x}))\frac{\partial H(\mathbf{x})}{\partial \mathbf{x}}$ of Eq. (6.9) can be known without explicit knowledge on the values of $\mathbf{J}(\mathbf{x})$, $\mathbf{R}(\mathbf{x})$ and $\partial H(\mathbf{x})/\partial \mathbf{x}$. With this experiment we have proved that our approach is also a promising alternative to multi-joint dynamical robots, thus strengthening the utility of the proposal.

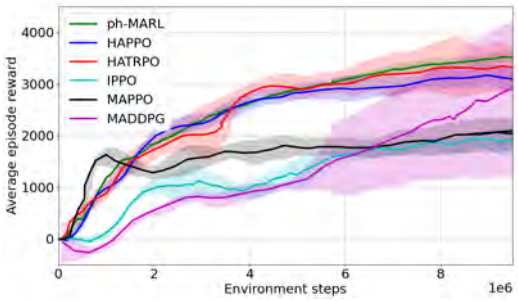


Figure 6.7: Average episode reward obtained by ph-MARL and other existing MARL approaches in the 6x1-Half Cheetah environment.

6.6 Extensions to imitation learning

The physics-informed policy parameterization proposed in this chapter can also be applied to imitation learning settings. Now, we assume that state trajectories from successful task executions are available as training data. Let $\mathbf{x}(t) = [\mathbf{x}_1^\top(t), \dots, \mathbf{x}_n^\top(t)]^\top$ and $\mathbf{u}(t) = [\mathbf{u}_1^\top(t), \dots, \mathbf{u}_n^\top(t)]^\top$ denote the joint state and control of the robot team. Given an initial state $\mathbf{x}(0) = \mathbf{x}_0$, let $\bar{\mathbf{x}}_{0:K} := [\bar{\mathbf{x}}(0), \dots, \bar{\mathbf{x}}(rT), \dots, \bar{\mathbf{x}}(KT)]$ and $\mathbf{x}_{0:K} := [\mathbf{x}(0), \dots, \mathbf{x}(rT), \dots, \mathbf{x}(KT)]$ be demonstrated and learned policy's trajectories, respectively, with K denoting the number of discrete samples r along the trajectories with sampling interval T. Let $\bar{\mathcal{D}} := \{\bar{\mathbf{x}}^l_{0:K}\}_{l=0}^{l=L}$ denote a dataset of $L > 0$ demonstrated trajectories. Let $\mathcal{D} := \{\mathbf{x}^l_{0:K}\}_{l=0}^{l=L}$ be the generated trajectories under policy $\pi_{\boldsymbol{\theta}}$. We aim to learn a control policy that minimizes the distance between the demonstrated and generated trajectories:

$$\mathcal{L}(\mathcal{D}, \bar{\mathcal{D}}) = \frac{1}{KL} \sum_{l=0}^{L} \sum_{r=0}^{K} ||\mathbf{x}^l(rT) - \bar{\mathbf{x}}^l(rT)||_2^2. \tag{6.20}$$

Formally, the problem we consider is:

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{D}, \bar{\mathcal{D}}) \tag{6.21a}$$

$$\text{s.t. } \dot{\mathbf{x}}_i^l(t) = \mathbf{f}_i(\mathbf{x}_i^l(t), \mathbf{u}_i^l(t)),\ \mathbf{x}_i^l(0) = \bar{\mathbf{x}}_i^l(0),\ \forall i, l, \tag{6.21b}$$

$$\mathbf{u}_i^l(t) = \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{x}^l_{\mathcal{N}_i^k}(t)),\ \forall i, l, t. \tag{6.21c}$$

As specified in the formulation above, the learned control policy should also handle time-varying communication and should be adaptable to changes in the total number of robots or the number of neighbors for each robot. Besides, a particular problem of the imitation learning setting is that, if the value of K in the dataset is large, the high dimensionality of the problem and the cumulative nature of the loss between the sequential states of the predicted and demonstrated trajectories leads to exploding gradients. The consequence is either convergence to wrong policy parameters that display poor long-term behaviors, or even an unstable training.

The latter issue can be resolved using Curriculum Learning (CL), a learning paradigm where an agent is trained iteratively following a curriculum to ease learning in difficult problems, such as (6.21). A curriculum is a sequence of training criteria over E training steps, $\mathcal{Q} = \{Q_1, \ldots, Q_E\}$, where Q_e, for $e = 1, \ldots, E$, denotes the training criterion at training step e. The training criterion encompasses the training parameters and features that characterize the complexity of the task, e.g., the features of the scenarios that compose the training set, the optimization goals included in the loss function, or the set of model parameters being optimized [Wang et al., 2021].

Following [Wang et al., 2021], we identify three key elements that we need to specify for each Q_e: (i) a Difficulty Measurer to sort the dataset in elements of increasing difficulty for the learning algorithm, (ii) a Training Scheduler to describe when we should increase the difficulty and for how many training steps and (iii) a loss function capable of adapt to the difficulty of the dataset at each training stage.

Difficulty Measurer. The Difficulty Measurer is responsible for sorting the available data according to its complexity. It uses the length of the trajectories as the complexity criterion to sort the dataset based on the difficulty. We denote by K_e the maximum trajectory length considered for the training criterion Q_e.

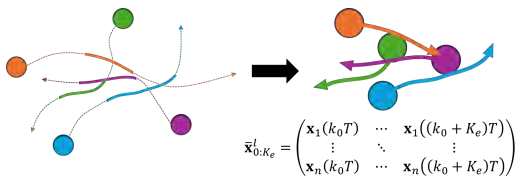


Figure 6.8: Trajectories of length K_e sampled by the Difficult Measurer from the original trajectories of the training dataset.

Since it might be difficult to have example trajectories of every required length, K_e, we propose a partitioning method of the existing trajectories to obtain such data dynamically at each training stage. Besides, with this approach we are also able to fully exploit the information provided by the expert trajectories. In this sense, we assume that the task and the trajectories respect the Markov property, i.e., the evolution of the trajectories over time does not depend on its history. In our context, this implies that transition from $\bar{\mathbf{x}}^l(k)$ to $\bar{\mathbf{x}}^l(k+1)$ only depends

on the configuration of the system at sampled instant k.

Given a trajectory of the dataset $\overline{\mathcal{D}}_K$, $\overline{\mathbf{x}}^l_{0:K} = [\overline{\mathbf{x}}^l(0), \ldots, \overline{\mathbf{x}}^l(KT)]$, we generate a sub-trajectory by randomly picking an initial time step, k_0, in the interval $[0, K - K_e]$. Then, we obtain a demonstrated trajectory of length K_e by considering the following K_e states in the data (see Figure 6.8),

$$\overline{\mathbf{x}}^l_{0:K_e} = [\overline{\mathbf{x}}^l(k_0T), \ldots, \overline{\mathbf{x}}^l((k_0 + K_e)T)]. \tag{6.22}$$

Following this procedure multiple times we generate the training dataset of difficulty K_e, $\overline{\mathcal{D}}_{K_e}$.

By randomly sampling sub-trajectories from the original set of expert demonstrations the curriculum is able to gradually move from local short-term interactions to long-term behaviors, avoiding biases with respect to the initial configuration or the goal of the demonstrations. For instance, in a multi-robot navigation task, the initial training criteria, associated to a small K_e, enables the learning of velocity constraints and collision avoidance among robots; the subsequent training criteria, associated to an increasing K_e, enables the learning of trajectory smoothness and goal-oriented maneuvers.

Training Scheduler. The Training Scheduler is in charge of defining the training step in which the curriculum transitions from training criterion Q_e to training criterion Q_{e+1}. We denote by N_e the number of training steps executed with criterion Q_e. Our proposed design uses a discrete *Baby Step* approach to model the evolution of N_e and the level of difficulty, specified by K_e. In this algorithm, the increments are equal for every training step, identified by the parameters c_K and c_N respectively.

Training Loss. We update the general loss function in Eq. (6.20) to reflect the time-horizon's dynamic behaviour along the training. The proposed dynamic training loss normalizes the error by the time-horizon K_e,

$$\mathcal{L}_e(\mathcal{D}_{K_e}, \overline{\mathcal{D}}_{K_e}) = \frac{1}{K_e L} \sum_{l=0}^{L} \sum_{k=0}^{K_e} ||\mathbf{x}^l(kT) - \overline{\mathbf{x}}^l(kT)||_2^2, \tag{6.23}$$

to make sure that gradients are informative enough in all training stages, but especially during the first training steps, when the value of K_e is much smaller than K.

We evaluate the proposed curriculum strategy in three different tasks:

- Fixed Swapping (FS): robots are initially arranged in two columns, each robot navigates to a diagonally opposite position in the other column. The communication graph is fixed to a ring topology, where each robot i communicates with robots $(i \pm 1)$ mod n.

- Time Varying Swapping (TVS): the task to accomplish is the same as in Fixed Swapping, but considering a time-varying communication topology. The communication graph dynamically adjusts during task executions depending on the distance between robots.

- Flocking (Flock): robots begin scattered in space, and their goal is to navigate towards a common goal destination. Robots should avoid collisions, thus forming an evenly distributed formation around a target location. This scenario also considers a time-varying communication graph, that depends on the distance between robots.

Examples of these tasks are given in Figure 6.10.

We consider a 2-dimensional space for all scenarios, defining the state of a robot as $\mathbf{x}_i(t) = [\mathbf{p}_i(t), \mathbf{v}_i(t)] \in \mathbb{R}^4$, which contains the robot's position $\mathbf{p}_i(t) \in \mathbb{R}^2$ and velocity $\mathbf{v}_i(t) \in \mathbb{R}^2$. For each task, we generate a training dataset composed by $L = 10000$ trajectories of length $K = 50$ and a sampling period $T = 0.05$s. Unless otherwise stated, all the demonstrations include $n = 4$ robots. The trajectories are generated using an expert analytical controller. For the Fixed Swapping and Time-Varying Swapping, the controller is a proportional-derivative regulator that includes a reactive term to avoid collisions (see Eq. (26) in [Sebastián et al., 2023b]). For the Flocking, the controller is the one derived in [Olfati-Saber, 2006].

We evaluate our proposed physics-informed imitation curriculum learning approach with three different policy parameterizations. The first is a Multi Layer Perceptron (MLP), an architecture that assumes a fully connected graph. Second, we design a Graph Neural Network (GNN) parameterization that considers the distributed nature of the multi-robot team and uses $\mathbf{A}(t)$ as graph shift operator. Finally, we employ the architecture proposed in Section 6.3, that we call LEMURS (LEarning Distributed MUlti-Robot interactionS). The curriculum proposed is parameterized with $c_K = 1$, and $c_N = 800$ epochs. A non-curriculum training with a fixed horizon of $K = 5$ serves as a baseline to measure improvements. All the models are trained using the Adam optimizer with a learning rate of 0.001 for $E = 40000$ training steps.

We consider five different metrics for evaluation. The first two are the most relevant since they represent errors between predictions and demonstrations: The square euclidean loss $\mathcal{L}$ defined in Eq. (6.20) and the area $\mathcal{A}$ between predicted and demonstrated trajectories, computed as proposed in Algorithm 1 of Appendix A of [Jekel et al., 2018]. The three other metrics reflect information about a single trajectory. The first two are the average and minimum distances between robots:

$$d_\mu(\mathbf{x}_{0:K}) = \frac{1}{K(n^2-n)} \sum_{k=1}^{K} \sum_{i=1}^{n} \sum_{j=1}^{n} ||\mathbf{p}_i(kT) - \mathbf{p}_j(kT)||_2, \tag{6.24}$$

$$d_{\min}(\mathbf{x}_{0:K}) = \min_{i,j \in \mathcal{V}, k<K} ||\mathbf{p}_i(kT) - \mathbf{p}_j(kT)||_2. \tag{6.25}$$

The last metric is trajectory smoothness, defined as

$$s(\mathbf{x}_{0:K}) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathbf{a}_i^2(kT), \tag{6.26}$$

where $\mathbf{a}_i(t) \in \mathbb{R}^2$ represents the acceleration of robot i at a time t. Acceleration is estimated from the expert trajectories using finite differences.

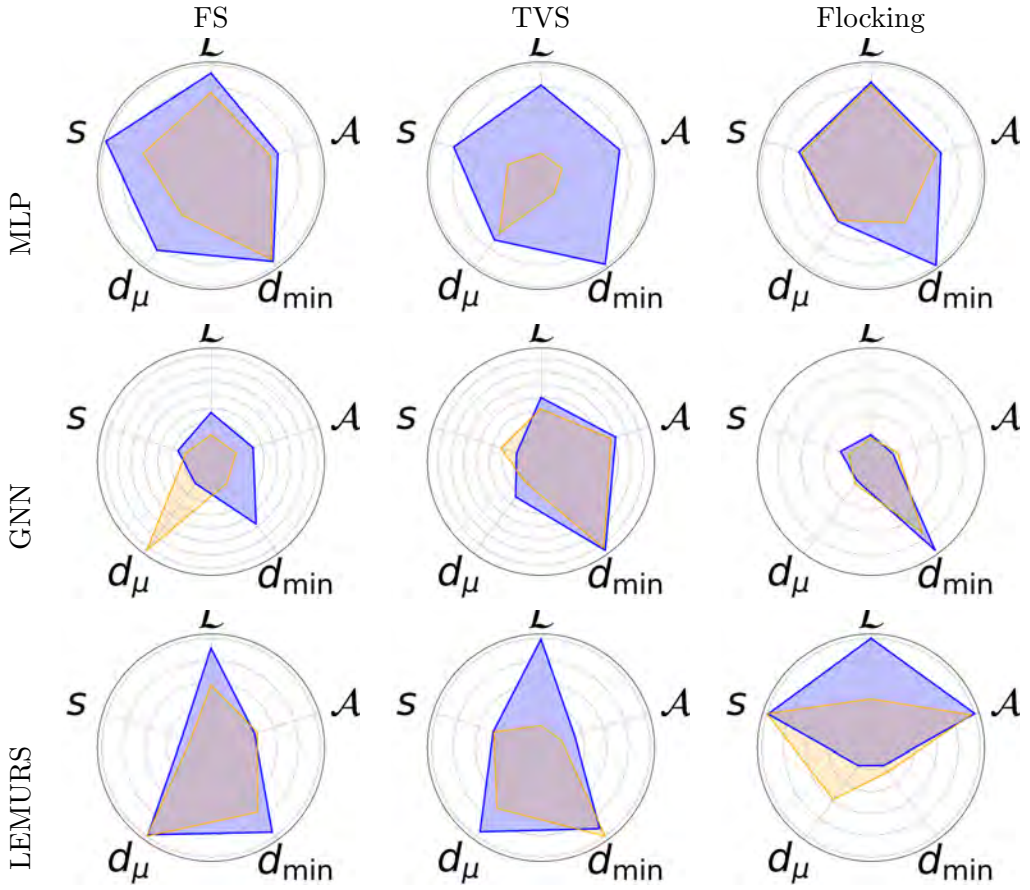


Figure 6.9: Radar pannels for the three tasks and three architectures. Each radar compares the normalized metrics for the case of using the proposed Curriculum Learning (blue) and the non-curriculum counterpart (yellow). The normalization, between 0 and 1, is for each task across all architectures (e.g., same column).

To report the results on $d_\mu(\mathbf{x}_{0:K})$, $d_{\min}(\mathbf{x}_{0:K})$ and $s(\mathbf{x}_{0:K})$, we compute the absolute difference between the demonstrated and predicted trajectories, $\tilde{d}_\mu = |d_\mu(\mathbf{x}_{0:K}) - d_\mu(\bar{\mathbf{x}}_{0:K})|$, $\tilde{d}_{\min} = |d_{\min}(\mathbf{x}_{0:K}) - d_{\min}(\bar{\mathbf{x}}_{0:K})|$ and $\tilde{s} = |s(\mathbf{x}_{0:K}) - s(\bar{\mathbf{x}}_{0:K})|$. The results are the average of the metrics over 100 randomly selected unseen test trajectories.

Table 6.2 presents the metrics obtained for every architecture in each scenario, with and without Curriculum Learning. First of all, in all the cases, LEMURS presents significantly better results than the other two policy parameterizations. On the other hand, the use of Curriculum Learning always improves $\mathcal{L}$, which is the metric that is directly minimized in imitation learning. This improvement is specially significant in the MLP, which is sample inefficient as long as, being the centralized baseline, it considers a fully connected graph where all the robots are aware of the others. In contrast, LEMURS, since it is already sample efficient due to its physics-informed parameterization, exhibits a reduced improvement in the final value of $\mathcal{L}$ because the margin for improvement is tighter. A similar conclusion can be drawn from the area between curves $\mathcal{A}$, which characterizes the long-term behavior of the learned policies. The few cases where the training without Curriculum Learning perform better than with Curriculum Learning present very similar values, so the differences

are not particularly significant.

The other three metrics are not directly optimized during training. Nevertheless, as with $\mathcal{A}$, they are generally better when Curriculum Learning is used during training; and for the cases where the results are worse, the difference with and without curriculum are slight.

Task+Policy	CL	$\mathcal{L}$	$\mathcal{A}$	$\tilde{s}$	$\tilde{d}_{\mu}$	$\tilde{d}_{\min}$
FS+MLP	No	1.384	1.837	260.98	0.686	0.026
FS+MLP	Yes	**0.420**	**1.602**	**13.04**	**0.206**	**0.019**
FS+GNN	No	4.408	3.067	573.57	**0.216**	0.256
FS+GNN	Yes	**3.485**	**2.618**	**530.30**	0.979	**0.138**
FS+LEMURS	No	0.021	**0.391**	43.66	**0.002**	0.020
FS+LEMURS	Yes	**0.017**	0.405	**39.33**	0.004	**0.005**
TVS+MLP	No	7.351	8.121	562.21	0.340	1.510
TVS+MLP	Yes	**1.713**	**2.886**	**163.04**	**0.270**	**0.031**
TVS+GNN	No	5.441	4.553	**550.75**	0.740	0.319
TVS+GNN	Yes	**4.624**	**4.178**	650.51	**0.616**	**0.283**
TVS+LEMURS	No	0.026	0.527	30.11	0.042	**0.002**
TVS+LEMURS	Yes	**0.016**	**0.457**	**29.76**	**0.008**	0.008
Flock+MLP	No	0.165	**0.296**	13.29	0.622	0.032
Flock+MLP	Yes	**0.144**	0.608	**12.25**	**0.608**	**0.0006**
Flock+GNN	No	0.664	0.511	29.61	**0.918**	0.017
Flock+GNN	Yes	**0.641**	**0.534**	**26.90**	0.953	**0.004**
Flock+LEMURS	No	0.023	0.102	**0.95**	**0.055**	**0.050**
Flock+LEMURS	Yes	**0.016**	**0.082**	1.44	0.104	0.054

Table 6.2: Results for the three architectures and the three tasks, with and without Curriculum Learning (best in bold).

For a more visual comparison among architectures and curriculum options, Figure 6.9 represents the five metrics aggregated in a radar pannel, normalizing the metrics with respect to the best performer of each task. It is important to remark that the focus is not on studying which architecture is the best one, but to analyze the impact of our proposed Curriculum Learning algorithm in improving the long-term performance of the learned policies. In this sense, Fig. 6.9 shows that it generally improves all the metrics for all

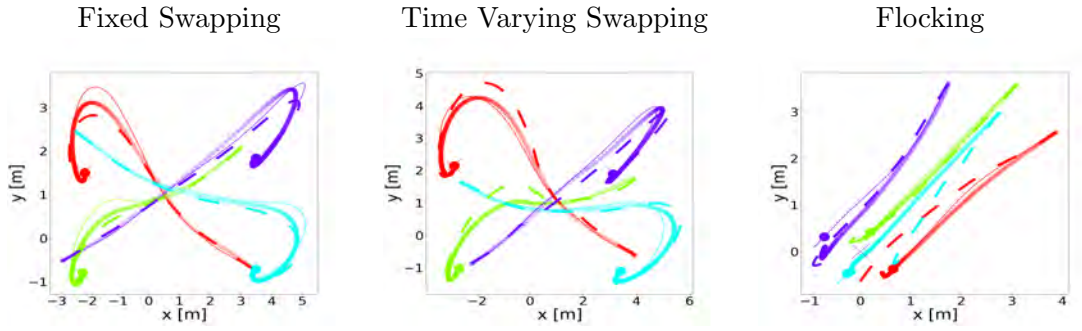


Figure 6.10: Qualitative visualization of the trajectories predicted by LEMURS. The solid lines correspond to the demonstrated trajectories. The dashed lines correspond to the trajectories predicted by the policy trained without curriculum. The trajectories visualized using circle marks correspond to policy trained with curriculum.

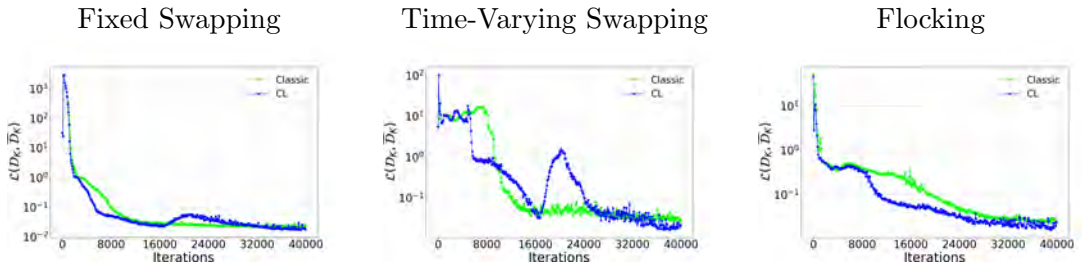


Figure 6.11: Evolution of $\mathcal{L}$ during training for the LEMURS policies. The pannels compare the evolution of the loss using Curriculum Learning (blue) and not using curriculum (green).

architectures and tasks. An interesting case is LEMURS. Table 6.2 proves that there is indeed improvement thanks to the Curriculum Learning, but, since it is the best performer, those improvements are harder to visualize in Fig. 6.9. Fig. 6.10 corroborates that the improvement achieved by the Curriculum Learning leads to qualitative improvement in the long term. The trajectories predicted by the control policies learned using our proposed curriculum are closer to the expert trajectories in the three tasks. Anyways, both Table 6.2 and Fig. 6.9 show that the improvement achieved by Curriculum Learning is greater when the policy parameterization is not sample efficient, i.e., does not consider the networked nature of the multi-robot team.

Another important aspect to consider when using the Curriculum Learning approach is whether it is effective or not in reducing the number of training steps needed to converge to a good policy in terms of accurate long-term behavior. Fig. 6.11 represents the evolution of $\mathcal{L}$ along the training using the LEMURS policy parameterization, with and without curriculum. As it can be observed, even if the final improvement in metrics achieved using Curriculum Learning ends to be marginal, the number of training steps to converge is reduced. This is specially highlighted in the case of Flocking, which is the most complex task among the three. This is an additional advantage of using Curriculum Learning.

Finally, we evaluate the scalability of LEMURS with the number of robots, with and without curriculum. It is worth noting that one of the motivations of designing a curriculum on the length of the trajectories is that graph-based policies already enable scalability with

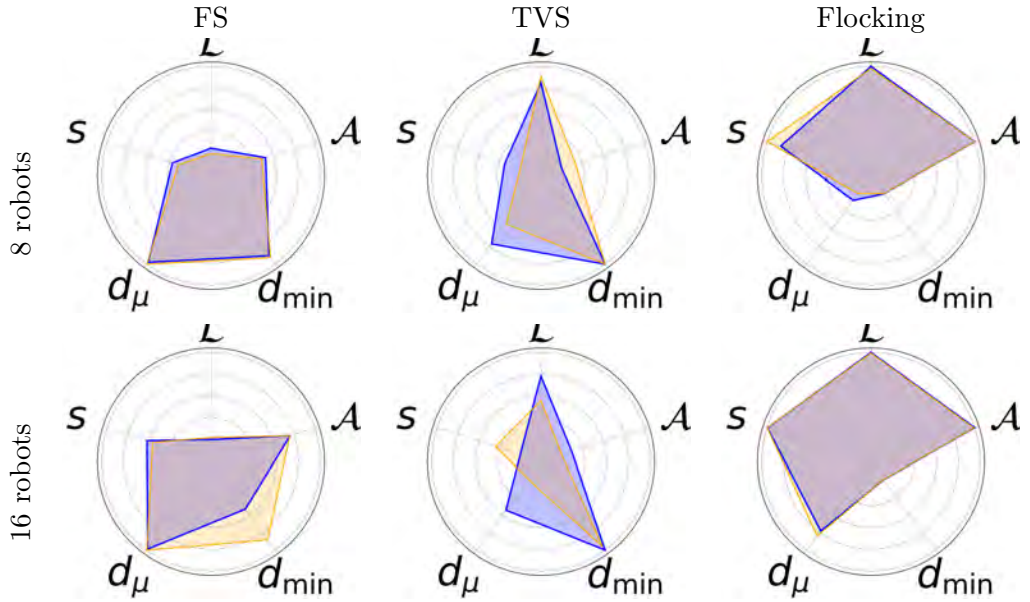


Figure 6.12: Radar pannels for the three tasks and two team sizes, using LEMURS. In all the cases, the policy is trained with $n = 4$ robots. Each radar compares the normalized metrics for the case of using the proposed Curriculum Learning (blue) and the non-curriculum counterpart (yellow).

the number of robots. It is instrumental to assess if the use of a curriculum in the length of the trajectories deteriorates the scalability on the number of robots. Fig. 6.12 uses the same radar representation employed in Fig. 6.9, but in this case we compare performance with and without Curriculum Learning when the number of training robots is $n = 4$ but the number of robots in evaluation is scaled to $n = 8, 16$. Fig. 6.12 allows us to conclude that the scalability with the number of robots is not deteriorated by the use of a curriculum in the length of the trajectories.

6.7 Conclusions

In this chapter, we have addressed the challenge of learning distributed multi-robot control policies that scale with the number of robots. As highlighted in the introduction, the complexity of the interactions between robots and the environment often impedes the analytical derivation of these policies. Our solution focuses on the design of a novel physics-informed policy parameterization based on the combination of port-Hamiltonian mechanics and self-attention networks. The port-Hamiltonian formulation offers several advantages. It naturally encodes the distributed nature of the multi-robot system, ensuring that the control policy only depends on the information available to each robot and its neighbors. Additionally, it respects the energy conservation laws of the individual robot dynamics, leading to a more physically realistic and interpretable model. To handle the time-varying information available to each robot from communication or perception, we have employed self-attention networks. These networks can effectively process variable-length sequences of neighboring states, enabling the control policy to adapt to changes in the communication topology or the number of robots. We have integrated this physics-informed policy

parameterization into a soft actor-critic algorithm, a reinforcement learning approach that has shown promising results in single-robot problems. By treating the entire multi-robot network as a single agent, we have been able to leverage the power of this algorithm while still learning distributed control policies. The effectiveness of our approach has been demonstrated in seven multi-robot scenarios, covering a wide range of cooperative and competitive behaviors, and including a validation experiment in a realistic robotic platform. In all cases, our method has shown superior performance in terms of cumulative reward per robot and scalability compared to other state-of-the-art multi-agent reinforcement learning approaches.

Furthermore, we have extended our physics-informed policy parameterization to imitation learning settings. To address the challenge of learning long-term behaviors, we have proposed a curriculum learning approach that gradually increases the length of expert trajectories used during training. This curriculum has been shown to improve the accuracy and stability of the learning process, particularly in scenarios with complex interactions.

The work presented in this chapter has significant implications for the design and control of multi-robot systems. By enabling the learning of distributed control policies that scale with the number of robots, we open up new possibilities for applications in areas such as environmental monitoring, search and rescue, and cooperative robotics. One motivating future direction is to study how to exploit the physics-informed port-Hamiltonian parameterization of the control policies to enforce performance or safety guarantees. For instance, passivity theory or control barrier functions can be employed to ensure convergence properties of the control policy or design safety modules on top of the control policy that provide collision avoidance guarantees. It also remains to be explored how to bypass the three-message passing scheme to compute $\partial H_{\boldsymbol{\theta}}(\mathbf{x})/\partial \mathbf{x}_j$ and rely only on, e.g., single-message broadcasting protocols or perception cues, since it is currently a computation and implementation bottleneck.

From a broader perspective, there still exist some open questions that deserve further exploration:

- The experiments conducted in competitive scenarios do not allow the simultaneous learning of the teams, i.e., each team must be trained separately, which leads to some performance gap with respect to the theoretical Nash Equilibrium that arises in dynamic multi-agent games. Therefore, a promising research direction is to exploit ideas on distributed policy parameterization developed in this chapter to address multi-agent dynamic games, where agents' interactions are restricted to neighboring intra- and inter-agent constraints. The initial steps have been already taken in [Sebastián et al., 2025].
- The multi-robot tasks considered so far are restricted to low-level commands, such as navigation, collision avoidance, resource collection or exploration. Higher-level commands, such as search and rescue assistance, requires a hierarchical control able to integrate natural language prompts and sensor information to decompose the high-level task and plan in the task space. It remains as open question how to build such hierarchy for multi-robot distributed task planning. Fortunately, there is already

some initial progress on this side [Zhan et al., 2025].

- Finally, a feature that has been assumed so far, particularly in the imitation learning setting, is whether the graph interactions during training are known. This is reasonable in a simulation environment, but harder to sustain when our data comes from recordings of experts. Therefore, the question is: is it possible to learn the underlying graph modeling the interactions among robots or agents within a network from trajectories? This research direction has been started recently, with some initial promising results [Sebastián et al., 2023a].

6.8 Network parameters

The self-attention-based control policy in ph-MARL is parameterized as follows:

- $[\mathbf{R}_{\boldsymbol{\theta}}]_{ij}$: $W = 3$, $h_w = [n_x, 16, 8]$, $r_w = [8, 16, 8]$, $d_w = [16, 8, 16]$; functions $\beta =$ sigmoid, $\chi = \psi =$ swish [Ramachandran et al., 2017].
- $[\mathbf{J}_{\boldsymbol{\theta}}]_{ij}$: $W = 3$, $h_w = [n_x, 16, 8]$, $r_w = [8, 16, 8]$, $d_w = [16, 8, 16]$; functions $\beta =$ sigmoid, $\chi = \psi =$ swish [Ramachandran et al., 2017].
- $H^i_{\boldsymbol{\theta}}$: $W = 3$ layers, $h_w = [n_x, 16, 8]$, $r_w = [16, 8, 8]$, $d_w = [16, 8, 25]$; with functions $\beta =$ sigmoid, $\chi = \psi =$ swish [Ramachandran et al., 2017].

For the variance network we use the same architecture of $[\mathbf{R}_{\boldsymbol{\theta}}]_{ij}$ but with $h_1 = n_x + n_u$ and $d_W = 2$. The other networks used in the ablation studies are as follows:

- MLP: the policy is a single multi-layer perceptron of size $[n \times n_x, n \times n_u, n \times n_u]$ with swish hidden activation function and linear output. For the variance network we use the same architecture but with size $[n(n_x + n_u), n \times n_u, n \times n_u]$.
- MSA: the policy is a single multi-layer perceptron of size $[n \times n_x, n \times n_x]$ with linear output followed by a self-attention layer where query, key and values are directly the feature vector from the multi-layer perceptron, and an additional multi-layer perceptron of size $[n \times n_x, n \times n_u]$ with linear output. For the variance network we use the same architecture but with size $[n(n_x + n_u), n \times n_u]$ for the first multi-layer perceptron.
- GSA: the policy is the same one used to predict $[\mathbf{R}_{\boldsymbol{\theta}}]_i j$ but with $d_W = 2$. For the variance network we use the same architecture but $h_1 = n_x + n_u$.

The $Q_{\boldsymbol{\Pi}_{\boldsymbol{\theta}}}(\mathbf{s}_t, \mathbf{a}_t)$ function is always parameterized as a multi-layer perceptron with layers of size $[n(n_x + n_u), 2n(n_x + n_u), n(n_x + n_u), (n_x + n_u), 1]$ with swish [Ramachandran et al., 2017] activation functions except the last layer, that is linear.

The food collection, grassland, and adversarial scenarios use a neural network to preprocess the observation vector to move from a time-varying observation size to a fixed

state size compatible with the control policies. To do so, the state is a concatenation of the position (2-dimensional vector), velocity (2-dimensional vector), aliveness (boolean quantity), closest goal relative distance (2-dimensional vector) and a 2-dimensional feature vector provided by a neural network. The neural network is composed by a multi-layer perceptron with no hidden layers of size $[n_x - 2, 2]$ and swish activation function, a self-attention layer of size $h_w = r_w = d_w = 2$, and another multi-layer perceptron with no hidden layers of size $[2, 2]$ and linear output.

6.9 Soft actor-critic parameters

The following table details the parameterization of the soft actor-critic algorithm for the different scenarios.

Table 6.3: Soft actor-critic hyperparameters.

Parameter	**Scenario**	**Value**
optimizer	all	Adam [Kingma and Ba, 2015]
r_{comm}	reverse transport, navigation	0.45m
	sampling	0.75m
	food collection, grassland, adversarial	0.15m
n training	reverse transport, sampling, navigation	4
	food collection, grassland, adversarial	8
# parallel environments	all	96
shared r	all	True
maximum steps per episode	reverse transport, navigation, food collection, grassland, adversarial	400
	sampling	1000
replay buffer size	all	2×10^6
initial random steps	all	10^3
γ	all	0.99

Continued on next page

Table 6.3: Soft actor-critic hyperparameters. (Continued)

α_0	all	5
$\alpha_{\min}$	all	0.1
$\alpha_{\max}$	all	10
ρ	all	0.005
learning rate α	all	10^{-5}
learning rate	all	10^{-4}
batch size	all	1024
# training steps	navigation, reverse transport, sampling, food collection, grassland	2×10^6
	adversarial	6×10^5
clip gradients	all	False
reward scaling	all	False
$\sigma_{\min}$	all	e^{-5}
$\sigma_{\max}$	all	e^2
landmark mass	sampling, navigation, food collection, grassland, adversarial	default
	reverse transport	1
evaluation interval	all	10^4
# evaluation episodes per interval	all	10

Part IV

Control Techniques for Power Electronics Problems

Chapter 7

Identification, Estimation and Control of Power Systems

Modern robots are complex systems demanding precise and efficient power management for optimal performance and safety. Control techniques play a fundamental role in regulating the power supply within these systems, ensuring reliable operation and extending operational life. From industrial manipulators to mobile service robots, sophisticated control algorithms address challenges such as varying workloads, energy efficiency, thermal management, and fault tolerance. By intelligently adapting to dynamic conditions and optimizing power utilization, these techniques enhance the capabilities and robustness of robotic systems across diverse domains. Parts I, II and III of this thesis focus on how to do distributed multi-robot control of input-nonaffine systems, how to provide the associated stochastic estimates in a distributed manner, and how to learn the underlying interaction models to, overall, accomplish the distributed multi-robot control of complex systems. However, we should answer the question on how, given such control commands, the robots can effectively exert them by ensuring an effective power supply.

Chapter 7 delves into the crucial role of control techniques in power systems. It emphasizes the increasing importance of precise and efficient power management to ensure optimal performance, safety, and operational longevity in autonomous applications. The chapter explores three core aspects of power systems: identification, estimation, and control, highlighting how advanced techniques address the growing complexities in these systems.

- **Identification**: Section 7.1 underscores the need for accurate models of power system components and their interactions. It discusses the use of black-box models derived from measured data. Advanced techniques like infrared thermal measurements and machine learning algorithms are presented as means to identify system parameters and behaviors under various operating conditions.

- **Estimation**: Real-time state determination in power systems is described as essential for monitoring, predicting potential issues, and enabling effective control actions. Section 7.2 examines how state estimation algorithms process sensor measurements

to provide a coherent picture of the system's condition.

- **Control**: Section 7.3 emphasizes the aim of maintaining stability and desired operating conditions despite disturbances and uncertainties. It discusses how controllers utilize real-time information and system models to adjust power generation, load shedding, and other control actions for efficiency, reliability, and cost optimization.

The chapter showcases the application of these concepts in various power electronic problems. It presents solutions for optimal power converter design, fault-tolerant prediction in photovoltaic-supplied systems, and the development of a nonlinear implementable control for a dual active bridge series resonant converter to provide a fast and stable energy supply. Each section is self-contained, including an introduction, problem formulation, proposed solution, and simulations and experiments.

7.1 Identification: Estimation of Semiconductor Power Losses Through Automatic Thermal Modeling

7.1.1 Introduction

Classical electrical characterization methods such as the double pulse test [Hu and Biela, 2023] do not satisfy the new high-speed switching conditions of devices, which require novel multiple pulse techniques [Li et al., 2019a] for an accurate characterization of power losses. These methods isolate the device from its desired used topology, therefore neglecting the overall parasitics and thermal coupling of the final target power converter [Bahman et al., 2017, Shahjalal et al., 2020]. In this sense, thermal coupling, parasites, limited probe bandwidth and calibration problems lead to robust time-frequency characterization techniques [Azpúrua et al., 2022].

A different line of research is to consider non-invasive methods based on indirect measurements. The most common approach is to take the difference between the measured input power and the output power [Xiao et al., 2007]. However, this is strongly dependent on the precision of the power meter and the losses between the different components cannot be distinguished. Instead, calorimetric methods propose to estimate the power losses through thermal measurements [Anderson et al., 2017]. Calorimetric methods have been applied not only to transistors [Rogers et al., 2023], but also to inductors [Papamanolis et al., 2021], capacitors [Coday and Pilawa-Podgurski, 2023], or microelectronic devices [Cruz-Duarte et al., 2021]. Despite the advantages in accuracy compared to other methods [Anurag et al., 2020], typical calorimetric methods must enclose the device under test inside an insulated chamber, which is not always feasible due to size restrictions [Weimer et al., 2022]. On the other hand, calorimetric methods need an equivalent analytical thermal-electric model of the power converter to relate the temperature and power losses [Foster, 1924, Eleffendi and Johnson, 2016]. For steady-state identification, the analytical models are based on equivalent thermal resistances [Koch et al., 2019, Schiestl et al., 2020, Kohlhepp et al., 2021]. If the dynamical behavioral model is desired, then thermal equivalent capacitors are

included [Hauck and Bohm, 2000,Herrera et al., 2013,Eleffendi and Johnson, 2016] [Kuprat et al., 2023]. Nevertheless, these models are highly dependent on the substrate [Yang et al., 2023] and the technology of the devices, as the coupling effects vary between the components, so these techniques are currently only suitable for simple circuits [Guacci et al., 2020]. When the complexity of the circuit increases, the thermal model becomes more challenging as more power and coupling sources arise [Azurza Anderson et al., 2021,Laloya et al., 2016].

Another alternative is to rely solely on a priori finite element analysis [Bohm et al., 2004, Van Der Broeck et al., 2018]. These methods exploit complex computer tools [ANSYS, 2022] that implement physical electrothermal interactions [Cheng et al., 2020] to model the effects of environmental temperature [Mian et al., 2013], control [Oetinger and Sawodny, 2015], or packaging conductivity analysis [Bohm et al., 2004]. Despite its accuracy in single devices and potential model reduction [Entzminger et al., 2024], the complex parameterization hinders its flexible applicability in real power converters, where it is usually easier to obtain experimental temperature-power profiles that capture all thermal processes. Besides, specially in intricate topologies such as multi-level power converters, it is rather possible to forget to model coupling power-thermal factors, leading to a miss-match between a priori simulated power losses and the actual ones. Therefore, a post-design approach to characterize the actual power losses in the device from thermal measurements is necessary for optimal power converter design.

7.1.2 Problem formulation

The device under study is a general power converter formed by dissipate power sources, like transistors, inductors, or printed circuit board (PCB) tracks. The topology of the power converter is not assumed a priori, so it can be a buck converter, a full-bridge resonant converter, etc. The power dissipation of the converter is characterized by temperature and power measurements, which can be acquired by any available means. For instance, in a transistor, the temperature might be measured from its top capsule surface, using a thermocouple or an infrared camera. On the other hand, the power of the devices is measured in DC, e.g., using low continuous current methods to decouple the effects of the tracks in the power measurements. Section 7.1.4 details how to measure the power in the devices.

Formally, the power converter is characterized by a set of n power measurement points $\mathcal{P} := \{1, \dots, i, \dots, n\}$ and a set of m temperature points $\mathcal{T} := \{1, \dots, j, \dots, m\}$. We denote by $P^i \in \mathcal{P}$ and $T^j \in \mathcal{T}$ the power at measurement point i and the temperature at measurement point j respectively.

Given all these measurable quantities, we aim to identify the relationship between power and temperature in the power device as a linear discrete-time dynamical system. To do so, we define the state and input of the thermal dynamics of the power converter as $\mathbf{x} = [P^1, \dots, P^i, \dots, P^n]^\top$ and $\mathbf{u} = [T^1, \dots, T^j, \dots, T^m]^\top$. The ordering of the quantities in $\mathbf{x}$ and $\mathbf{u}$ is arbitrary and does not affect the automatic modeling process. Then, the linear

discrete-time power-temperature dynamics is defined as

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k). \tag{7.1}$$

In Eq. (7.1), $\mathbf{A}$ and $\mathbf{B}$ are unknown matrices, whereas $k \in \mathbb{N}$ denotes the discrete instants when the system is sampled, with $\Delta t > 0$ the sample time. This model can be considered as the Generalized Average Model [Sanders et al., 1991, Mahdavi et al., 1997] of the relationship between the power and temperature in the converter elements in discrete time. The linear assumption is also supported by the fact that the relationship between temperature and power in a semiconductor device is dominated by linear dynamics, whereas the relationship between current, voltage and power is non-linear, as it is observed in Fig. 7.1. Note that there is no assumption regarding the underlying structure of both matrices, in contrast to classical power losses identification techniques where the elements of these matrices are parameterized by equivalent resistance and capacitor parameters.

Now, assume that the quantities in $\mathbf{x}$ and $\mathbf{u}$ can be measured and recorded during $K > 0$ instants of time. We can build a set of data $\mathcal{D} = \{\mathbf{x}(k), \mathbf{u}(k)\}_{k=1}^{K}$. Given a collection of measurements $\mathcal{D}$, the goal is to identify the dynamics in (7.1) characterized by $\mathbf{A}$ and $\mathbf{B}$, such that the model is as accurate as possible in estimating the power losses and it fulfils any a priori constraint of the power converter. The a priori constraints can be, e.g., the sign of an element of matrix $\mathbf{B}$ or a rank condition over matrix $\mathbf{A}$ to ensure a well-conditioned identification. Formally, the problem to solve is the following:

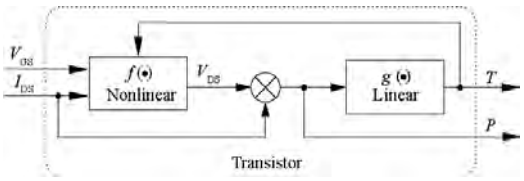


Figure 7.1: Block diagram of the dynamics of a transistor. The dynamics can be decoupled in two terms, (i) a non-linear term that models the relationship between current, voltage and power, and (ii) a linear term that models the relationship between power and temperature. In this work we are interested in the latter.

$$\min_{\hat{\mathbf{A}},\hat{\mathbf{B}}} \ \mathcal{L}(\mathcal{D}) \tag{7.2a}$$

$$\text{s.t. } \mathbf{x}(k+1) = \hat{\mathbf{A}}\mathbf{x}(k) + \hat{\mathbf{B}}\mathbf{u}(k), \tag{7.2b}$$

$$g_{\mathbf{A}}(\hat{\mathbf{A}}) \leq 0, \ h_{\mathbf{A}}(\hat{\mathbf{A}}) = 0, \tag{7.2c}$$

$$g_{\mathbf{B}}(\hat{\mathbf{B}}) \leq 0, \ h_{\mathbf{B}}(\hat{\mathbf{B}}) = 0. \tag{7.2d}$$

In problem (7.2), $\mathcal{L}(\mathcal{D})$ is an objective function that measures the quality of the estimation accuracy of the identified model given by $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$. Meanwhile, $g_{\mathbf{A}}(\hat{\mathbf{A}})$, $h_{\mathbf{A}}(\hat{\mathbf{A}})$, $g_{\mathbf{B}}(\hat{\mathbf{B}})$ and $h_{\mathbf{B}}(\hat{\mathbf{B}})$ are functions that model any constraint on $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$.

7.1.3 Optimization-based identification

To find the solution for problem (7.2) we need, first, to generate the dataset $\mathcal{D}$. Ideally, a temperature input $\mathbf{u}(k)$ would be applied to the power converter, measuring the evolution of the power $\mathbf{x}(k)$. However, in practice, this not practical because the temperature in the devices is a consequence of the power losses during the operation of the power converter. Thus, in real power converters, the power losses are generated from an electrical

excitation and the temperature is allowed to evolve freely. More information is provided in Section 7.1.4. For now, the consequence is that problem (7.2) cannot directly be solved nor $\mathbf{A}$ and $\mathbf{B}$ identified from $\mathcal{D}$ and through $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$, so an alternative method must be developed to estimate the power losses given the temperature of the power converter.

To do so, the first step is to identify the complementary dynamic temperature-power model. This is described by the following discrete-time linear dynamics:

$$\mathbf{u}(k+1) = \bar{\mathbf{A}}\mathbf{u}(k) + \bar{\mathbf{B}}\mathbf{x}(k). \tag{7.3}$$

This model is directly related to the collection of the dataset $\mathcal{D}$, where the power converter is excited with electrical signals such that power losses appear in the different components of the converter and the temperature evolves accordingly.

The next step is to identify $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ such that they can be used later to estimate power losses from temperature measurements, and the requirements in terms of accuracy and restrictions in problem (7.2) are accomplished. Let reformulate Eq. (7.3) as:

$$\mathbf{u}(k+1) = \begin{pmatrix} \bar{\mathbf{A}} & \bar{\mathbf{B}} \end{pmatrix} \begin{pmatrix} \mathbf{u}(k) \\ \mathbf{x}(k) \end{pmatrix} = \mathbf{W}\mathbf{z}(k). \tag{7.4}$$

The idea of this reformulation is to allow to frame the identification problem as a least-squares minimization. Thus, we define $\hat{\mathbf{W}} = \begin{pmatrix} \hat{\bar{\mathbf{A}}} & \hat{\bar{\mathbf{B}}} \end{pmatrix}$ as the estimated temperature-power model, and we define the error between predicted and actual temperature as

$$\mathbf{e}(k) = \mathbf{u}(k) - \hat{\mathbf{W}}\mathbf{z}(k-1) \tag{7.5}$$

Then, to identify the most accurate model in terms of Mean Square Error (MSE), we have to minimize the sum of the norm of all the errors, leading to

$$\begin{pmatrix} \hat{\bar{\mathbf{A}}}^* & \hat{\bar{\mathbf{B}}}^* \end{pmatrix} = \arg\min_{\hat{\bar{\mathbf{A}}}, \hat{\bar{\mathbf{B}}}} \sum_{k=2}^{K} ||\mathbf{e}(k)||_2^2, \tag{7.6}$$

where $||\bullet||_2$ is the L2-norm. Under the assumption that no further requirements are needed, the solution of (7.6) is given by

$$\begin{pmatrix} \hat{\bar{\mathbf{A}}}^* & \hat{\bar{\mathbf{B}}}^* \end{pmatrix} = (\mathbf{Z}^\top \mathbf{Z})^{-1}\mathbf{Z}^\top \mathbf{U} = \mathbf{Z}^\dagger \mathbf{U}. \tag{7.7}$$

Matrix $\mathbf{Z} = [\mathbf{z}(1), \mathbf{z}(2), \ldots, \mathbf{z}(K-1)]$ and matrix $\mathbf{U} = [\mathbf{u}(2), \mathbf{u}(3), \ldots, \mathbf{u}(K)]$ stack the $\mathbf{z}(k)$ and $\mathbf{u}(k)$ elements of the dataset $\mathcal{D}$ to form the input and output data matrices respectively. Meanwhile, $\mathbf{Z}^\dagger = (\mathbf{Z}^\top \mathbf{Z})^{-1}\mathbf{Z}^\top$ is the Moore-Penrose inverse of $\mathbf{Z}$.

Given the optimal $\hat{\bar{\mathbf{A}}}^*, \hat{\bar{\mathbf{B}}}^*$ in the MSE sense, the power losses can be estimated from temperature measurements using the identified dynamics from Eq. (7.3). The temperature-power dynamics leads to:

$$\begin{aligned} \mathbf{x}(k) &= ((\hat{\bar{\mathbf{B}}}^*)^\top \hat{\bar{\mathbf{B}}}^*)^{-1} (\hat{\bar{\mathbf{B}}}^*)^\top (\mathbf{u}(k+1) - \hat{\bar{\mathbf{A}}}^* \mathbf{u}(k)) \Rightarrow \\ \mathbf{x}(k-1) &= ((\hat{\bar{\mathbf{B}}}^*)^\top \hat{\bar{\mathbf{B}}}^*)^{-1} (\hat{\bar{\mathbf{B}}}^*)^\top (\mathbf{u}(k) - \hat{\bar{\mathbf{A}}}^* \mathbf{u}(k-1)) \end{aligned} \tag{7.8}$$

Equation (7.8) defines an estimator for the power losses given the temperature at the power converter.

At this point, a few considerations are in order. First, according to Eq. (7.8), the estimator has a delay of one discrete step. This is not a problem since $((\hat{\hat{\mathbf{B}}}^*)^\top \hat{\hat{\mathbf{B}}}^*)^{-1}(\hat{\hat{\mathbf{B}}}^*)^\top$ and $\hat{\hat{\mathbf{A}}}^*$ can be pre-computed from the calibration data in $\mathcal{D}$, and a sufficiently small Δt can be chosen to fit the application requirements. Second, Eqs. (7.7) and (7.8) include the Moore-Penrose inverse of the matrices $\mathbf{Z}$ and $\hat{\hat{\mathbf{B}}}^*$. The Moore-Penrose inverse matrix of any real $n \times m$ matrix $\mathbf{K}$ exists if and only if the rank of $\mathbf{K}$ is maximum, that is, if $\mathsf{rank}(\mathbf{K}) = \min(n, m)$. For Eq. (7.7), this means that there are as many linearly independent measurements $\mathbf{z}(k)$ as the number of temperature and power test points, which is easy to accomplish since, typically, $K >> (n + m)$. Another option is to use a regularizer that avoids ill-conditioned identification, so $(\mathbf{Z}^\top \mathbf{Z})^{-1}$ in Eq. (7.7) is replaced by $(\mathbf{Z}^\top \mathbf{Z} + \varepsilon \mathbf{I})^{-1}$, where $\varepsilon > 0$ is a small constant designed by the practitioner and $\mathbf{I}$ is the identity matrix. This regularizer penalizes large values of $\hat{\hat{\mathbf{A}}}^*, \hat{\hat{\mathbf{B}}}^*$ and, therefore, provides robustness against the noise or possible outliers in the measurements used for the identification. On the other hand, to ensure that $\hat{\hat{\mathbf{B}}}^*$ in Eq. (7.7) has full rank, a constraint can be used, leading to the following reformulation of the optimization problem:

$$\left(\hat{\hat{\mathbf{A}}}^* \quad \hat{\hat{\mathbf{B}}}^*\right) = \arg\min_{\hat{\hat{\mathbf{A}}}, \hat{\hat{\mathbf{B}}}} \sum_{k=2}^{K} ||\mathbf{e}(k)||_2^2 \tag{7.9a}$$

$$\text{s.t. } \mathsf{rank}(\hat{\hat{\mathbf{B}}}) = n + m, \tag{7.9b}$$

Finally, we can include additional constraints to (7.9) from, e.g., a priori knowledge on the properties of the power converter. For instance, the element (i, j) of $\hat{\hat{\mathbf{B}}}^*$ is necessarily positive if we know that the power losses of the measurement point i of vector $\mathbf{x}$ always increase the temperature at the measurement point j of vector $\mathbf{u}$. Thus, with the additional constraints, problem (7.9) turns to be

$$\left(\hat{\hat{\mathbf{A}}}^* \quad \hat{\hat{\mathbf{B}}}^*\right) = \arg\min_{\hat{\hat{\mathbf{A}}}, \hat{\hat{\mathbf{B}}}} \sum_{k=2}^{K} ||\mathbf{e}(k)||_2^2 \tag{7.10a}$$

$$\text{s.t. } \mathsf{rank}(\hat{\hat{\mathbf{B}}}) = n + m, \tag{7.10b}$$

$$g_{\bar{\mathbf{A}}}(\hat{\hat{\mathbf{A}}}) \leq 0, \; h_{\bar{\mathbf{A}}}(\hat{\hat{\mathbf{A}}}) = 0, \tag{7.10c}$$

$$g_{\bar{\mathbf{B}}}(\hat{\hat{\mathbf{B}}}) \leq 0, \; h_{\bar{\mathbf{B}}}(\hat{\hat{\mathbf{B}}}) = 0. \tag{7.10d}$$

The optimization problem in (7.10) is non-convex due to constraint (7.10b). Nevertheless, there exist many solvers and optimization methods that find the local/global minima of (7.10) with guarantees of convergence to a local minima, including those based on convex relaxations [Sebastián et al., 2021b,Sebastián et al., 2023a] or proximal-gradient-based methods [Henrion and Lasserre, 2004].

7.1.4 Characterization methodology

After providing the theoretical foundations, we illustrate the experimental procedure to characterize the thermal properties of a power converter. We recall that our goal is, given an already designed power converter, identify its actual power-thermal dynamics, with the main target on the semiconductor power losses estimation.

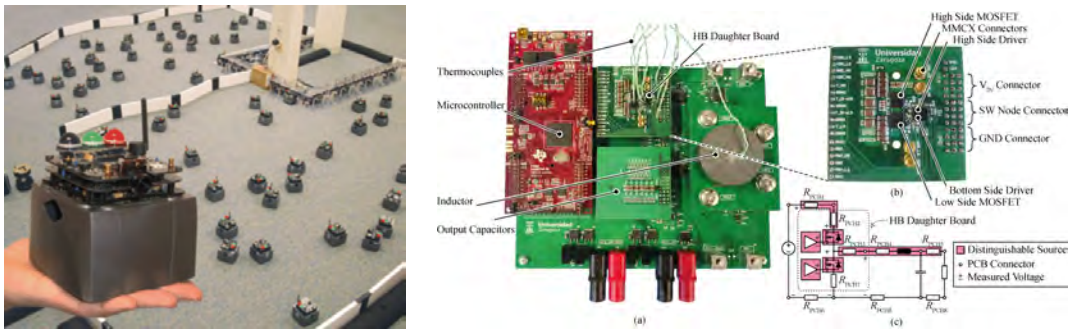


Figure 7.2: Hardware design of the synchronous buck converter under study (the left figure is for illustration purposes, source: [McLurkin and Yamins, 2005]). (a) Illustration of the entire power converter, along with its main parts. (b) Half-bridge board, along with its main components. (c) Schematic circuit of the synchronous buck converter, highlighting the main thermal sources and connections.

First, for a consistent characterization of a linear system, all the power sources are excited independently. In addition, all the components must remain connected and the different power sources cannot be removed for an independent characterization, because any change in the physical connection of the devices can modify the thermal behavior. Second, the complexity of the coupling thermal dynamics depends on the physical design of the converter. Typically, high-frequency low-power converters tend to have a more compact design, leading to greater couplings between components, whereas the design of the modules in high-power converters tends to be thermally isolated. In this sense, not all power sources have the same level of interest. Therefore, for every power converter, a study of the influence of the surrounding components on the semiconductors must be performed for a higher accuracy of the power estimation. As most power converters have transistors, PCB tracks, drivers and inductors, these are the target power sources we will consider. To measure temperature with a low level of noise, a general rule is to choose representative points in terms of power loss sensitivity. For instance, the most representative point in a semiconductor is the junction; however, since the junction is usually inaccessible, a closed enough location can be used. Another example is the inductor: depending on its geometry, different measurement points can be placed on the copper or the core.

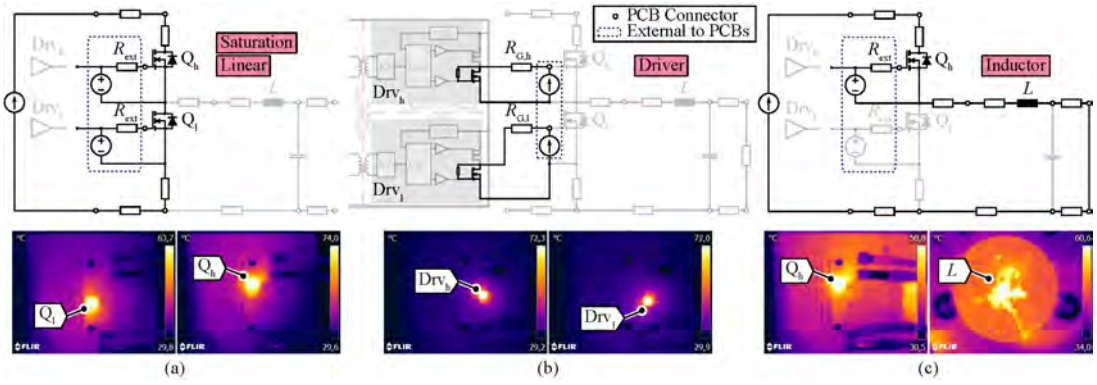


Figure 7.3: Calibration circuit diagrams and associated calorimetric measurements. (a) Transistors (saturation configuration) and power loop tracks (linear configuration). (b) Driver (driver configuration). (c) Inductor and output tracks (inductor configuration). The infrared camera frames illustrate the main sources of thermal dissipation for each calibration circuit.

We study a synchronous buck power converter to illustrate the approach. The setup is shown in Fig. 7.2. To generate $\mathcal{D}$, thermocouples capture the thermal measurements, with a sampling rate of 1 s. In this section, we use thermal frames from an infrared camera for visually understanding the characterization process, they are not used as inputs of the model. It is noteworthy the importance of a proper attachment of the thermocouples to ensure the repeatability of the tests. In addition, the thermocouples may suffer from pickup noise in switching conditions. This depends on the specific power converter design; if necessary, screened extension cables, shielded thermocouples or other thermal measurements methods as infrared thermal cameras or optical fiber can be used.

On the other hand, the microcontroller's ADCs capture the electrical measurements with a sampling rate of 100 ms. Nonetheless, any other measurement means is valid. The DUTs are 100 V@ 3 mΩ Infineon OptiMOS™ 6 silicon transistors [Technologies, 2021]. Fig. 7.2c presents the circuit diagram of the topology along with its PCB tracks and the available voltage measurement points. The different heat sources are highlighted in red. Ground PCB tracks are overlooked because they are designed as large copper planes, so they can be considered as an equipotential surface.

After describing the general practical background and the power converter under study, in the following subsections we detail how to calibrate each component.

7.1.5 Transistors calibration

In all power converters there exists a high thermal coupling between the transistors and the PCB tracks, which is crucial to characterized for an accurate thermal modeling. Typically, to calibrate the transistors, high currents are needed to generate large losses on the semiconductors since this calibration stage is done in DC. Instead, to isolate switching and DC losses from each other, the transistor gate voltage shall not be fully activated, forcing its operation in the saturation region. In this region, a low DC current is enough for bringing the transistor to its thermal limit without heating up the tracks. This method is universal as it is valid for any transistor technology [Sanz-Alcaine et al., 2023]. On the other hand, the gate voltage shall be below the temperature compensation point to ensure that the temperature coefficient of the current, $\alpha = \frac{dI_D}{dT}$, is always positive. To avoid thermal instability, the transistor current must be limited.

The circuit diagram to calibrate the devices is shown in Fig. 7.3a, where the input power supply is used in current mode (1 A for the target converter). The probe test points (MMCX) in Fig. 7.2b are used as connectors for an external voltage supply. Due to transconductance effects, it is desirable to use a linear power supply such that ripples in the gate voltage do not provoke ripples in the power. On the other hand, a resistor R_{ext} is placed externally to the PCB, in series between the gate and the power supply, to mitigate potential oscillations between the power supply and the parasitic capacitances of the MOSFET. The existing surface-mount gate resistors are removed to avoid undesirable current flows between driver and transistors. Furthermore, the converter load is removed to allow free DC current flow when its impedance is similar to Q_l in the saturation region.

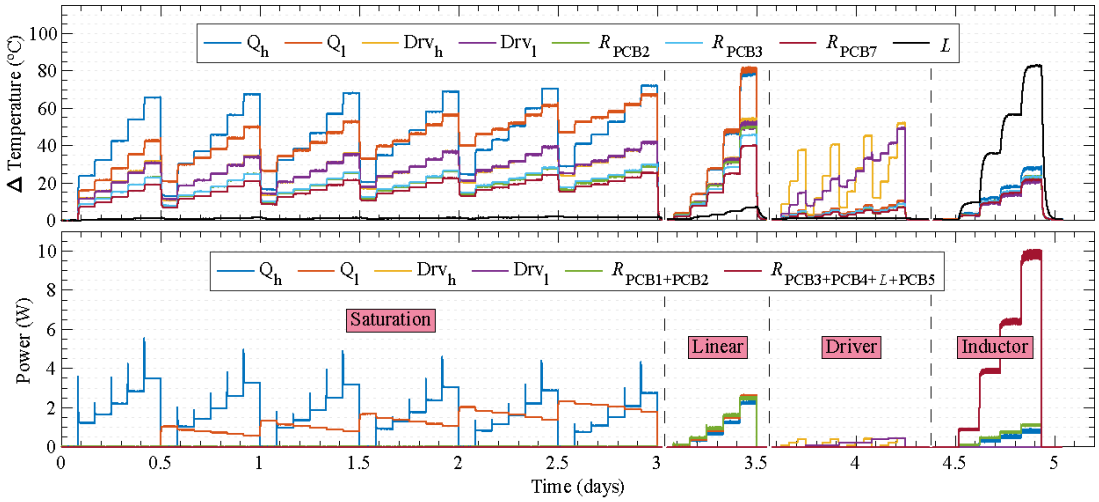


Figure 7.4: Collected dataset to evaluate the synchronous buck converter under study. The four calibration steps are separated by dashed lines.

This external supply injects gate voltages slightly above the threshold voltage of the device. For illustration purposes, Fig. 7.3a. shows thermal images of the low side transistor, Q_l and the high side transistor, Q_h when they are electrically excited independently. Tracks are heated up as a consequence of the transistors' thermal load and not because of their self-heating effect.

7.1.6 Power loop tracks calibration

Once the transistors are calibrated, it is possible to force large currents through them (up to 25 A for the target converter) to excite PCB power loop tracks, characterizing the thermal properties of the PCB by using the same setup of Fig. 7.3a, but with the gate voltage recommended by the manufacturer for switching in the linear region of the MOSFET (10 V). Since the calibration data already includes the individual excitation of the transistors, the temperature increase can only be due to the power loop tracks. These are R_{PCB1} and R_{PCB2}, which have been grouped into a single power source ($R_{\mathrm{PCB1+PCB2}}$) as shown in Fig. 7.2c. For simplicity and without loss of generality, power losses that depend on the frequency, such as the skin effect, are overlooked, but could be included as a perturbation in the system dynamics.

7.1.7 Driver calibration

Simultaneous high switching speed and large semiconductors' gate current produce a self-heating effect on the driver output stage. To minimize the gate loop inductance, the distance between the drivers and the semiconductors is usually reduced as much as possible. Therefore, they are considered an additional power source that needs to be calibrated.

The driver is the Infineon dual-channel isolated MOSFET gate-driver 2EDF7275K [Technologies, 2018]. It is optimized for the driving of OptiMOS™ devices, being able

to provide a 4 A/8 A source/sink from its output stage. This is possible thanks to two rail-to-rail output stages, built from complementary pairs of PMOS and NMOS transistors for the high side ($\mathrm{Drv_h}$) and low side ($\mathrm{Drv_l}$) inside the same package. When both $\mathrm{Q_l}$ and $\mathrm{Q_h}$ switches operate in the same regime, the shape of their current evolution is similar, and therefore it is hard to disambiguate their thermal behavior. In contrast, when they are in different operating conditions, the devices present differences in their thermal curves. Thus, it is recommended to consider them as independent power sources for a greater accuracy of the calibration stage.

A power supply can be used to limit the DC current and sweep it through each of the outputs pairs (up to 300 mA for the target converter). The output NMOS transistor is self-polarized due to the drain-source resistor and has the effect of thermal load, so there is no need of fixing the gate voltage as in the transistor calibration stage. This way, it is possible to generate the corresponding dataset of power and temperature for both output ports. To do so, we re-solder the $R_{\mathrm{G,h}}$ and $R_{\mathrm{G,l}}$ gate resistors with a value of 0 Ω and use the MMCX connector of V_{GS} for the current source connection, as shown in Fig. 7.3b. Current only flows in the direction of the driver due to the high input impedance of DUT MOSFETs. Fig. 7.3b also depicts the associated thermal images of this process. On the left, only the high side ($\mathrm{Drv_h}$) is excited, whereas on the right only the low side ($\mathrm{Drv_l}$) is excited.

7.1.8 Inductor and output tracks calibration

The ideal converter modeling entails the individual calibration of every power source. However, due to the importance of minimizing changes to the mechanical system, it is not possible to attach wires to the inductor [VISHAY, 2021] connectors, as this could introduce additional power sources or act as a heat sink. In addition, since our main goal is to estimate the semiconductor power losses, and the current that flows through the inductor also flows through its surrounding PCB tracks, then R_{PCB3}, R_{PCB4} and R_{PCB5} are grouped into an individual power source ($R_{\mathrm{PCB3+PCB4+}L\mathrm{+PCB5}}$) as highlighted in Fig. 7.2c. For its calibration, we use the configuration illustrated in Fig. 7.3c, where a power supply is used in current mode and forced to circulate through the high side MOSFET (up to 15 A because of inductor thermal limitations). The effects of power loop tracks R_{PCB1}, R_{PCB2}, and $\mathrm{Q_h}$ have already been independently characterized in previous tests, so additional heat on the system is due to this new power source. Thermal images of Fig. 7.3c show the heating on the high side device ($\mathrm{Q_h}$) while the resistance R_{PCB4} is also heated up. The image on the right shows the heated inductor.

7.1.9 Automatic thermal modeling results

We evaluate the automatic approach proposed in Section 7.1.3 with the synchronous buck converter described in Section 7.1.4. Fig. 7.4 represents the collected data that builds the dataset $\mathcal{D}$. The temperature measurements are oversampled (×10) to match the available power data and represented as relative quantities with respect to measured ambient tem-

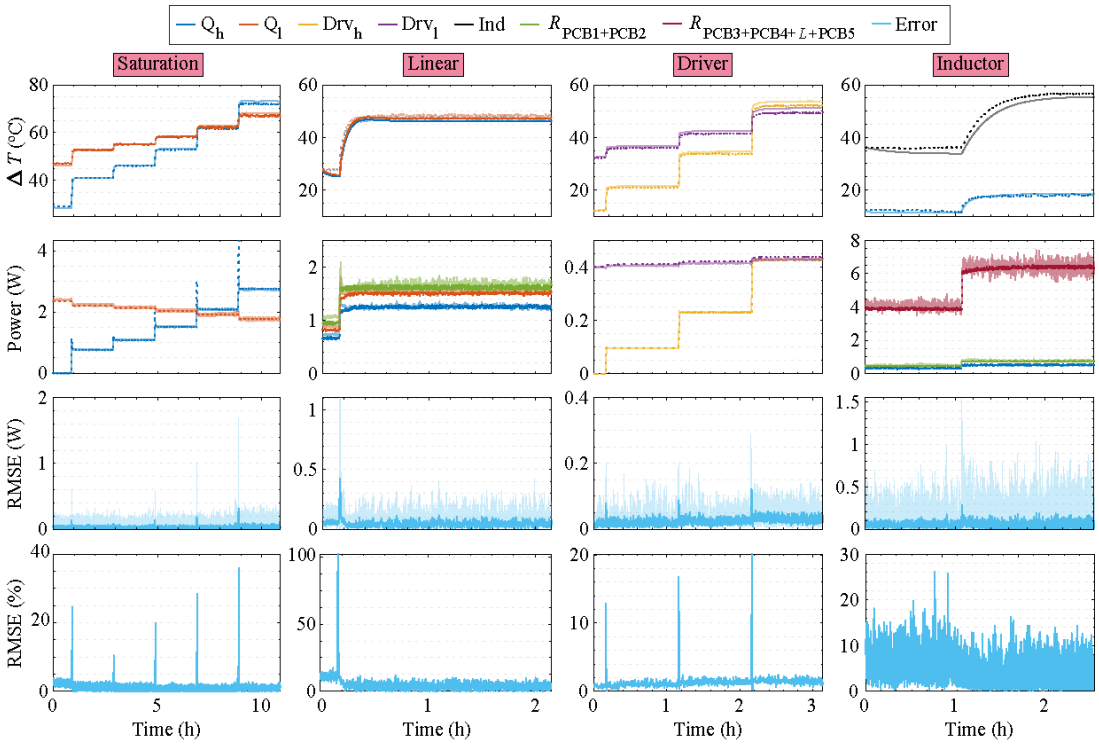


Figure 7.5: Quantitative results. The figure shows, for the four different configurations, the following dynamic information, where dashed lines represent the measurement and bold lines the estimations: (first row) temperature estimation from the identified $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$, obtained from the solution of (7.9); (second row) power losses estimation results from the estimator developed in Eq. (7.8), where each column depicts the relevant heating elements; (third row) evolution of the mean μ_{RMSE} and confidence interval between real and estimated power with time; (fourth row) evolution of the normalized mean $\tilde{\mu}_{\text{RMSE}}$ between real and estimated power with time. It is seen that, for all the regimes, both temperature and the power losses are accurately estimated.

perature. To generate the dataset, we sequentially conduct all the calibration steps detailed in Section 7.1.4, as depicted in Fig. 7.4. To ensure well-posedness in the generation of the dataset, we leave enough time between calibration steps to ensure that the boundary conditions are respected and steady-state is reached. In this sense, for each power converter configuration, we wait until the steady-state is also reached ($\approx$ 2h for the power converter under study).

Regarding the voltage measurements, in constant current conditions, we remark that its dynamics behavior changes along with the sign of α (value of V_{GS}) [Sanz-Alcaine et al., 2023], which can be seen at the beginning of the transient interval in Fig. 7.4. Thus, in the saturation regime, the temperature in Q_h increases while the power decreases. On the other hand, in the linear regime, the temperature in Q_h increases when the power increases. This nonlinear behavior in the electrical data does not affect the linear relationship between temperature and power as illustrated in Fig. 7.1. The experiments also prove the importance of calibrating the PCB tracks: in the linear regime, the power losses are similar to those in the transistors, despite reaching different temperature values.

To evaluate the proposed approach, we split the dataset $\mathcal{D}$ in two different sets. The first set, $\mathcal{D}_1$, gathers most of $\mathcal{D}$ and is used for identifying the temperature-power dynamics

of the power converter. The second dataset, $\mathcal{D}_2$, composed by the rest of $\mathcal{D}$, is used for evaluation. In particular, this split is conducted for each calibration step independently, so we ensure that both the identification and evaluation stages have data from all the calibration steps.

7.1.10 Automatic identification of power-temperature dynamics

The results of the identification are shown in Fig. 7.5. To assess the model, we initialize the state and input vectors of the identified dynamics with the initial configuration of the test dataset $\mathcal{D}_2$. After that, we conduct two types of open-loop simulations. The first one assesses the accuracy of the identified $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$. Using as input the sequence of power measurements $\mathbf{x} \in \mathcal{D}_2$, we leave the temperature $\hat{\mathbf{u}}$ evolve freely, comparing the obtained estimates with the associated real temperature measurements $\mathbf{u} \in \mathcal{D}_2$. However, the main goal of this work is to estimate power losses from temperature measurements. Therefore, the second type of open-loop simulations assesses the estimator proposed in Eq. (7.8). Using as input the sequence of temperature measurements $\mathbf{u} \in \mathcal{D}_2$, we leave the power losses $\hat{\mathbf{x}}$ evolve freely, comparing the obtained estimates with the associated real power losses $\mathbf{x} \in \mathcal{D}_2$. In Fig. 7.5, the real measurements are depicted in dashed lines while the estimates are represented with bold lines with less color opacity. Besides, for quantitative analysis, we compute the mean, the standard deviation and the normalized mean of the Root Mean Square Error (RMSE) across power losses as

$$\mu_{\text{RMSE}} = \frac{\sum_{i=1}^{n} ||\hat{\mathbf{x}}_i - \mathbf{x}_i||_2}{n}, \quad \sigma_{\text{RMSE}} = \sqrt{\frac{\sum_{i=1}^{n} (\hat{\mathbf{x}}_i - \mu_{\text{RMSE}})}{\mu_{\text{RMSE}}}}$$

and

$$\tilde{\mu}_{\text{RMSE}}(k) = \frac{\mu_{\text{RMSE}}}{||\mathbf{x}(k)||_2}.$$

The standard deviation is used to depict the 95 % confidence interval of the estimation ($\mu_{\text{RMSE}} + 2\sigma_{\text{RMSE}}$). Success in power and temperature estimation can be verified from the dynamic results of Fig. 7.5 where, for the four configurations, low estimation error is achieved. For the case of temperature estimation, the model is able to accurately follow the real system dynamics for all situations. Slightly larger errors are found in the inductor configuration. This could be due to the reduced number of temperature measurement points in this area. On the other hand, to reduce the impact of noise measurement in the data, a moving average of 5 s was used. This filtering enhanced the estimation, but it can increase the instantaneous error in the transient due to the time delay inherent to the noise filtering. This effect seems to be more predominant in the semiconductors. To compensate these errors, it is enough to wait the delay time. Higher order filters can be applied to the measurements if larger noise is found.

Importantly, thanks to identifying the optimal linear discrete-time dynamic system in terms of the acquired data, systems theory tools can be used to analyze the behavior of the real power converter. Fig. 7.7 represents the sensitivity matrix in steady-state given by

$$\mathbf{S} = (\mathbf{I} - \hat{\mathbf{A}}^*)^{-1} \hat{\mathbf{B}}^*$$

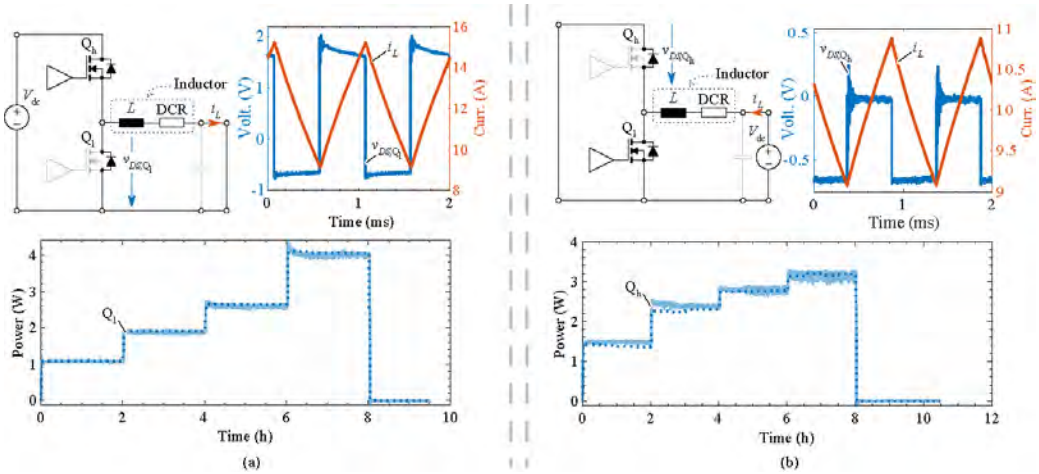


Figure 7.6: Circuit diagram of each of the configurations needed to validate the power semiconductor losses in the (a) low side and (b) high side semiconductors. The figure displays power estimations and measured values for two different test operating points, along with an oscilloscope capture.

after identification. The element (i, j) of matrix $\mathbf{S}$ expresses how sensitive is the power at measurement point i to changes in the temperature at measurement point j

$$[\mathbf{S}]_{i,j} = \frac{\Delta P^i}{\Delta T^j},$$

where ΔP^i and ΔT^j denote an infinitesimal increment in P^i and T^j respectively. From matrix $\mathbf{S}$, as an example, we can see that 1 W of power dissipation on $\mathrm{Q_h}$ produce a 13.5 °C increment on its own temperature and only 0.5 °C on the inductor. Due to their small size, the drivers of the high and low sides are the two main sources of sensitivity to changes in power. On the other hand, it can be observed that the transistors and the PCB tracks are highly coupled to the other parts of the converters, proving that the coupling effects have a major impact in their power losses. It is also seen that the inductor is the most isolated part of the power converter. Our proposed approach can be directly used to conduct this kind of analyses without the need of complex analytical formulations, applicable to any power converter topology and configuration. By identifying a linear discrete-time dynamical system, our approach cannot only be used to analyze the sensitivity of the power converter, but also leverage other control theory tools to analyze the controllability and stability of the power-temperature dynamics. None of this is possible with a black-box model.

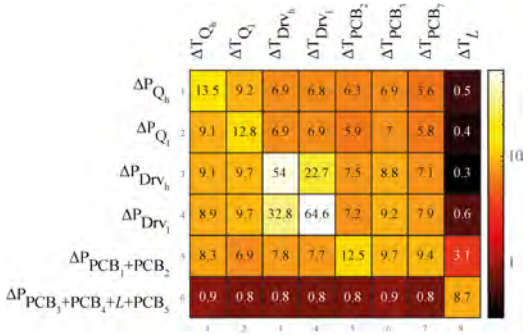


Figure 7.7: Representation of the steady-state sensitivity matrix $\mathbf{S}$ of the identified model for the power converter.

7.1.11 Validation through direct measurements

The previous section has validated the performance in the estimation of semiconductor power losses through the novel automatic thermal modeling technique in static conditions. However, a validation in switching operating conditions is required to show the potential of the technique. Limitations on actual state-of-the-art methods for the switching losses measurements, as calorimetric or invasive current measurements, lead to a lack of a ground-truth model. Therefore, we validate the model with a switching power converter where almost all semiconductor dissipation is due to conduction losses and therefore easy to measure with conventional voltage and current probes. The converter switches at 1 kHz with a bus voltage from 1 V to 1.5 V, so large power losses on the components are produced. Despite having high speed turn ON and OFF on the semiconductors (7 ns), the absence of high frequency switching and reduced bus voltage eliminates measurements complexities, such as possible switching effects from the output capacitance C_{oss}. To do so, we enforce all losses to occur in the internal body diode of the MOSFET, as shown in Fig. 7.6.

The topology uses the internal DC resistance of the inductor (DCR, 21.5 mΩ) and shorts the output to produce a large current bias while keeping one of the semiconductors with zero gate voltage input. Fig. 7.6a shows the configuration where $\mathrm{Q_h}$ switches and $\mathrm{Q_l}$ is used as a diode. As it can be observed from the oscilloscope capture, when $\mathrm{Q_h}$ turns OFF, the current slope becomes negative along with a 12 A bias and the internal $\mathrm{Q_l}$ diode starts its conduction with forward voltage drop of around 0.65 V. This operating point produces large losses that are only a consequence of conduction effects. Due to the large bias current, the inductor is also heated up so the model is fully validated as in a higher frequency point. Voltage variations on the ON state are provoked by a lack of decoupling capacitance; hence, for this low frequency operating point, an external 75 mF capacitance was included. The output capacitors are not relevant as most of the energy on the circuit is DC and not AC. Henceforth, the miss-match between the actual power losses in the device and the ones predictable from the manufacturer data sheets is directly proportional to the forward voltage drop. We measured a forward voltage drop of 0.65 V, whereas the manufacturer provides a value at room temperature of 0.82 V. This non-linear variability in production stands out the need for experimental methods for power semiconductor losses estimation, because if the value of the manufacturer is used instead of the actual one, then we would have an error of $(0.82 - 0.65)/0.82 \times 100 = 26\%$ with respect to the actual power losses. An error of 26% is significant in power converters design and highlights the importance of our approach instead of purely relying on manufacturer's data sheets.

Similar effects take place when driving $\mathrm{Q_l}$ and forcing losses to occur on $\mathrm{Q_h}$. These are shown in Fig. 7.6b where the voltage supply along with its decoupling has been placed on the right side of the schematic. When $\mathrm{Q_l}$ turns OFF the current slope becomes negative and large losses are produced on both the high side device and the inductor. Alternately, when $\mathrm{Q_l}$ turns ON, the device stands no voltage due to the short-circuit.

Fig. 7.6 depicts the estimated and real power values for each of the two configurations for $\mathrm{Q_h}$ and $\mathrm{Q_l}$. For each of the operating points, four oscilloscope captures were processed and used for power representation. Estimates in Fig. 7.6a match the measured values with

errors of less than 1% in steady state. On the other hand, the error in Fig. 7.6b does not exceed the 3%. These results validate the performance of our approach for estimating the semiconductor losses on switching conditions. Moreover, the experiment also validates the superposition principle underpinning our approach, since the inductor, the PCB tracks, and one transistor (low-side in Fig. 7.6a, high-side in Fig. 7.6b) present significant thermal coupling, and still a worst-case accuracy below the 3% of estimation error is achieved.

7.1.12 Comparison with existing post-design approaches

We now compare our proposed method with to other existing a priori approaches. The state-of-the-art in physical-based thermal models either depart from Cauer or Foster models [Ma et al., 2016]. Although Cauer models provide more information about each layer of the thermal path, they are often complex to obtain, requiring FEM simulations. On the other hand, Foster models are more popular on power converters thermal modeling because they are based on the measurement of the temperature dynamics and are independent of the internal structure of the materials. Therefore, for this comparison, we extract a physical-based dynamic thermal model based on a Foster model [Guacci et al., 2020]. The equivalent thermal circuit, shown in Fig. 7.8, contains the heat sources $P_{\mathrm{Q,h}}$ and $P_{\mathrm{Q,l}}$, which denote the power losses in each of the semiconductors; the thermal capacitance $C_{\mathrm{th,hs}}$ of the heat sink connected to $\mathrm{Q_h}$ and $\mathrm{Q_l}$; and the thermal resistance $R_{\mathrm{th,hs\text{-}a}}$ between the heat sink and the ambient; T_{amb} denotes the ambient temperature. In addition, it includes the thermal capacitances $C_{\mathrm{th,h}}$ and $C_{\mathrm{th,l}}$ of the case of $\mathrm{Q_h}$ and $\mathrm{Q_l}$, and the network of resistances formed by $R_{\mathrm{th,h}}$, $R_{\mathrm{th,l}}$ and $R_{\mathrm{th,m}}$. The resistances $R_{\mathrm{th,h}}$ and $R_{\mathrm{th,l}}$ denote the thermal resistance between the heat sink and the case of $\mathrm{Q_h}$ and $\mathrm{Q_l}$, respectively, while $R_{\mathrm{th,m}}$ models the thermal coupling between $\mathrm{Q_h}$ and $\mathrm{Q_l}$.

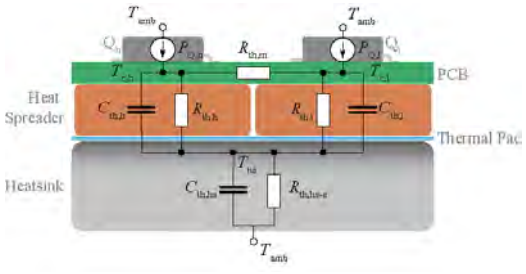


Figure 7.8: Thermal equivalent circuit of the semiconductors power losses. The figure is inspired by [Guacci et al., 2020].

The system of ordinary differential equations describing the evolution of T_{c} is

$$\begin{bmatrix} \dot{T}_{\mathrm{c,h}} \\ \dot{T}_{\mathrm{c,l}} \end{bmatrix} = \begin{bmatrix} -\frac{R_{\mathrm{th,h}}+R_{\mathrm{th,m}}}{C_{\mathrm{th,h}}R_{\mathrm{th,h}}R_{\mathrm{th,m}}} & \frac{1}{C_{\mathrm{th,h}}R_{\mathrm{th,h}}} \\ \frac{1}{C_{\mathrm{th,l}}R_{\mathrm{th,m}}} & -\frac{R_{\mathrm{th,l}}+R_{\mathrm{th,m}}}{C_{\mathrm{th,l}}R_{\mathrm{th,l}}R_{\mathrm{th,m}}} \end{bmatrix} \begin{bmatrix} T_{\mathrm{c,h}} \\ T_{\mathrm{c,l}} \end{bmatrix} + \begin{bmatrix} \frac{P_{\mathrm{Q,h}}}{C_{\mathrm{th,h}}} \\ \frac{P_{\mathrm{Q,l}}}{C_{\mathrm{th,l}}} \end{bmatrix} \tag{7.11}$$

where $R_{\mathrm{th,hs}}$ and $C_{\mathrm{th,hs}}$ are not included, as no heat sink was mounted on our setup. To proceed with the identification of $\mathcal{P} = \{R_{\mathrm{th,h}}, R_{\mathrm{th,l}}, R_{\mathrm{th,m}}, C_{\mathrm{th,h}}, C_{\mathrm{th,l}}\}$, we use a trust region reflective nonlinear optimization method [Li, 1993] under the following cost function

$$\min_{\mathcal{P}} \sum_{k=2}^{K} ||\mathbf{u}(k) - (\mathbf{I} - \Delta t\mathbf{F})\mathbf{u}(k-1) - \mathbf{G}\mathbf{x}(k-1)||_2^2, \tag{7.12}$$

Table 7.1: Parameters of the thermal equivalent circuit.

	Initial 1	**Final 1**	**Initial 2**	**Final 2**
$R_{\mathrm{th,h}}(^\circ\mathrm{C/W})$	3.9	87.1	0.8	3.4
$R_{\mathrm{th,l}}(^\circ\mathrm{C/W})$	3.3	96.0	0.1	1.2
$R_{\mathrm{th,m}}(^\circ\mathrm{C/W})$	30	41.4	0.6	2.9
$C_{\mathrm{th,h}}(\mathrm{J}/^\circ\mathrm{C})$	1.5	3308.7	0.9	15699.0
$C_{\mathrm{th,l}}(\mathrm{J}/^\circ\mathrm{C})$	1.8	393.6	0.9	13936.1

where

$$\mathbf{F} = \begin{bmatrix} -\frac{R_{\mathrm{th,h}}+R_{\mathrm{th,m}}}{C_{\mathrm{th,h}}R_{\mathrm{th,h}}R_{\mathrm{th,m}}} & \frac{1}{C_{\mathrm{th,h}}R_{\mathrm{th,h}}} \\ \frac{1}{C_{\mathrm{th,l}}R_{\mathrm{th,m}}} & -\frac{R_{\mathrm{th,l}}+R_{\mathrm{th,m}}}{C_{\mathrm{th,l}}R_{\mathrm{th,l}}R_{\mathrm{th,m}}} \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \frac{1}{C_{\mathrm{th,h}}} \\ \frac{1}{C_{\mathrm{th,l}}} \end{bmatrix} \tag{7.13}$$

and $\Delta t = 1\,\mathrm{s}$. The expression $(\mathbf{I} - \Delta t\mathbf{F})\mathbf{u}(k-1) + \mathbf{G}\mathbf{x}(k-1)$ comes from the Euler-forward discretization of (7.11). Observe that the cost function of the optimization problem is nonlinear and nonconvex with respect to the parameters to be identified. Fig. 7.8 represents the dynamic equivalent circuit model that leads to (7.11).

Fig. 7.9 compares the identification error $\tilde{\mu}_{\mathrm{RMSE}}$ for our approach and that from [Guacci et al., 2020] by using operating points from saturation regime. It can be seen that both methods achieve the same accuracy after only a few seconds from changing the working point, proving that, despite its generality, ours preserves the identification accuracy of state-of-the-art methods. Significantly, this is done without the need of physical and mathematical modeling, which is a key aspect for more intricate topologies. Besides, there is an additional caveat observed in [Guacci et al., 2020]. Since the cost function is nonlinear and nonconvex in the parameters, the solution obtained by the trust region method is highly dependent on the initialization value. We tried two initialization shown in Table 7.1. In this sense, in Table 7.1 it can be observed that, after optimization, the final values can change dramatically depending on the initial seeds. Therefore, it is hard to extract a physical meaning from the method in [Guacci et al., 2020]. In fact, Foster models do not provide a unique representation of a specific system. In contrast, our method, by proposing a linear method and a linear convex cost function, obtains the unique minimizer of the optimization function and, therefore, there is no dependence on the initialization of the parameters, leading to a unique representation.

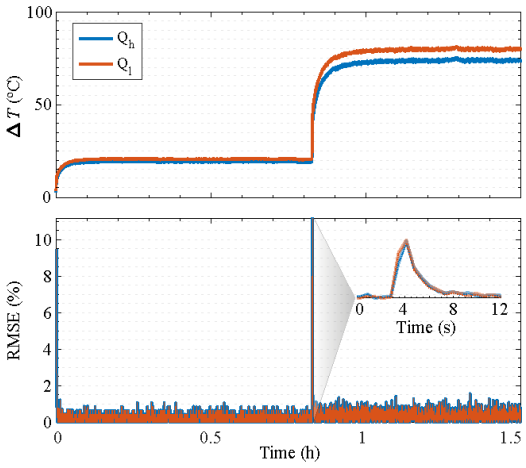


Figure 7.9: Comparison of the equivalent circuit method and the automatic methods. Top figure shows one saturation regime operating point. Bottom figure shows the relative error between the equivalent circuit method and the automatic method.

To conclude, once the power device optimally design and correctly characterized, it is

time to deploy it in a real system. Usually, tracking of the internal factors that affect the operation of the power devices is feasible, since probes and measurement points can be included in design time. However, the operation of the power devices is also influenced by external factors that cannot be predicted a priori, which demands estimation techniques that provide such information in order to develop fault-tolerant power systems. The next section is devoted to this problem.

7.2 Estimation: Online Voltage Prediction Using Gaussian Process Regression for Fault-Tolerant Photovoltaic Applications

7.2.1 Introduction

The progress in technological infrastructures and the awareness about environmental resource consumption are driving a revolution in how energy is supplied [Vazquez et al., 2010]. There is a need of not only providing stable energy but also accomplishing efficiency requirements and prediction capabilities to anticipate issues. The energy is not consumed at the same rate at which is generated so, e.g., robotic systems, need an Energy Storage System (ESS) to provide a stable power supply. This stability is subject to unpredictable meteorology phenomena, turning the efficient autonomy into a challenge. The solution involves designing an intelligent system capable to predict future low power generation scenarios, preserving the infrastructures typically coordinated by low-cost embedded systems. In this sense, data-driven techniques are a must. Among others, GPR appears as a promising tool [Quinonero-Candela and Rasmussen, 2005] because of the absence of overfitting and the measurement of the uncertainty of the prediction. However, GPR requires a large computational effort, limiting its use in real-time applications.

The problem of feature prediction in ESSs based on Battery Management Systems (BMSs) is a motivating topic. For instance, [El Mejdoubi et al., 2015] works in predicting both State of Charge (SoC) and State of Health (SoH) by an Extended Kalman Filter (EKF) and an adaptive observer. A Probabilistic Neural Network is presented in [Lin et al., 2012] to predict the SoH. Following with the learning tools, [Hu et al., 2014] proposes a Support Vector Machine (SVM) model for regression. Nevertheless, due to the aforementioned advantages, GPR is a cornerstone in the state-of-the-art discussion. The authors in [Sahinoglu et al., 2017] apply Regular GPR to the prediction of the State-of-Charge (SoC) of a single battery. A similar approach is presented in [Pajovic et al., 2017], focused on the voltage prediction in a certain horizon applying a recursive GPR. Another interesting parameter to predict in BMSs is the SoH [Liu et al., 2012] and the Remaining Useful Live (RUL). In this regard, recent works like [Cong et al., 2020,Liu and Chen, 2019] explore the use of GPR to determine the aging of Lithium-ion batteries and control their SoH under different life-cycle. Furthermore, the authors of [Liu et al., 2019a,Liu et al., 2020,Liu et al., 2019b] have proved the convenience of GPR techniques to predict the health of Lithium-ion batteries under uncertain scenarios, modifying the structure of the underlining kernels to

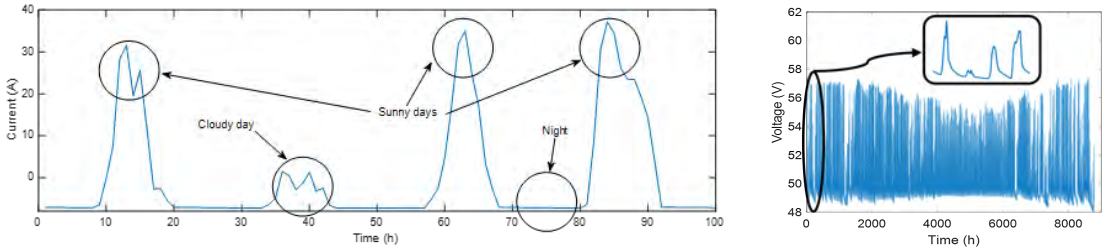
better consider empirical and electrochemical knowledge. Despite the great promise of the aforementioned instances, GPR is typically studied with artificial and/or informal datasets.

7.2.2 Dataset development

We first present the real data set used, its main features and the context in which it has been obtained. The dataset comes from a photovoltaic system powered by 24 lead-acid batteries (VRLA, Exide Classic Solar 2V 4600Ah C120) with nominal voltage of 2V in series and for a total nominal voltage of 48 and a capacity of 4600Ah. The system has 24 solar panels (Scheuten multi 180P6), with 4680W of rated power. The equipment of the system has a typical consumption of 350W, with occasional peaks up to 550W.

The control system monitors the current, voltage and temperature waveforms of the batteries. Photovoltaic systems features irregular generation profiles that depend on weather and sun irradiation. They have three main generation-consumption dynamics: day-night cycles, cloudy periods (days or weeks), and the annual dynamic. Therefore, the reliability of the system is subject to significant uncertainties. In this application in particular, the prediction of a short-term system failure is especially interesting due to the expensive actions needed to restart the system. These critical system failures are mainly related to insufficient charge in the battery and could be detected by predicting an excessive low voltage level in the battery. A GPR based voltage predictor is a suitable option for this task, since not only provides the prediction, but it also indicates the confidence interval of the prediction [Quinonero-Candela and Rasmussen, 2005]. Data has been recorded during 2 years with a sampling rate of one sample every hour. Voltage, current and temperature waveforms are the information available and, as they are the typical variables in this type of system, this work can be easily extended to similar systems. Fig. 7.10 shows examples of current waveforms recorded in different weather scenarios, manifesting clear differences. According to the specifications of the client, i.e., the CHE, a failure prediction of 48 hours ahead is sufficient to avoid a critical failure, which is tackled by increasing the battery charge voltage level (floating voltage). As the consumption of the system is nearly constant, the critical moment is the moment just before the sunrise, where the amount of charge is the lowest of the cycle. Our data set is different from others due to the amount of real data available. In this sense, it is noteworthy that there are two different phenomena in the dataset. The first is the periodical dynamics, which are predictable with an accurate model of a battery. The other is the random events associated with weather from which, due to its inherent quasi-chaotic nature, it is difficult to predict its impact in the batteries. The purely periodical dynamics are related to several phenomena: day/night, yearly and season-based dynamics. Thus, to account for all of them, the training set has been selected according to the greatest period, using a complete year of data. Thanks to the size of the dataset, tests are performed with data from the next year.

Fig. 7.10 illustrates the voltage waveforms recorded in the system in 2008, from where it is clear that the floating voltage of the batteries is not constant. This is due to the temperature-regulated charge strategy of the solar regulator. Thus, the information encoded in that strategy provides important knowledge to our problem.



(a) (b)

Figure 7.10: (a) Example of the current waveforms recorded in different weather scenarios. (b) Voltage waveforms from 2008.

To end up this section, we introduce the metrics we will use to evaluate and study the different methods and hyperparameter selections. To evaluate the performance of the models, we choose the Root Mean Square Error (RMSE) and the Maximum Absolute Error (MaxAE) as metrics, defining them as

$$\mathsf{RMSE}(l) = \sqrt{\frac{1}{\mathsf{N}_{\text{test}}} \sum_{t=1}^{\mathsf{N}_{\text{test}}} (V_{t+l} - \hat{V}_{t+l})^2}, \tag{7.14}$$

$$\mathsf{MaxAE}(l) = \max_{i=1,\ldots,\mathsf{N}_{\text{test}}} |V_{t+l} - \hat{V}_{t+l}|, \tag{7.15}$$

where N_{test} refers to the number of test samples and $l = 1, \ldots, \mathsf{M}$, where $\mathsf{M} = 48$ hours for this work. In addition, we present two additional metrics which refer to RMSE and MaxAE at the end of the night

$$\mathsf{RMSE}_{\text{EON}}(l) = \sqrt{\frac{1}{\mathsf{N}_{\text{test}}} \sum_{t=1}^{\mathsf{N}_{\text{test}}} (V_{t+l}^{\text{EON}} - \hat{V}_{t+l}^{\text{EON}})^2}, \tag{7.16}$$

$$\mathsf{MaxAE}_{\text{EON}}(l) = \max_{i=1,\ldots,\mathsf{N}_{\text{test}}} |V_{t+l}^{\text{EON}} - \hat{V}_{t+l}^{\text{EON}}|, \tag{7.17}$$

This value of voltage is crucial in photovoltaic systems as it gives the BMS the information of the voltage of the system if no more solar energy can be absorbed.

7.2.3 GPR models for large datasets

Let $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ denote a training dataset, comprising n $D-$dimensional input and scalar output pairs $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^D$, and the corresponding outputs $\mathbf{y} = \{y_i\}_{i=1}^n$, where $y_i \in \mathbb{R}$. It is assumed that there is an underlying nonlinear latent function $f(\bullet)$, which can be used to parameterize the probabilistic mapping between inputs and outputs

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \tag{7.18}$$

where ϵ_i denotes zero-mean additive Gaussian noise with variance σ_i^2. The GPR main hypothesis relies on assuming a priori that function values behave according to a multivariate Gaussian distribution

$$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{f},\mathbf{f}}), \tag{7.19}$$

where, $\mathbf{f} = [f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)]^\top$ is a vector of latent function values, and $\mathbf{0}$ is an $n \times 1$ vector whose elements are all 0. In addition, $\mathbf{K}_{\mathbf{f},\mathbf{f}}$ is a covariance matrix, whose entries are given by the covariance function $\mathbf{K}_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$, from now on named kernel function, evaluated at each pair of training inputs.

In GPR, the kernel function plays a major role, since it encodes the prior assumptions about the properties of the underlying latent function that we are trying to model.

Indeed, one of the tasks of the designer is to select the kernel which best fits the phenomena modelled. An instance of kernel typically adopted is the Squared Exponential (SE) kernel, defined as

$$k_s(\mathbf{x}_i, \mathbf{x}_j) = \varrho_0^2 \exp\left(-\frac{1}{2}\sum_{d=1}^{D}\left(\frac{x_{id} - x_{jd}}{l_d}\right)\right), \tag{7.20}$$

where x_{id} and x_{jd} correspond to the $d-$th element of vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively, and $\Theta = [\varrho_0, l_1, \ldots, l_D]^\top$ denotes the hyperparameters. Distinctly, ϱ_0^2 denotes the signal variance and quantifies the variation of the underlying latent function from its mean, and l_D represents the characteristic length scale for each input dimension. Finally, l_D fixes the width of the kernel and thereby represents how smooth the functions in the model are. In addition to the aforementioned kernel function, we incorporate the additive Gaussian white noise term into the selected kernel function

$$k(\mathbf{x}_i, \mathbf{x}_j) = k_s(\mathbf{x}_i, \mathbf{x}_j) + \sigma_i^2 \delta_{ij}, \tag{7.21}$$

where δ_{ij} denotes the Kronecker delta, which takes value of 1 if $i = j$ and 0 otherwise. Then, the distribution of $\mathbf{y}$, conditioned on the latent function values $\mathbf{f}$ and the input $\mathbf{X}$, is given by

$$p(\mathbf{y}|\mathbf{f}, \mathbf{X}) = \mathcal{N}(\mathbf{f}, \sigma_i^2 \mathbf{I}), \tag{7.22}$$

where $\mathbf{I}$ is an $n \times n$ identity matrix. Throughout integration over the latent function values $\mathbf{f}$ and by using (7.19) and (7.22) the marginal distribution of $\mathbf{y}$ can be obtained:

$$p(\mathbf{y}|\mathbf{X}, \Theta) = \int p(\mathbf{y}|\mathbf{f}, \mathbf{X}) p(\mathbf{f}|\mathbf{X}) df = \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{f},\mathbf{f}}) + \sigma_i^2 \mathbf{I}. \tag{7.23}$$

Then, the marginal log-likelihood, which refers to the marginalization over the function values $\mathbf{y}$ can be written as

$$\log p(\mathbf{y}|\mathbf{X}, \Theta) = \frac{-1}{2}\mathbf{y}^\top (\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma_i^2 \mathbf{I})^{-1} - \frac{1}{2}\log|\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma_i^2 \mathbf{I}| - \frac{n}{2}\log 2\pi, \tag{7.24}$$

where $|\bullet|$ is the determinant of a matrix. The optimum hyperparameters are found by maximizing the marginal log-likelihood. To do so, we use the partial derivatives of the marginal log-likelihood, from where we obtain that

$$\frac{\partial}{\partial \theta_i}\log p(\mathbf{y}|\mathbf{X}, \Theta) = -\frac{1}{2}\mathrm{tr}\left((\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma_i^2 \mathbf{I})^{-1}\frac{\partial(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma_i^2 \mathbf{I})}{\partial \theta_i} + \frac{1}{2}\mathbf{y}^\top \sigma_i^{-2}\mathbf{I}\frac{\partial(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma_i^2 \mathbf{I})}{\partial \theta_i}(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma_i^2 \mathbf{I})^{-1}\mathbf{y}. \tag{7.25}$$

It is important to notice that the complexity of computing (7.24) is dominated by the inversion of $\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma_i^2 \mathbf{I}$, which requires a computational time of $O(n^3)$. Hence, a simple implementation of the GPR is advisable for datasets with up to a thousand training examples.

The characterization in (7.25) allows the use of any gradient based optimization method to optimize the marginal log-likelihood function (7.24). It is important to note that, generally, objective functions are nonconvex with respect to the hyperparameters which can lead to converge to a local optimum. In order to tackle this problem, gradient-based optimization can be performed with different initial points, and the optimal hyperparameters that yield the largest marginal log-likelihood can be chosen. Once we obtain the optimal hyperparameters, it is possible to express the joint distribution of the training outputs $\mathbf{y}$ and the test output $\mathbf{y}_*$ as

$$p(\mathbf{y}, \mathbf{y}_*|\mathbf{X}, \mathbf{x}_*, \Theta) = \mathcal{N}\left(\begin{bmatrix}\mathbf{0}\\ 0\end{bmatrix}, \begin{bmatrix}\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma_i^2\mathbf{I} & \mathbf{K}_{*,\mathbf{f}}\\ \mathbf{K}_{\mathbf{f},*} & \mathbf{K}_{**} + \sigma_i^2\end{bmatrix}\right), \tag{7.26}$$

where the asterisk $*$ is used as a shorthand for $\mathbf{f}_*$, which is the corresponding latent function value at the test input, $\mathbf{K}_{*,\mathbf{f}} = [k(\mathbf{x}_1, \mathbf{x}_*), \ldots, k(\mathbf{x}_n, \mathbf{x}_*)]^\top$ and $\mathbf{K}_{**} = k_s(\mathbf{x}_*, \mathbf{x}_*)$. The target of the GPR is to find the predictive distribution of the test output $\mathbf{y}_*$ which are conditioned on both the dataset $\mathcal{D}$ and test input $\mathbf{x}_*$. Thus, it is possible to marginalize the joint distribution (7.26) over the training dataset output $\mathbf{y}$ and to obtain that the predictive distribution of the test output, $\mathbf{y}_*$, is Gaussian distributed, i.e.,

$$p(\mathbf{y}_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*, \Theta) = \mathcal{N}(\mu_*, \Sigma_*), \tag{7.27}$$

where the mean and the covariance of the predictive distribution are given, respectively, in the following

$$\mu_* = \mathbf{K}_{*,\mathbf{f}}(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma_i^2\mathbf{I})^{-1}\mathbf{y}, \tag{7.28}$$

$$\Sigma_* = \sigma_i^2 + \mathbf{K}_{**} - \mathbf{K}_{*,\mathbf{f}}(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma_i^2\mathbf{I})^{-1}\mathbf{K}_{\mathbf{f},*}. \tag{7.29}$$

This mean, which is effectively the point estimate of the test output, is obtained as a linear combination of the noisy dataset outputs $\mathbf{y}$. Also, the variance of the predictive distribution, Σ_* in (7.29), serves as a measure of the uncertainty in the estimate of the test output. After performing the inversion of $\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma_i^2\mathbf{I}$, the computational complexity of the testing stage is $O(n)$ and $O(n^2)$ for the mean μ_* and the variance Σ_* respectively, which makes the proposed method highly appropriate for online operation.

The output of the GPR model is an estimation of the future sample of voltage data, V_{k+1}, predicted from $L+1$ past samples of voltage $V_k, \ldots, V_{k-L}$, current $I_k, \ldots, I_{k-L}$, and temperature $T_k, \ldots, T_{k-L}$; and future current I_{k+1} and temperature T_{k+1} samples. Thus, L is the memory length of the number of previous samples processed. Since the power demand of the system is known, we can feed the GPR models with future current and temperature waveforms. This data is assembled in a vector $\mathbf{x}_{*k}$, of length $3(L+1)+2$, formatted as:

$$\mathbf{x}_{*k} = [I_{k+1}, T_{k+1}, V_k, I_k, T_k, \ldots, V_{k-L}, I_{k-L}]^\top. \tag{7.30}$$

The output y_k of the algorithm is the voltage at the sample $k+1$

$$y_k = V_{k+1}. \tag{7.31}$$

GPR techniques consist in two processes. In the first part, GPR is trained offline to learn the relationship between the inputs $\mathbf{x}_{k_i}$ and outputs $\mathbf{y}$. Then, optimal values of

the hyperparameters of the chosen kernel are determined through a conjugate gradient method based on a training dataset, $\mathcal{D}$. In the second part, online voltage prediction of the battery is performed based on voltage, current, and temperature measurements of the battery. The mean μ_{*k} of the predicted distribution represents the estimated voltage V_{k+1}. Additionally, the variance of the predicted distribution gives us a measure of the confidence of the prediction. We consider a confidence interval to the 95%

$$[V_{k+1} - 1.96\Sigma_{*k}^{1/2}, V_{k+1} + 1.96\Sigma_{*k}^{1/2}]. \tag{7.32}$$

As variance decreases, the confidence interval becomes smaller and it indicates a more accurate prediction. GPR techniques can be used to directly provide the prediction of a variable one sample ahead.

7.2.4 Regular GPR

The scope of this method is to obtain the estimated voltage corresponding to $k + z$ time, until reaching the desired $\mathsf{M} = 48$ hours prediction for our purposes. This means that the predicted output of the method corresponds to the expected voltage at time instant $k + z + 1$. Due to the recursive structure, in the first step, what it is estimated is the mean $\mu_{*k}^{(1)}$ calculated trough (7.28). This value is then fed back along with voltage, current and temperature at time instants $k, k-1, \ldots, k-L-1$, and current and temperature $I_{k+1}, T_{k+1}, I_{k+2}, T_{k+2}$ in a one-step prediction algorithm to obtain $\mu_{*k}^{(2)}$. This process is repeated until we obtain the 48 step-ahead voltage prediction. In the testing stage vectors are formatted as

$$\mathbf{x}_{*k}^{(z)} = [\mathbf{I}_{k+m} \quad T_{k+m} \quad \mathbf{V}_k^{(z)} \quad \mathbf{I}_k^{(z)} \quad \mathbf{T}_k^{(z)}]^\top, \tag{7.33}$$

with

$$\mathbf{V}_k^{(z)} = [\mu_{*k}^{(z-1)} \ldots \mu_{*k}^{(1)} \quad V_k \ldots V_{k-L+z-1}]^\top \tag{7.34}$$

$$\mathbf{I}_k^{(z)} = [I_{k+m-1} \ldots I_{k+1} I_k \ldots I_{k-L+z-1}]^\top \tag{7.35}$$

$$\mathbf{T}_k^{(z)} = [T_{k+m-1} \ldots T_{k+1} T_k \ldots T_{k-L+z-1}]^\top \tag{7.36}$$

The GPR model is trained with 720 samples of data, i.e., a complete month. To capture all kind of phenomena, the 720 samples comes from an equally spaced selection of the original training dataset (the complete year of 2008). In each selection, a complete day is captured in order to preserve the temporary properties and waveform features, so the training set can be seen as a cascade of days from different seasons and weather conditions. For testing, the complete month of March 2009 has been used, selected due to the fact that it involves all kinds of uncertain phenomena (both sunny and cloudy periods) since it belongs to a low irradiation season.

The selection of the kernel used in the GPR model is required. Conceptually, the kernel is the tool that captures the relationships with the training data. In this application, it is especially difficult to detect these hidden relations because of the 48 hours recurrent approach. In non-recurrent previous related works [Sahinoglu et al., 2017, Pajovic et al., 2017] SE kernel exhibits good performances. However, in this work we make an extensive

	RQard	SEard	SEard + SEard + NN [Pajovic et al., 2017]	Matenard
RMSE(V)	0.592	3.315	1.699	3.071
MaxAE(V)	1.596	1.635	1.919	2.045
Train time (s)	13.2	13.4	31.6	10.7
Test tie (s)	51.0	50.1	369.2	67.6

Table 7.2: Comparative of kernels using Regular GPR.

test of different kernels, as proposed in [Duvenaud, 2014]. According to the results over the RMSE and MaxAE, the kernel that gives a better performance for our dataset is the Rational Quadratic (RQ) [Rasmussen and Nickisch, 2010]

$$k_s(\mathbf{x}_i, \mathbf{x}_j) = \varrho_1^2 \left(1 + \frac{1}{2\alpha} \sum_{d=1}^{D} \left(\frac{x_{id} - x_{jd}}{\mu_d}\right)^2\right)^{-\beta} \tag{7.37}$$

where ϱ_1^2 is the signal variance, β is a smoothness parameter and μ_d is the characteristic length scale. Table 7.2 shows the best results of the tests, where it can be appreciated that more complex kernels do not improve the performance whilst computational cost is increased. In all cases, Automatic Relevance Determination is used [Wipf and Nagarajan, 2007].

7.2.5 GPR based multiple experts

Regular GPR considers equally all the patterns, irrespective of classes. The idea now is to model different types of canonical current waveforms (related to specific daily dynamics during a year), or classes. Several GPR-based "experts" are designed, each one related to a class. A non-supervised clustering algorithm is used for establishing the different areas of expertise of each "GPR expert". Self-Organizing Maps (SOM) [Kohonen, 1997] is the selected technique that makes the non-linear projection of a high dimensional input space (dataset) to a two-dimensional map processing element (neurons). After the SOM training, every input pattern (each day) is projected into five classes:

- Voltage at the beginning of the discharge.
- Voltage at the end of the discharge (at the sunset of the day).
- Duration in hours of the charging process.
- Hours of the discharge.
- Charge stored.

After training a 10×6 SOM by using the SOM Toolbox [Vesanto et al., 1999] library, the well-known K-means clustering algorithm [Wagstaff et al., 2001] has been applied to the outputs of the SOM to establish the different classes obtained. K-means needs the number of clusters to make the separation, so for establishing this number of clusters the algorithm

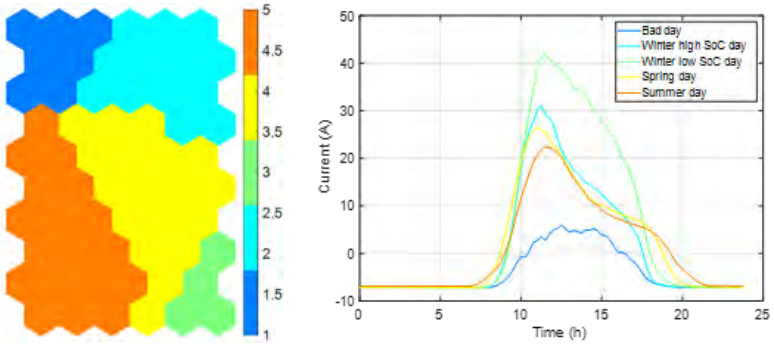


Figure 7.11: (left) Results of K-means applied to SOM. (right) Mean waveforms of the five classes of days from SOM.

has been applied several times changing this parameter. The results show that using more than five classes does not provide relevant information. Fig. 7.11a is the final result of this process. Each sample is labelled regarding its characteristics in the following classes: low irradiation days, winter days with high SoC, winter days with low SoC, spring days, and summer days. For a better understanding of the differences among the type of days, Fig. 7.11b, shows the mean current waveform of each data related to each type of day. With this tool all the days of the original training dataset are classified and tagged. Afterwards, five GPR experts are designed and trained, using for training days of their corresponding class. The same parameters obtained with Regular GPR (train size, test set, memory length of 15 hours, RQ kernel) are used in the five experts. In the inference stage, after each expert estimates the future sample, one of the experts is chosen and its output is fed back for the next iteration. As the confidence interval calculated by the GPR provides information about the accuracy of the estimation, the expert with the most accurate confidence interval is selected as the one to provide the final output.

7.2.6 Sparse GPR

Sparse GPR [Quinonero-Candela and Rasmussen, 2005] is a modification of the Regular GPR that enables large dataset processing. For larger datasets, sparsity solves the problem of the computational cost. The computational cost of a regular GPR is reduced by introducing inducing variables and modifying the joint prior distribution, $p(f_*, \mathbf{f})$. Let $\mathbf{u} = [u_0, ..., u_m]^\top$ denote the inducing variables which correspond to a set of input locations X_u called inducing points, where $m \leq n$ is the number of inducing points. The inducing points are chosen as a subset of the data points. Given the inducing points, the joint prior distribution, $p(f_*, \mathbf{f})$ can be rewritten as

$$p(f_*, \mathbf{f}) = \int p(f_*, \mathbf{f}|\mathbf{u})p(\mathbf{u})d\mathbf{u}, (25) \tag{7.38}$$

where $p(\mathbf{u}) = \mathcal{N}(0, \mathbf{K}_{\mathbf{u},\mathbf{u}})$. For the approximation of $p(f_*, \mathbf{f})$, it is assumed that f_* and $\mathbf{f}$ are conditionally independent given u in the following [Quinonero-Candela and Rasmussen, 2005]

$$p(f_*, \mathbf{f}) \approx q(f_*, \mathbf{f}) = \int q(f_*|\mathbf{u})q(\mathbf{f}|\mathbf{u})p(\mathbf{u})d\mathbf{u}. \tag{7.39}$$

	Storage	Training	Mean	Variance
Regular	$O(n^2)$	$O(n^3)$	$O(n)$	$O(n^2)$
FITC	$O(mn)$	$O(m^2n)$	$O(m)$	$O(m^2)$

Table 7.3: Computational cost of Regular and FITC GPR.

Besides, it is assumed that the training conditional $q(\mathbf{f}|\mathbf{u})$ is fully independent and the test conditional keeps exact as

$$q(\mathbf{f}|\mathbf{u}) = \Pi_{i=1}^{n} p(f_n|\mathbf{u}) = \mathcal{N}(\mathbf{K_{f,u}}, \mathbf{K_{u,u}^{-1}}\mathbf{u}, \mathrm{diag}(\mathbf{K_{f,f}} - \mathbf{Q_{f,f}})), \tag{7.40}$$

$$q(f_*|\mathbf{u}) = p(f_*|\mathbf{u}), \tag{7.41}$$

where $\mathrm{diag}(A)$ denotes the diagonal matrix in which the entries outside the main diagonal are all zero. The distributions in (7.40) and(7.41) can be substituted in (7.39), so that integrating over $\mathbf{u}$ gives the the joint prior

$$q(\mathbf{f}, f_*) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{Q_{f,f}} - \mathrm{diag}(\mathbf{Q_{f,f}} - \mathbf{K_{f,f}}) & \mathbf{Q_{f,*}} \\ \mathbf{Q_{*,f}} & \mathbf{K_{*,*}} \end{bmatrix}\right) \tag{7.42}$$

where $\mathbf{Q_{a,b}} = \mathbf{K_{a,u}}\mathbf{K_{u,u}^{-1}}\mathbf{K_{u,b}}$ is a low-rank matrix (i.e., rank m). The predictive distribution can be obtained by using the above joint prior distribution,

$$q(\mathbf{y}_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*, \Theta) = \mathcal{N}(\tilde{\mu}_*, \tilde{\Sigma}_*), \tag{7.43}$$

where

$$\tilde{\mu}_* = \mathbf{K_{*,u}}\Omega\mathbf{K_{u,f}}\Lambda^{-1}\mathbf{y}, \tag{7.44}$$

$$\tilde{\Sigma}_* = \sigma_i^2 + \mathbf{K_{*,*}} - \mathbf{Q_{*,*}} + \mathbf{K_{*,u}}\Omega\mathbf{K_{u,*}}. \tag{7.45}$$

Here, $\Omega = (\mathbf{K_{u,u}} + \mathbf{K_{u,f}}\Lambda^{-1}\mathbf{K_{f,u}})^{-1}$ and $\Lambda = \mathrm{diag}([\mathbf{K_{f,f}} - \mathbf{Q_{f,f}} + \sigma_i^2\mathbf{I}])$. Notice that the only matrix requiring inversion is the $n \times n$ diagonal matrix Λ, which translates into a significant reduction in computational complexity. The computational cost of training becomes $O(m^2n)$, i.e., linear in n and quadratic in m. Larger values of m should yield to better accuracy at the expense of an increase in the computational requirements. Additionally, the complexity of the testing stages is $O(m)$ and $O(m^2)$ for calculating the mean and the variance, respectively. Among many different kinds of sparsity algorithms [Quinonero-Candela and Rasmussen, 2005], Fully Independent Training Conditional (FITC) seems to get the best performance [Chalupka et al., 2013] and, therefore, it is the one used in this work. Table 7.3 shows a comparison of computational cost between Regular GPR and FITC Sparse GPR.

7.2.7 Results and discussion

We validate the proposed models in the experimental dataset. The experiments are implemented in MATLAB2020a in a 1.8-GHz Intel Core i7-10510U CPU. In order to test the performance of the three proposed models, it is first necessary to see the relevance of the selection of the input variables. By this, we study which variables (voltage, current and temperature) are relevant to predict future voltage. There are four possibilities

	I and T	I only	T only
RMSE(V)	0.592	0.590	2.283
MaxAE(V)	1.596	1.603	5.700
RMSE_{EON}(V)	0.225	0.228	0.279
$\text{MaxAE}_{\text{EON}}$(V)	0.739	0.728	1.539

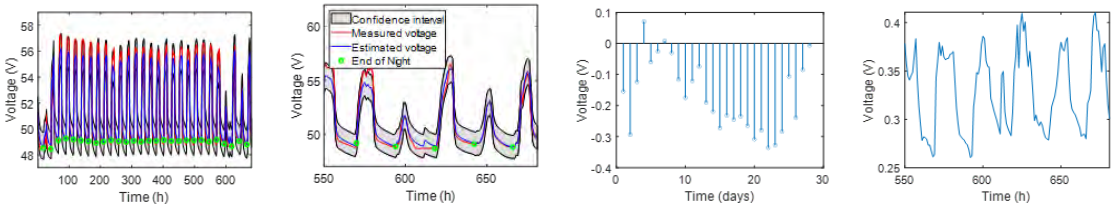
Table 7.4: Error metrics as a function of the inputs considered in the models.

- Future current and temperature profiles known.
- Only future current profile known.
- Only future temperature profile known.
- Neither current nor temperature known (not possible in the context of this problem).

In all cases past profiles are known. To test the influence of the input parameters in the prediction we use the Regular GPR, as long as the results are expandable to the other models. The resulting error metrics are listed in Table 7.4. The results show that temperature has almost no influence on voltage prediction and therefore it can be omitted in future analysis. This is related to the high thermal inertia of the battery pack, since they contain tons of liters of water. Sudden irradiation changes caused by a cloud are not enough to vary relevantly the temperature. In practice, this result helps to reduce the complexity of the system since the matrix dimensions get reduced. As the second method in Table 7.4 gets similar error metrics to the first method but with less measures, it is chosen to test the models. Once the future input profile has been chosen, we discuss the performance of the three different algorithms. In contrast with Regular and ME GPR, as Sparse GPR can deal with larger datasets, it is trained with the entire 2008 year. In order to keep a fair comparison between the algorithms, they all are tested with March 2009.

To illustrate the performance of the three methods, in Fig. 7.12 we show the waveforms obtained with the Regular GPR implementation. The model fits the real waveform accurately even in the isolation days. These results manifest that there exists a periodic error located at midday which matches with the highest value of the confidence interval, and therefore of standard deviation, of the day. This is because, at this point of the day, any interfering cloud may reduce the irradiation on the panels and, thus, there is a higher uncertainty. The error at the end of the night is low in all cases. This is an intuitive result since clouds do not really have an impact on the irradiation at night (which is nonexistent). As the waveforms from the three GPR models are similar, Table 7.5 provides quantitative information.

All methods perform, to a greater or lesser extent, a proper voltage estimation in a month with heterogeneous events. ME GPR improves the RMSE of Regular GPR but Sparse GPR gets the better performance of all methods when it comes to this metric. Besides, Regular GPR has the lower MaxAE so it seems to be the most reliable option when the system needs accurate voltage estimation. In the photovoltaic systems field, where the EON voltage estimation has a strong interest, all methods reduce RMSE and MaxAE values



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

Figure 7.12: Results of the prediction with Regular GPR and current as the only known future value: (a) waveforms over the month test, (b) zoom to the crucial days of the month, (c) $\mathsf{RMSE}_{\mathrm{EON}}$ obtained in the month test, (d) Σ obtained in the crucial days of the month.

	Regular	ME	Sparse
RMSE(V)	0.590	0.517	0.469
MaxAE(V)	1.603	2.751	2.138
$\mathsf{RMSE}_{\mathrm{EON}}$(V)	0.228	0.190	0.238
$\mathsf{MaxAE}_{\mathrm{EON}}$(V)	0.728	0.308	0.127

Table 7.5: Error metrics with only future current profile as input, for a battery with a total nominal voltage of 48V.

with a noteworthy performance of ME and Sparse. We remark that using GPR for prediction (extrapolation) in photovoltaic applications represents a more demanding task than in the typical regression problem (interpolation) because of the uncertainty of the results. As means of comparison, we apply the same approach to two non GPR related algorithms in this application: Nonlinear Autoregressive Exogenous Model (NARX) networks [Chen et al., 1990] and Long Short-Term Memory (LSTM) [Graves and Schmidhuber, 2005], both well-known techniques to process and predict real time data series. The tests are carried with the same conditions of GPR: the same year for training, the same month for testing, same L and failure prediction of 48 hours ahead. Results are shown in Table 7.6.

Results are rather similar, which confirms the usefulness of GPR as a prediction tool. Only NARX shows slightly better results predicting EON situations (40mV better than GPR) but, in contrast, it has higher MaxAE figures (1.6V worse than GPR). Train and test times information are not especially important because they are only related to the design

	ME	Sparse	NARX	LSTM
RMSE(V)	0.517	0.468	0.469	0.454
MaxAE(V)	2.751	2.138	3.741	3.861
$\mathsf{RMSE}_{\mathrm{EON}}$(V)	0.190	0.238	0.095	0.220
$\mathsf{MaxAE}_{\mathrm{EON}}$(V)	0.308	0.127	0.083	0.197
Train time (s)	60	24	55.7	5437
Test time (s)	416	47	0.041	2.7
Execution time (ms)	610	69	0.06	4

Table 7.6: Error metrics of GPR techniques vs NARX and LSTM for a battery with a total nominal voltage of 48V.

phase of the algorithm. Execution time (Execution Time), which is the time needed to make one single prediction of a sample, is more important regarding to implementation. However, as a single prediction is carried out each hour, there is enough time for each execution. For example, the most time-consuming technique (ME) spends only 0.61 seconds. Besides, it is important to note that GPR algorithms have an advantage over other techniques, that is, it provides the confidence interval. This measure of the algorithm accuracy enhances the robustness and allows to design more sophisticated fault-tolerant protocols.

7.3 Control: Nonlinear Implementable Control of a Dual Active Bridge Series Resonant Converter

7.3.1 Introduction

Power electronics is present in many of today's technological advances. In particular, bi-directional DC/DC resonant converters are an efficient option when high voltage and power conversion ratios are required [Pavlović et al., 2012], enabling applications with fast and changing dynamics [Zhao et al., 2017] such as autonomous robot systems. Another application example can be found on automated guided vehicles (AGV), where a DC/DC converter links the low voltage (48V) battery with the industrial level voltage (600V) DC bus of the inverter. Due to the small value of the DC bus capacitor, fast load steps lead to deep voltage variations at the bus capacitor. The DC/DC converter must exhibit a fast control response. Similar conclusions can be derived from other applications such as renewable energy stations [Baroudi et al., 2007] or the electric car [Tavakoli and Pantic, 2018]. A popular topology in this kind of applications is the Dual Active Bridge (DAB) [De Doncker et al., 1991, Kheraluwala et al., 1992] due to its low stress and small filter components. When low switching losses are required, the best option is to include a Series Resonant Converter (SRC) [Vorperian and Cuk, 1982, Oruganti and Lee, 1985a, Erickson and Maksimovic, 2007]. All the aforementioned features make the control of the DAB-SRC [Krismer and Kolar, 2009, Li and Bhat, 2009] specially challenging, evidenced in the absence of advanced controllers on the DAB-SRC in the literature.

A common practice in the control of complex converters is the use of nested feedback loops [Vasca and Iannelli, 2012] based on linear techniques. This is the case of [Seltzer et al., 2011] and [Seltzer et al., 2012] for the DAB. Indeed, the use of linear regulators is the standard in several industrial applications [Engel et al., 2012] because they are simple to tune and enough for a stable and robust performance, at the expense of slowing down the system to reject perturbations and nonlinear effects. The resulting poor transient dynamic is not a problem in many applications, e.g., when charging batteries. However, there are other applications where the transient response of the DC/DC converter is critical, as it is the case of regulating the voltage of the DC bus capacitor of an inverter. In this case, the DC/DC converter must react rapidly in order to avoid strong voltage variations at the DC bus capacitor. A DAB converter with a PI controller exhibits a fast response but in the case of a DAB-SRC converter, its strong nonlinearities and resonance behavior makes almost impossible to get a fast response based on linear converters.

In the search of faster and more robust control solutions, several works employ nonlinear control techniques. An instance is [Zhang et al., 2015], where robust control is used to ensure the stability of a Buck against uncertain parameters. Lyapunov's stability theory is the base of [Meza et al., 2012] to develop a passive, adaptive control in a single-phase inverter. Compared to them, we deal with a more complex topology due to the DAB topology and the SRC circuit. A well-known technique to regulate power converters is the Sliding Mode Control [Slotine et al., 1991], approach taken by [Jeung and Lee, 2018] in a DAB converter. Nevertheless, this approach requires high control frequency and precise knowledge of the parameters to perfectly compensate the nonlinear dynamics. There are two other control alternatives with interesting properties. The first one is the deadbeat control [Saggini et al., 2007, Cortés et al., 2008, Rodríguez and Cortés, 2012], which obtains the fastest response in simple topologies [Saggini et al., 2007] and can be integrated in optimal controllers [Cortés et al., 2008]. The second alternative is State Trajectory Control (STC) [Oruganti and Lee, 1985b, Oruganti et al., 1988]. The seminal work in [Feng et al., 2012] proposes a simplification which enables its implementation in an *LLC* converter. However, both methodologies, deadbeat and STC, require the control frequency to be similar to the switching frequency, which in the converter used in our experiments is not possible because the control frequency is 15 to 40 times slower than the switching frequency.

7.3.2 Model description

The converter under study is shown in Fig. 7.13a. There are two full bridges interconnected by an LC series resonant tank and a transformer. In this case V_a is the high voltage side and V_o corresponds with the low voltage side. Parameter L models both the resonant inductance and the leakage inductances of the high frequency transformer. A resistor R includes power losses at the resonant tank, transformer and semiconductors.

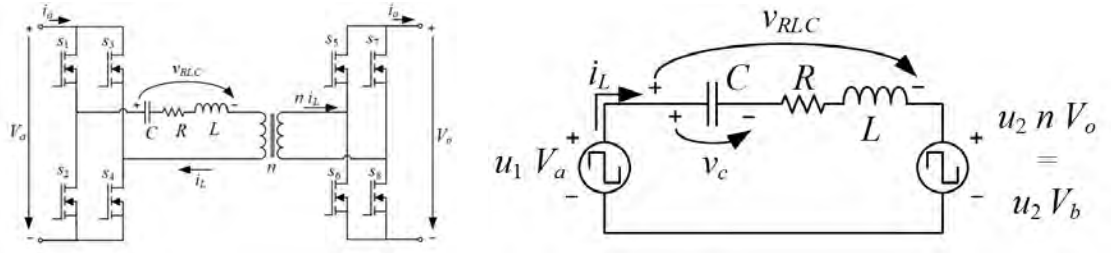


Figure 7.13: (a) DAB-SRC under study. (b) Equivalent circuit of the DAB-SRC under study.

One of the bridges works as an inverter and the other as a rectifier. In the example, the power flows from V_a to V_o. Fig. 7.13b is the transformer-left-side equivalent circuit of the converter, where $V_b = nV_o$, with n the transformer turns ratio.

Signals $u_1, u_2 \in \{-1, 1\}$ are switching variables in such a way that S1 and S4 (S5 and S8) are on if $u_1 = 1$ ($u_2 = 1$) whereas S2 and S3 (S6 and S7) are on if $u_1 = -1$ ($u_2 = -1$). A full wave modulation is applied at each bridge ($u_1 V_a$ and $u_2 V_b$), being δ the phase shift between the output waves of both bridges. As the switching frequency is above the resonant

frequency ZVS operation is assured and therefore only turn-off losses must be considered, i.e., losses due to i_{toffa} at the high-side bridge and caused by i_{toffb} at the low-voltage side. The switching frequency w (shared by the two bridges) and the phase shift δ are the control variables. The topological model of the converter is represented by Eqs. (7.46)-(7.47)

$$\frac{di_L}{dt} = \frac{1}{L}(u_1 V_a - R i_L - v_C - u_2 V_b) \tag{7.46}$$

$$\frac{dv_C}{dt} = \frac{i_L}{C} \tag{7.47}$$

Applying the Generalized Averaging Method up to the first harmonic [Sanders et al., 1991, Mahdavi et al., 1997], it yields to (7.48)-(7.49)

$$\frac{d\langle i_L\rangle_1}{dt} = \frac{\langle u_1\rangle_1 V_a}{L} - \frac{R\langle i_L\rangle_1}{L} - \frac{\langle v_C\rangle_1}{L} - \frac{\langle u_2\rangle_1 V_b}{L} - jw\langle i_L\rangle_1 \tag{7.48}$$

$$\frac{d\langle v_C\rangle_1}{dt} = \frac{\langle i_L\rangle_1}{C} - jw\langle v_C\rangle_1 \tag{7.49}$$

Considering the phase of the output modulation wave u_2 as the phase reference, the next variable changes are possible

$$\langle i_L\rangle_1 = i_{LR} + j i_{LI} \qquad \langle v_C\rangle_1 = v_{CR} + j v_{CI}$$

$$\langle u_1\rangle_1 = \frac{2}{\pi}\sin(\delta) - j\frac{2}{\pi}\cos(\delta) \qquad \langle u_2\rangle_1 = -j\frac{2}{\pi}$$

where i_{LR} and i_{LI} (resp. v_{CR} and v_{CI}) are the real and imaginary part of i_L (resp. v_C).

Finally, the model of the converter is described by

$$\frac{di_{LR}}{dt} = -\frac{Ri_{LR}}{L} + wi_{LI} - \frac{v_{CR}}{L} + \frac{2V_a}{L\pi}\sin(\delta), \tag{7.50}$$

$$\frac{di_{LI}}{dt} = -wi_{LR} - \frac{Ri_{LI}}{L} - \frac{v_{CI}}{L} + \frac{2V_b}{L\pi} - \frac{2V_a}{L\pi}\cos(\delta), \tag{7.51}$$

$$\frac{dv_{CR}}{dt} = \frac{i_{LR}}{C} + wv_{CI}, \tag{7.52}$$

$$\frac{dv_{CI}}{dt} = \frac{i_{LI}}{C} - wv_{CR}, \tag{7.53}$$

where i_{LR}, i_{LI} are the controlled variables, and v_{CR}, v_{CI} are the so-called free variables. The selection of this controlled variables provides both flexibility and generality in the proposal, allowing the practitioner to map in them any desired application requirement regarding the power conversion, efficiency and technological requests. The control variables are the switching frequency, w, which is normally related to the converted power, and the phase shift δ, which has a great influence in the turn-off current. As it can be observed in Eqs. (7.50)-(7.53), the control variables drive the dynamics of the controlled variables through strongly nonlinear equations. Therefore, conventional control strategies as PI controllers are not able to provide fast and stable transients.

7.3.3 Control law

In order to adequate to the control notation, let $\mathbf{x} = [x_1, x_2, x_3, x_4]^\top = [i_{LR}, i_{LI}, v_{CR}, v_{CI}]^\top$ denote the state vector and $\mathbf{u} = [\delta, w]^\top$ the action vector. The goal is to drive the converter

to a desired configuration, denoted by $\mathbf{x}^*$, so that the control error is $\widetilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}^*$. According to the description of the model in Section 7.3.2, x_1 and x_2 are the controlled states, so $x_3^* = x_3(t)$ and $x_4^* = x_4(t)$ for all t. Recalling our objectives, we seek an expression for $\mathbf{u}$ to achieve stability of the converter $\forall \mathbf{x}$.

To this end, we propose a nonlinear control strategy based on two types of controllers. The intuition is the following. When the converter is far from the desired state, we apply nonlinear control expressions in δ and w which achieve a fast initial transient response and small error. Near the desired configuration we also include in the control law two PIs, one for δ and the other for w, to ensure a smooth convergence of the converter to the desired configuration. The control expressions are described by

$$\delta = \begin{cases} \arctan 2\left(K_2\tilde{x}_2, K_1\tilde{x}_1\right), & \text{if } \tau > \varepsilon \\ K_P^\delta \tilde{x}_1 + K_I^\delta \displaystyle\int_0^t \tilde{x}_1 dt, & \text{otherwise} \end{cases} \tag{7.54}$$

$$w = \begin{cases} \dfrac{\frac{1}{L}K_1 x_3 \widetilde{x}_1 - \frac{R}{L}K_1 x_1 x_1^* + \frac{1}{L}K_2 x_4 \widetilde{x}_2}{K_1 x_2 \tilde{x}_1 - K_2 x_1 \tilde{x}_2} + \dfrac{-\frac{2V_b}{L\pi}K_2 \widetilde{x}_2 - \frac{R}{L}K_2 x_2 x_2^*}{K_1 x_2 \tilde{x}_1 - K_2 x_1 \tilde{x}_2}, & \text{if } \tau > \varepsilon \\ K_P^w \tilde{x}_2 + K_I^w \displaystyle\int_0^t \tilde{x}_2 dt, & \text{otherwise} \end{cases} \tag{7.55}$$

with $K_1, K_2, K_P^\delta, K_I^\delta, K_P^w, K_I^w$ and ε positive constants and

$$\tau = \left| \frac{\tilde{x}_2}{x_2} \right|. \tag{7.56}$$

The inclusion of the PIs is required because the nonlinear control law for w has a division by zero in the desired configuration, i.e., when $\tilde{x}_1 = \tilde{x}_2 = 0$ the denominator in (7.55), $K_1 x_2 \tilde{x}_1 - K_2 x_1 \tilde{x}_2$, is zero. From an application point of view, this also becomes a problem near $\mathbf{x}^*$ because small changes in the error result in big changes in the action w. The transition between the nonlinear expression and the PIs is done in such a way that the action preserves continuity, initializing the integral terms by

$$\int_0^t \tilde{x}_1 dt = (\delta_{Ly} - K_P^\delta \tilde{x}_1)/K_I^\delta \tag{7.57}$$

$$\int_0^t \tilde{x}_2 dt = (w_{Ly} - K_P^w \tilde{x}_2)/K_I^w \tag{7.58}$$

with δ_{Ly} and w_{Ly} the actions from the nonlinear terms when the transition occurs. The parameter ε defines the transition moment. On one hand it needs to be big to prevent undesired behaviors of the nonlinear term in (7.55) near the singularity. On the other hand, since the model is nonlinear, the PIs will only stabilize the system if the state is sufficiently close to the desired configuration, which requires a small value of ε. The following proposition demonstrates that there is always a sufficiently small value of ε that formally guarantees the global and asymptotic stability (GAS) of the converter.

Proposition 17. *Let $K_P^\delta, K_I^\delta, K_P^w, K_I^w$ be some constants such that $\widetilde{\mathbf{x}} = \mathbf{0}$ is a stable point of the linearized system in* (7.50)-(7.53) *under the PI control. Then, there exists a sufficiently small ϵ such that the control strategy of Eqs.* (7.54)-(7.55) *is GAS.*

Proof. Let $V(\tilde{\mathbf{x}}) = \frac{1}{2}K_1\tilde{x}_1^2 + \frac{1}{2}K_2\tilde{x}_2^2 \in \mathbb{R}$ be a Lyapunov function candidate. By developing $\dot{V}$ it follows that

$$\begin{aligned}\dot{V}(\tilde{\mathbf{x}}) =& K_1\tilde{x}_1\dot{\tilde{x}}_1 + K_2\tilde{x}_2\dot{\tilde{x}}_2 = \\ & K_1\tilde{x}_1\dot{x}_1 + K_2\tilde{x}_2\dot{x}_2 = \\ & wK_1x_2\widetilde{x}_1 - \frac{R}{L}K_1x_1^2 - \frac{1}{L}K_1x_3\widetilde{x}_1 + \frac{R}{L}K_1x_1x_1^* - \\ & - wK_2x_1\widetilde{x}_2 - \frac{R}{L}K_2x_2^2 - \frac{1}{L}K_2x_4\widetilde{x}_2 + \frac{2V_b}{L\pi}K_2\widetilde{x}_2 + \\ & \frac{R}{L}K_2x_2x_2^* + K_1\tilde{x}_1\frac{2V_a}{L\pi}\sin\delta - K_2\tilde{x}_2\frac{2V_a}{L\pi}\cos\delta.\end{aligned} \tag{7.59}$$

If $\tau > \varepsilon$, substitution of Eqs. (7.54) and (7.55) in Eq. (7.59) gives

$$\dot{V}(\widetilde{\mathbf{x}}) = -\frac{R}{L}K_1x_1^2 - \frac{R}{L}K_2x_2^2, \tag{7.60}$$

which is negative $\forall\widetilde{\mathbf{x}}$. This means that from any initial conditions, there will be a time such that $\tau \leq \varepsilon$, forcing the switch to the PI controllers. The initialization in (7.57)-(7.58) ensures continuity of the action. By assumption in the claim, the PIs are designed such that the linearized system is stable in $\widetilde{\mathbf{x}} = \mathbf{0}$. This implies [Khalil, 2014] that there exists a neighborhood $\mathcal{D}$ of $\widetilde{\mathbf{x}} = \mathbf{0}$ where the PIs achieve local asymptotic stability of the nonlinear system. Thus, there exists a sufficiently small ε such that the control achieves GAS, concluding the proof. □

The only missing part is the choice of an appropriate value for ε. This could be tackled by means of simulations and optimization search over the state space of the converter. Nonetheless, in practice it is just enough to set ε to a sufficiently small value, e.g., 0.05 in our experiments, to obtain a stable and fast response even with an aggressive PI design.

7.3.4 Adaptations

In an ideal scenario, this proposal is ready to control the converter. However, some issues arise from the implementation requirements.

Firstly, the control strategy depends on x_3 and x_4. In particular, this dependence is represented in Eq. (7.59) by

$$-\frac{1}{L}K_1x_3\tilde{x}_1 - \frac{1}{L}K_2x_4\tilde{x}_2. \tag{7.61}$$

However, this information is not accessible in the real system since x_3 and x_4 correspond to the real and imaginary part of v_C. As we can not measure them, a sensorless method is proposed which eliminates this dependence.

Considering the limitations in the design of power converters, the voltage at the resonance capacitor v_C is always below a certain value V_{lim}. Recalling the harmonic approximation in Section 7.3.2,

$$v_{C,pp} = \sqrt{2}v_{C,rms} = \sqrt{x_3^2 + x_4^2}, \tag{7.62}$$

and so both x_3 and x_4 are limited by the maximum admissible voltage in the capacitor

$$v_{C,pp} = \sqrt{2}v_{C,rms} = \sqrt{x_3^2 + x_4^2} \leq V_{lim} \rightarrow |x_3|, |x_4| \leq V_{lim}. \tag{7.63}$$

In the worst case, that is, Eq. (7.61) is at its maximum value, the term

$$\frac{1}{L}V_{lim}(K_1|\tilde{x}_1| + K_2|\tilde{x}_2|) \tag{7.64}$$

must be compensated. The modification consists in replacing (7.61) by (7.64) in the nonlinear term of (7.55). Additionally to the sensorless feature, this adaptation has the advantage that it only adds speed to the transient response since the numerator in Eq. (7.55) when $\tau > \varepsilon$ increases. Finally, the value of C is no longer required, so the control is robust to any uncertainty or tolerance with respect to the resonance capacitor.

The second modification is related to the difference between the characterized parameters and the real ones. This could lead to an incorrect compensation of the terms in (7.54) and (7.55) and destabilize the system. To overcome this circumstance, we propose an adaptive law which works in real time. Let us define $a_1 = \frac{R}{L}$ and $a_2 = \frac{1}{L}$ to ease the design. These parameters correspond to the coefficients multiplying the terms in the model (7.50)-(7.53). In case R and L are not known with prevision, a_1 and a_2 can be expressed as the sum of two terms $a_1 = \hat{a}_1 + \tilde{a}_1$ and $a_2 = \hat{a}_2 + \tilde{a}_2$. Here, $\hat{a}_1$ and $\hat{a}_2$ are the estimated parameters whilst $\tilde{a}_1$ and $\tilde{a}_2$ are the error between the real and estimated parameters.

The adaptation law seeks to ensure the stability of the converter by evolving the values $\tilde{a}_1$ and $\tilde{a}_2$, in such a way that the control defined in Eqs. (7.54) and (7.55) still works. So the condition of negative derivative over $\dot{V}$ must be preserved.

Proposition 18. *Given the control law defined by Eqs.* (7.54)-(7.55) *and the design conditions described in Proposition 17, the adaptation law*

$$\dot{\hat{a}}_1 = -\frac{1}{K_{a_1}}(K_1 x_1 \tilde{x}_1 + K_2 x_2 \tilde{x}_2) \tag{7.65}$$

$$\dot{\hat{a}}_2 = -\frac{1}{K_{a_2}}(K_1 V_{lim} \tilde{x}_1 + K_2 V_{lim} \tilde{x}_2 - \frac{2V_b}{\pi} K_2 \tilde{x}_2) \tag{7.66}$$

preserves the GAS, with K_{a_1} and K_{a_2} two positive gains.

Proof. Redefine the Lyapunov function in Proposition 17 to consider the error in the parameters

$$V(\tilde{\mathbf{x}}, \tilde{a}_1, \tilde{a}_2) = \frac{1}{2}K_1\tilde{x}_1^2 + \frac{1}{2}K_2\tilde{x}_2^2 + \frac{1}{2}K_{a_1}\tilde{a}_1^2 + \frac{1}{2}K_{a_2}\tilde{a}_2^2. \tag{7.67}$$

Then, $\dot{V}$ changes to

$$\begin{aligned} \dot{V}(\tilde{\mathbf{x}}, \tilde{a}_1, \tilde{a}_2) =& K_1\tilde{x}_1\dot{x}_1 + K_2\tilde{x}_2\dot{x}_2 + K_{a_1}\tilde{a}_1\dot{\tilde{a}}_1 + K_{a_2}\tilde{a}_2\dot{\tilde{a}}_2 = \\ & K_{a_1}\tilde{a}_1\dot{\tilde{a}}_1 + K_{a_2}\tilde{a}_2\dot{\tilde{a}}_2 - \tilde{a}_1K_1x_1\tilde{x}_1 - \tilde{a}_2K_1x_3\tilde{x}_1 \\ & + \tilde{a}_2K_1\tilde{x}_1\frac{2V_a}{\pi}\sin\delta - \tilde{a}_1K_2x_2\tilde{x}_2 - \tilde{a}_2K_2x_4\tilde{x}_2 \\ & + \tilde{a}_2\frac{2V_b}{\pi}K_2\tilde{x}_2 - \tilde{a}_2K_2\tilde{x}_2\frac{2V_a}{\pi}\cos\delta + W, \end{aligned} \tag{7.68}$$

where W includes all the terms which depend on $\hat{a}_1$ and $\hat{a}_2$ and it is strictly negative, since is the result of applying the control law in Eqs. (7.54) and (7.55). To preserve the GAS properties over $\dot{V}$, the other terms are distributed

$$K_{a_1}\tilde{a}_1\dot{\tilde{a}}_1 = \tilde{a}_1 K_1 x_1 \tilde{x}_1 + \tilde{a}_1 K_2 x_2 \tilde{x}_2, \tag{7.69}$$

$$K_{a_2}\tilde{a}_2\dot{\tilde{a}}_2 = \tilde{a}_2 K_1 x_3 \tilde{x}_1 + \tilde{a}_2 K_2 x_4 \tilde{x}_2 - \tilde{a}_2 \frac{2V_b}{\pi} K_2 \tilde{x}_2, \tag{7.70}$$

considering that $\tilde{a}_2 K_1 \tilde{x}_1 \frac{2V_a}{\pi} \sin\delta = \tilde{a}_2 K_2 \tilde{x}_2 \frac{2V_a}{\pi} \cos\delta$ by the definition of δ. Then,

$$\dot{\tilde{a}}_1 = \frac{1}{K_{a_1}}(K_1 x_1 \tilde{x}_1 + K_2 x_2 \tilde{x}_2), \tag{7.71}$$

$$\dot{\tilde{a}}_2 = \frac{1}{K_{a_2}}(K_1 x_3 \tilde{x}_1 + K_2 x_4 \tilde{x}_2 - \frac{2V_b}{\pi} K_2 \tilde{x}_2). \tag{7.72}$$

Assuming that the real parameters are quasi-static,

$$\dot{\hat{a}}_1 = -\dot{\tilde{a}}_1, \qquad \dot{\hat{a}}_2 = -\dot{\tilde{a}}_2. \tag{7.73}$$

The sensorless term in (7.64) is included in Eq. (7.66) to consider the absence of measurements in x_3 and x_4. Additionally, Eq. (7.55) changes to

$$w = \begin{cases} \dfrac{-\hat{a}_2 V_{lim}(K_1|\tilde{x}_1| + K_2|\tilde{x}_2|) - \hat{a}_1 K_1 x_1 x_1^*}{K_1 x_2 \tilde{x}_1 - K_2 x_1 \tilde{x}_2} + \dfrac{-\frac{2V_b}{L\pi} K_2 \widetilde{x}_2 - \hat{a}_1 K_2 x_2 x_2^*}{K_1 x_2 \tilde{x}_1 - K_2 x_1 \tilde{x}_2}, & \text{if } \tau > \varepsilon \\ K_P^w \tilde{x}_2 + K_I^w \displaystyle\int_0^t \tilde{x}_2 dt, & \text{otherwise} \end{cases} \tag{7.74}$$

At this point, following a similar procedure to the one described in the proof of Proposition 17, GAS is proved. □

It is underlined that the adaptation law does not seek to obtain the precise real value of the parameters. It simply evolves the estimates to ensure the stability of the converter.

The last aspect to address is the discretization of the control strategy. Its implementation is done by a digital device, which immediately implies that the control strategy must work in discrete time with a sampling time T_s. This contrast with the continuous time control proposal. Aiming to preserve the properties and structure of the nonlinear controller, a limitation in the velocity of the actions is considered, $|\dot{\delta}| < \dot{\delta}_{\max}$ and $|\dot{w}| < \dot{w}_{\max}$, with $\dot{\delta}_{\max}$ and $\dot{w}_{\max}$ part of the design. Thanks to this, the control strategy can be implemented in discrete time by a Zero Order Hold (ZOH). By limiting the change in $\mathbf{u}$, the behavior of the converter is smoothed out because the overdamped dynamic dominates the underdamped dynamic characteristic of the resonant circuit. The speed of the transient response is not deteriorated because the control strategy pursues to control the overdamped dynamic. Furthermore, by limiting $|\dot{\delta}|$ and $|\dot{w}|$ the devices are being protected from premature degradation.

Algorithm 10 summarizes the sequence of control steps of the complete strategy, and Fig. 7.14 details the design process for the control parameters. Constants K_1 and K_2 are tuned according to the desired settling time of x_1 and x_2 respectively, since they are

multiplied by the corresponding errors in Eq. (7.74). Then, we obtain the linearized model in Eqs. (7.50)-(7.53) to design $K_P^\delta, K_I^\delta, K_P^w, K_I^w$ using pole placement, to ensure stability in the small gain. Nevertheless, any other classic technique can be applied for this step and, if needed, it can be further refined in experiments. The transient response is finally tuned by K_{a_1} and K_{a_2}, which permit us to speed up x_1 and x_2 since they drive the evolution of $\hat{a}_1$ and $\hat{a}_2$. The selection of ε is empirical and depends on the model of the DAB-SRC: it must be sufficiently small to leverage the speed of the nonlinear controller and big enough to satisfy stability and obtain zero steady-state error. Finally, $\dot{\delta}_{\max}, \dot{w}_{\max}$ are selected to verify security, sampling time and other constraints on the real device.

Algorithm 10 Complete control strategy

```
Initialize variables
while True do
    Read Sensors and References: x_1, x_2, x_1^*, x_2^*, V_a, V_b
    if Change in references or voltages and τ > ε then
        Change to Lyapunov mode
    end if
    if Lyapunov Mode then
        Calculate adaptation law: â_1, â_2, Eqs. (7.65) (7.66)
        Calculate δ and w, Eqs. (7.54) and (7.74)
        Calculate τ, Eq. (7.56)
        if τ < ε then
            Load integrals of PIs, Eqs. (7.57) and (7.58)
            Activate PI mode
        end if
    else {PI mode}
        Calculate δ and w, Eqs. (7.54) and (7.74)
    end if
    Apply saturations
    Wait until (k + 1)T_s
end while
```

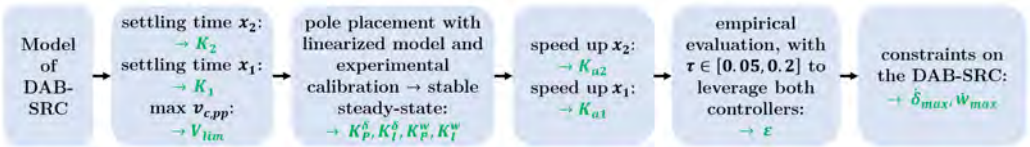


Figure 7.14: Flow chart detailing the design of the control parameters.

Note that all expressions can be implemented in a digital device. The only operation that might not be available is the arc tangent, but in that case it could be implemented through a LUT. The design formally guarantees the stability of the converter and, therefore, zero error in the steady state. On the other hand, the design constants allow to regulate the transient behavior although, due to the changing structure of Eqs. (7.54) and (7.74), as well as the saturations in $\dot{\delta}$ and $\dot{w}$, the settling time can not be set a priori.

Lastly, it is noteworthy that the expressions of the control proposal are independent on the particular operating point the converter is: it only needs to track the difference between the desired reference and the actual configuration to decide the operating control mode (checking τ) and to calculate the inputs w and δ. This is a consequence of Propositions 17 and 18, which prove global asymptotic stability. Therefore, it is not necessary to study whether there are multiple solutions for a given operating point of the converter or it is unique, and no special treatments given the operating point are required.

7.3.5 Simulations

A complete model has been created in Simulink®2011b, testing the control strategy directly with the circuit equations (7.46)-(7.47). Besides, the control algorithm has been implemented in *C* language (*S-Function Builder* [MathWorks®, 2020a] [MathWorks®, 2020b]), so the code can be directly transferred to a microcontroller or DSP. The model includes the filters to measure i_{LR} and i_{LI}, the modulator to generate u_1 and u_2 and a worst-case latency of a sampling period T_s between the calculation of the action and its application.

To validate the proposal, we compare the results with those obtained by two other control strategies. The first one is based on PIs, which is the existing policy in the device and it is the common alternative in industrial applications [Engel et al., 2012]. The second one implements a more advanced control, namely, a Sliding Mode Control, taking the state-of-the-art solution proposed in [Jeung and Lee, 2018]. We have adjusted the values of the control parameters to fit our controller. It is noteworthy that [Jeung and Lee, 2018] deals with a DAB converter without resonance, so our problem is harder to solve due to the SRC circuit. Besides, its proposal applies to Single-Input Single-Output systems; therefore, to implement it in our setup we have fixed one of the inputs and used the solution in [Jeung and Lee, 2018] to control the other one. In particular, we have fixed $\delta = \arctan 2(\tilde{x}_2, \tilde{x}_1)$, and apply the Differential Integral Sliding Mode controller from [Jeung and Lee, 2018] in w to be able to cover the entire dynamic range of the converter. Finally, the Sliding Mode controller includes PIs adjusting the steady-state error since, unlike [Jeung and Lee, 2018], the sampling frequency is $15 - 40$ times lower than the switching frequency, making them necessary to avoid oscillations around the desired reference.

It must be underlined that i_{LI} is obtained from the full-wave rectification of i_L and its subsequent filtering by an analogue filter. Moreover, to avoid the influence of spurious in i_{LR} and mitigate high frequency noise, a digital filter is included after the measurement to achieve a slight filtering without modifying the hardware. In particular, a first order Butterworth low-pass filter is considered, with a sampling frequency of $T_s = 2.5$kHz and a normalized cut-off frequency of 0.6

$$F(z) = \frac{0.5792z + 0.5792}{z + 0.1584}. \tag{7.75}$$

This filter is auxiliary and does not belong to the control strategy. The amplitude of the measurement noise has been scaled to fit the one obtained with the real hardware equipment of the experiments on Section 7.3.6, which is in the range $[0.25, 0.30]V$ of maximum

amplitude. Table 7.7 summarizes the main features of the commercial converter.

Parameter	Symbol	Value
Sample time	T_s	400μs
High bridge voltage	V_a	$[325-400]$V
Nominal high bridge voltage	$V_{a,n}$	375V
Low bridge voltage	V_o	$[93-138]$V
Nominal low bridge voltage	$V_{o,n}$	116V
Equivalent coil	L	110μH
Capacitor	C	$254n$F
Equivalent resistance	R	1Ω
Transformation ratio	n	2.17
Minimum frequency	$f_{\min}$	35kHz
Maximum frequency	$f_{\max}$	100kHz
Resonance frequency	f_o	30kHz
Maximum δ	$\delta_{\max}$	$\pi/2$rad
Minimum δ	$\delta_{\min}$	$-\pi/2$rad
Maximum admissible voltage in C	V_{lim}	$450\sqrt{2}$V

Table 7.7: Main parameters of the converter.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
V_a(V)	375	375	375	400	400	400	325	325	325
V_o(V)	116	138	93	116	138	93	116	138	93

Table 7.8: Simulated combinations of voltages V_a and V_o.

Given the voltage ranges of V_a and V_o in Table 7.7, 9 combinations are selected to cover all possible scenarios, listed in Table 7.8. For each combination, three situations are studied: given a fixed reference value $i^*_{LR} = 1$A, we apply either positive or negative steps on the reference value i^*_{LI} (denoted as Δi_{LI}), always starting from the midpoint of the characterized static range. For each case we record the settling time, checking if i_{LR} goes below 0A during the transient for security reasons. These metrics are named ts_{cp}, ts_{pi} and ts_{sm} for our proposal, the PIs and the Sliding Mode respectively.

The control parameters are adjusted to the following: $K_1 = 1$, $K_2 = 1$, $K^\delta_P = 0.0002$, $K^\delta_I = 0.015$, $K^w_P = 10$, $K^w_I = 5000$, $K_{a_1} = 2000$, $K_{a_2} = 1000$, $\dot{\delta}_{\max} = 2\pi/360$rad, $\dot{w}_{\max} = 2\pi 5000$rad/s, $\varepsilon = 0.05$. We compare with two PIs with the following parameters: $K^\delta_P = 0.0001$, $K^\delta_I = 0.015$, $K^w_P = 1$, $K^w_I = 1000$. The difference between the PIs of the proposed control strategy and the latest is due to the fact that, in our proposal, the PIs perform in the small signal range, in contrast with the existing control policy which works in the entire dynamic range. Thus, in the existing control policy the PIs must be slow enough to reject the effects of nonlinearities and to evade instability. This is another advantage of the proposal, where the PIs can be tuned to be more demanding since they always work in the small signal range. Besides, the Sliding Mode Controller has been tuned as follows (following the notation from [Jeung and Lee, 2018]): $K_1 = 50$, $K_2 = 50$, $K_3 = 1000$, $\mu = 20000$, $K^\delta_P = 0.0002$, $K^\delta_I = 0.015$, $K^w_P = 10$, $K^w_I = 5000$.

The results are shown in Table 7.9. The control strategy achieves a fast dynamic response for all initial and desired configurations, with settling times between 3ms and 6ms. Obtaining such a uniform settling time is an advantage because the convergence time can be known in advance despite the amplitude of the step. Taking into account that $T_s = 400\mu$s, the controller employs $8 - 15$ samples to drive the converter to the desired configuration, which is the limit usually considered in discrete control theory to capture the relevant information during the transient. In all cases the requirements with regard to i_{LR} are accomplished.

Combination	Δi_{LI} (A)	i^*_{LI} (A)	Settling Times (ms)		
			ts_{cp}	ts_{pi}	ts_{sm}
(1)	4.0 – 8.5	4.75	4.0	16.0	4.0
(1)	4.0 – 8.5	7.25	3.0	10.0	3.5
(1)	4.0 – 8.5	8.25	4.0	12.0	4.0
(2)	3.5 – 6.5	4.0	4.0	12.0	5.0
(2)	3.5 – 6.5	5.5	3.0	10.0	3.0
(2)	3.5 – 6.5	6.25	5.0	12.0	4.5
(3)	4.5 – 9.5	5.5	4.0	16.0	5.0
(3)	4.5 – 9.5	8.0	4.0	10.0	4.5
(3)	4.5 – 9.5	8.75	4.0	12.0	4.5
(4)	4.0 – 10.0	5.0	4.0	18.0	5.0
(4)	4.0 – 10.0	7.5	3.0	8.0	3.5
(4)	4.0 – 10.0	8.5	3.0	10.0	4.0
(5)	3.5 – 7.0	4.0	4.0	14.0	4.5
(5)	3.5 – 7.0	5.75	3.0	10.0	3.0
(5)	3.5 – 7.0	6.25	5.0	12.0	4.0
(6)	5.0 – 11.0	6.0	4.0	14.0	4.0
(6)	5.0 – 11.0	9.0	4.0	6.0	4.0
(6)	5.0 – 11.0	10.0	4.0	8.0	4.0
(7)	3.0 – 6.5	3.5	4.0	20.0	5.0
(7)	3.0 – 6.5	5.5	5.0	10.0	5.0
(7)	3.0 – 6.5	6.0	6.0	12.0	6.0
(8)	2.5 – 5.0	3.0	5.0	8.0	5.0
(8)	2.5 – 5.0	4.25	4.0	8.0	4.5
(8)	2.5 – 5.0	4.75	6.0	8.0	6.0
(9)	3.0 – 8.0	4.0	4.0	16.0	4.5
(9)	3.0 – 8.0	6.5	4.0	10.0	4.5
(9)	3.0 – 8.0	7.25	4.0	10.0	5.0

Table 7.9: Results of the systematic evaluation of the converter. Red cell means a violation in the condition $I_{LR} > 0$A.

Comparing the static ranges it can be deduced that greater jumps between V_a and V_o yield to greater static ranges, which is reasonable since there is a greater voltage in the resonant tank. Furthermore, the higher the value of V_a and V_o, the higher the values of Δi_{LI}. Table 7.9 also shows the results obtained by the PI control. The settling time is of

the order of 2-4 times greater than with our control proposal. In addition, it depends on the amplitude of the step and the specific configuration in which the DAB-SRC is, so it is more complicated to predict. Regarding the Sliding Mode solution, Table 7.9 shows that the speed of the transient response is a bit slower than the one achieved by our proposal. The settling time is almost constant as well, but with a very important disadvantage. It can be seen that the red cells marking the cases where the condition $I_{LR} > 0$A is violated appear when we impose a negative step in the I_{LI} reference. This is not allowed in our application. The reason of this is that the Sliding Mode Control requires high control frequencies to be able to correctly approach the sliding surfaces. In our setup, with an appropriate tuning of control parameters, the small control frequency makes the states overpass the sliding frontiers, which in the end make i_{LR} to go beyond 0A.

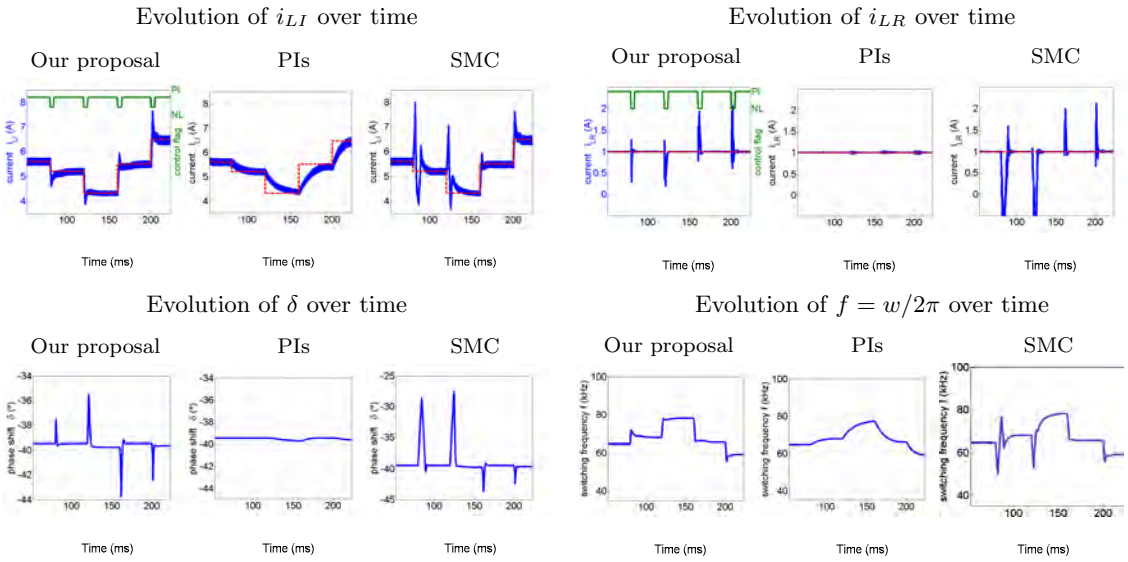


Figure 7.15: Behavior of the converter against a uniformly random sequence of steps in i^*_{LI}, where $V_a = V_{a,n}$ and $V_o = V_{o,n}$. Green flag indicates the operation mode of our control strategy. PI label means PI mode and NL label means Nonlinear mode.

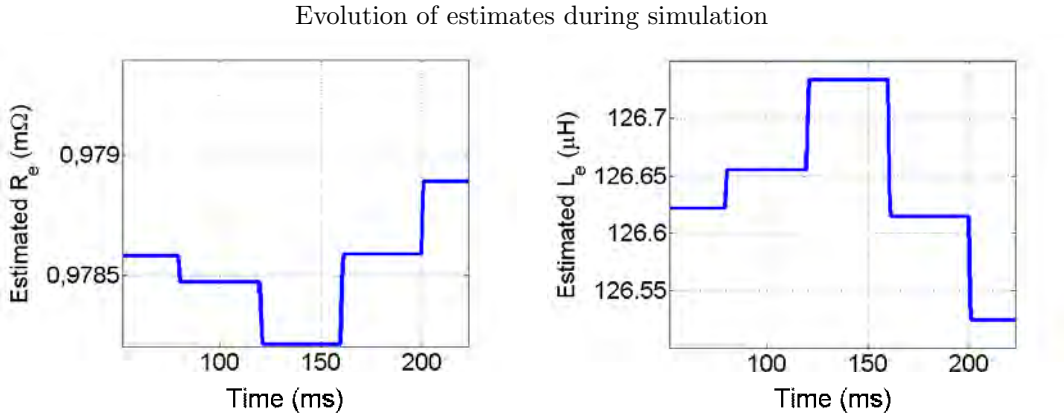


Figure 7.16: Evolution of the estimation of R and L.

To exemplify the behavior of the converter with the control policies, Fig. 5 shows the evolution of the converter against a uniformly random sequence of steps of i_{LI} every 40ms,

with nominal values in V_a and V_o. Thanks to the proposed control strategy, the response is satisfactory covering all the static range of the converter with identical settling times. In the first moments after a reference change, i_{LR} and i_{LI} exhibit some overshoot due to the speed demanded, but always satisfying the operation requirements. The actions are achievable and the adaptation law modifies the values of $\hat{R}$ and $\hat{L}$ (see Fig. 7.16) to ensure the stability of the system. In this case, the reset of $\hat{R}$ and $\hat{L}$ to the original estimation is not considered but it can be done to guarantee that they prevail near the characterized values. No effect caused by control latency is appreciated. With the PI policy, the response of the converter is slower than before and is dependent on the amplitude of the step introduced. As δ hardly changes, i_{LR} does not change either. The current circulating through the transistors is very stable but since the control barely has δ to regulate i_{LI}, w must take care of regulating the power, with a response in i_{LI} much slower than with the control proposal. This justifies the differences in the response of i_{LR}, which influences the transient in i_{LI} and explains the brief overshoot in our control strategy. With the Sliding Mode Control, the transient response is the most spiky in both i_{LR} and i_{LI}. In the case of i_{LR} this yields to violations in the condition $i_{LR} > 0$A. Besides, a small control frequency with respect to the switching frequency makes the sliding controller to overpass the sliding frontiers between surfaces and, thus, achieve the spiky transient response in i_{LI} as well. As a last remark, the Sliding Mode Control has moderate robustness against uncertainties in the parameters. However, this intrinsic robustness is highly dependent on the particularities of the system and is not quantified, so at some point degradation or uncertainty may destabilize the converter. On the other side, our proposal has theoretical guarantees of robustness due to the adaptive law and global asymptotic stability is irrespective of uncertainty.

The advantages and disadvantages of our proposal, compared to the other techniques, are summarized in Table 7.10. Green color is advantage, red color is disadvantage, and yellow is neutral.

Our proposal	PI Control	Sliding Mode Control
Global Asymptotic Stability	Small Gain Stability	Global Asymptotic Stability
Constant Fast Settling Time	Step-dependent Slow Settling Time	Constant Fast (less than our proposal) Settling Time
Overshoot may appear	No overshoot	Overshoot may appear
Preserves $i_{LR} > 0$A	Preserves $i_{LR} > 0$A	Does not preserve $i_{LR} > 0$A
Adaptation adds robustness against uncertainty and keeps fast transient	Dynamics must be slowed down for robustness	Sliding is robust against moderate levels of uncertainty

Table 7.10: Comparison among controllers. Green color is advantage, red color is disadvantage, and yellow is neutral.

7.3.6 Experiments

The setup of the experiments is shown in Fig. 7.17. The converter, provided by Epic Power®, is disassembled, with access to the motherboard, the microcontroller and the auxiliary measuring devices. The oscilloscope is a model MSO7104B of InfiniiVision® [Technologies, 2010] and has four channels and a bandwidth of 1GHz. The oscilloscope is over a bank of power supplies, which provide the energy to the converter. Specifically, these power supplies can set the voltage for bi-directional current demands, so they can supply energy (load) or recover it (discharge). The 32-bit microcontroller that implements the control strategy comes with floating point unit (FPU) and can perform the arctan 2 operation [Instruments, 2018]. It also has enough configurable output pins to monitor, either through serial communication or direct measurement with an oscilloscope, internal variables for debugging.

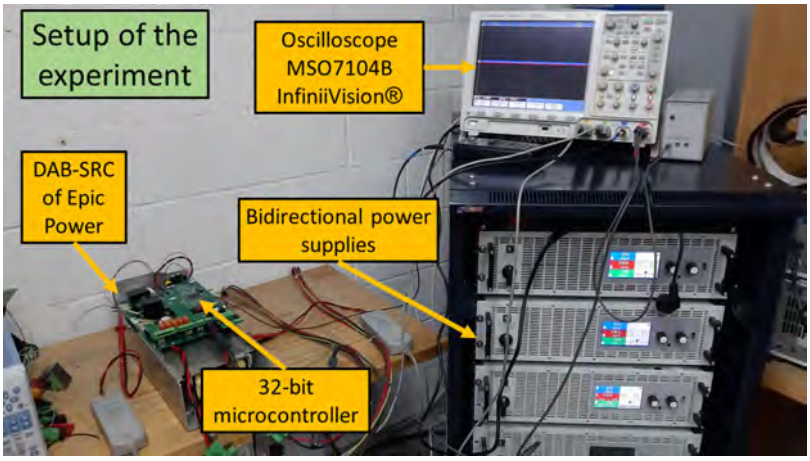


Figure 7.17: Setup of the experiment.

The results of the experiments are shown in figures. 7.18-7.20. The first series of experiments, in Fig. 7.18, consists in applying reference steps of different amplitude (positive and negative) with different voltages and starting up conditions. Fig. 7.18 shows that the control strategy achieves a fast and stable response even when the converter is turned on. In relation with i_{LR}, the microcontroller receives a data each 10ms so it is not possible to follow its transient response. However, we have verified that after 10ms from the change in the reference, i_{LR} converges to i^*_{LR}, a hundred times faster with our control solution than with the previous policy. The experiments validate that the control strategy globally achieves a fast and stable response, converging to the desired configuration in not more than 6ms. It is noteworthy that in Fig. 7.18c and Fig. 7.18e a spike appears in the reference change. This is due to the influence of the output filter of the converter when it is started up. By construction of the commercial converter and for security reasons, the PWM lasts one sample time to apply the control inputs, so the first 400μs the converter is alone against the predefined values of w and δ, resonating with the $4.5k$Hz frequency of the output filter. After that, the control is applied and the spike disappears.

To further investigate our proposal, the next experiment evaluates the response of the converter against uncertainties in R and L, recalling that our proposal is immune to the

value of C. To do so, we have mapped the uncertainty by initializing the control with values of R and L different from those characterized. In particular, we have assumed differences of the 50%, which is much higher than the typical deviation of these parameters. The experiments have been conducted with the converter operating at $V_a = 375\text{V}$, $V_o = 116\text{V}$, i^*_{LI} from 7A to 11A. As we can see in Fig. 7.19, the transient and steady-state response is almost mirrored irrespective of the uncertainty in R and L. This is an evidence of the robustness achieved by the adaptation law of our proposal. In all the cases the control obtains zero steady-state error.

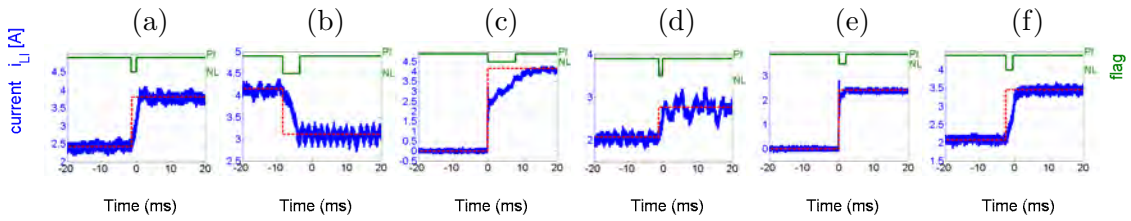


Figure 7.18: Results of the experiment with the commercial converter against different values of voltage and current. All of them have been conducted using the nominal parameters of the converter. Green flag indicates the operation mode. PI label means PI mode and NL label means Nonlinear mode. The configurations are: (a) V_a=375V,V_o=117V, i^*_{LI} from 2.4A to 3.8A; (b) V_a=375V,V_o=116V, i^*_{LI} from 4.2A to 3.1A; (c) V_a=375V,V_o=116V, i^*_{LI} from 0A to 4.2A; (d) V_a=325V,V_o=138V, i^*_{LI} from 2.1A to 2.8A; (e) V_a=325V,V_o=93V, i^*_{LI} from 0A to 2.4A; (f) V_a=400V,V_o=93V, i^*_{LI} from 2.1A to 3.5A.

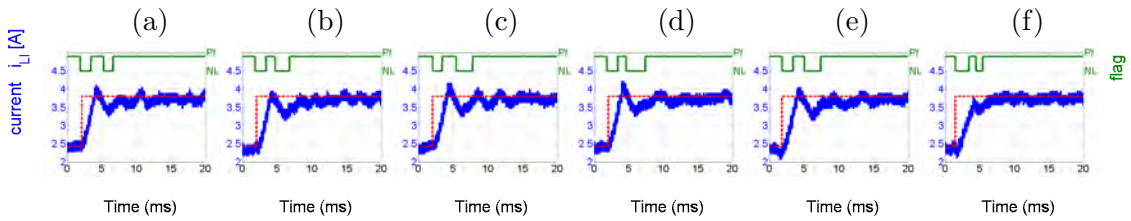


Figure 7.19: Results of the experiment with the commercial converter against uncertainty in the parameters. All of them have been conducted with the converter operating at $V_a = 375\text{V}$, $V_o = 116\text{V}$, i^*_{LI} from 2.4A to 3.8A. Green flag indicates the operation mode. PI label means PI mode and NL label means Nonlinear mode. The configurations are: (a) $\hat{L}(0)$=L,$\hat{R}(0)$=R; (b) $\hat{L}(0)$=0.5L,$\hat{R}(0)$=R; (c) $\hat{L}(0)$=1.5L,$\hat{R}(0)$=R; (d) $\hat{L}(0)$=L,$\hat{R}(0)$=0.5R; (e) $\hat{L}(0)$=L,$\hat{R}(0)$=1.5R; (f) $\hat{L}(0)$=0.5L,$\hat{R}(0)$=1.5R

The final experiment assesses the robustness of the control strategy against sudden changes in V_a and V_o. We have programmed a sequence of random uniformly random steps in V_a and V_o, positive or negative, each 40ms. The delay between steps in V_a and V_o is also random. The converter has been driven to steady-state at $V_a = 375\text{V}$, $V_o = 116\text{V}$, and $I_{LI} = 3.1\text{A}$. Fig. 7.20 shows that the converter is completely robust against sudden changes in the voltages at both sides. The worst case, around $t = 120\text{ms}$, corrects I_{LI} in less than 5ms and corresponds to a high-amplitude step in V_a. The rest of the time I_{LI} follows the reference irrespective of the disturbances.

7.4 Conclusions

This chapter has contributed onto the critical aspects of identification, estimation, and control in power systems, with a focus on their significance in modern robotics systems.

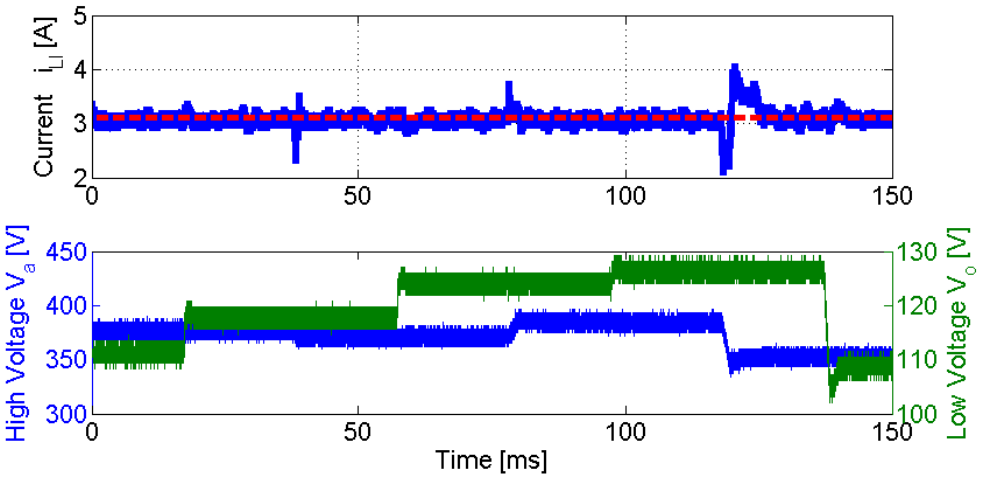


Figure 7.20: Results of the experiment with the commercial converter against disturbances on V_a and V_o: (top) evolution of i_{LI}; (bottom) disturbances in V_a (blue) and V_o (green). The converter has been firstly driven to $V_a = 375$V, $V_o = 116$V, and $I_{LI} = 3.1$A

By addressing the increasing complexities in these systems, the chapter has presented novel solutions and insights:

- **Automatic identification of power-temperature dynamics**: An innovative automatic technique for estimating semiconductor power losses in power converters has been introduced. This method, based on an optimization-based identification of the linear discrete-time dynamic system, offers high accuracy and the flexibility to incorporate desired requirements and constraints. It surpasses traditional post-design approaches in terms of accuracy and interpretability, proving its efficacy in various scenarios, including switching operating conditions.

- **Voltage prediction using Gaussian Process Regression**: The chapter addressed the challenges of future voltage estimation in photovoltaic systems by employing three different GPR recursive multistep models: Regular GPR, Multiple Expert GPR, and Sparse GPR. Extensive analysis, utilizing a large dataset with voltage, current, and temperature data, revealed that simple kernels like RQ offer a good balance between computational efficiency and accuracy. The results demonstrated the robustness and effectiveness of GPR models, particularly the ME and Sparse models, in providing accurate voltage predictions and facilitating fault-tolerant applications.

- **Nonlinear control of a dual active bridge series resonant converter**: A novel control strategy for a series resonant converter in a dual active bridge topology has been proposed. This nonlinear control approach ensures global asymptotic stability, leading to significant improvements in response speed and overall performance compared to traditional PI and Sliding Mode control loops. The strategy's robustness and stability properties have been validated through both simulations and experiments with a commercial converter.

These contributions collectively address the growing demands for precise and efficient

power management in modern robots and sustainable energy applications. The insights and solutions presented in this chapter pave the way for future research and development in the field, promising further advancements in the identification, estimation, and control of complex power systems.

Part V

Conclusions

Chapter 8

Conclusions

This thesis explored the exciting domain of distributed multi-robot coordination, with a particular focus on the formulation of control policies that are simultaneously scalable and accurate. The research exposed in the thesis proposes novel control theoretical mechanisms to guarantee effective collaboration in teams of robots subject to a distributed infrastructure, regardless of the number of robots involved. The research adopted a synergistic approach, integrating principles from physics, geometric control theory and machine learning, enabling robots to interact with their environment and with each other in the absence of a central coordination unit, thereby advancing the frontiers of robotic autonomous collaboration.

In particular, the thesis first addressed the control of input-nonaffine systems, a class of nonlinear systems that has historically presented significant challenges for traditional control design, and which underlies distributed multi-robot applications from herding to navigation in cluttered environments. Moreover, the thesis pioneered new approaches to distributed stochastic estimation. These methods enable robots to estimate and track variables of interest in a distributed manner, even when faced with noisy or incomplete information, thereby enhancing their perception and situational awareness; these tools are of general applicability in multi-robot domains, with special significance in active exploration or search and rescue. Another pioneering contribution of this thesis is the introduction of innovative techniques for the learning of distributed multi-robot policies, which enable robots to learn and adapt their behavior through experience and interaction with their surrounding environment. These techniques capitalize on the capabilities of machine learning algorithms, with robots acquiring new skills and refining their performance over time, ultimately leading to more intelligent and autonomous systems in multi-robot applications such as cooperative transportation of unknown objects or competitive resource collection. Furthermore, recognizing that the realization of such sophisticated multi-robot control policies hinges critically on a stable and reliable power supply, the thesis extends its investigation to encompass the control of power devices. This aims to ensure that the robots maintain operational integrity and efficiency, even under demanding conditions, by addressing challenges related to energy consumption, thermal management, and power conversion, with applicability beyond robotics in general power electronics problems.

8.1 Contributions

Specifically, the research conducted in this thesis has yielded the following key contributions:

- **Implicit Control for input-nonaffine systems**: We introduced Implicit Control (Chapter 2), a groundbreaking technique capable of stabilizing general input-nonaffine dynamics, which are prevalent in many real-world multi-robot systems. This new control framework offers a solution to the limitations of traditional nonlinear control techniques that struggle to handle such dynamics. We demonstrated the efficacy of Implicit Control in challenging multi-robot herding scenarios, where robots must coordinate to steer a group of non-cooperative evaders to a desired region. Our results showed that Implicit Control can effectively stabilize the system and achieve the desired herding behavior even with limited knowledge of the evaders' dynamics.

- **Distributed control techniques**: To extend the applicability of Implicit Control to decentralized settings, we developed novel accelerated dynamic consensus algorithms, distributed optimization methods, and dissensus methods based on opinion dynamics (Chapter 3). These techniques enable robots to coordinate their actions and achieve global objectives in a distributed manner, without relying on a central coordinator. Our accelerated consensus algorithms allow robots to quickly agree on shared information, while our distributed optimization methods enable them to solve complex optimization problems in a decentralized way. The dissensus methods based on opinion dynamics offer a mechanism for robots to negotiate and resolve conflicts when their individual objectives diverge.

- **Ellipsoidal methods for distributed stochastic estimation**: We devised ellipsoidal methods that leverage the outer Löwner-John method to achieve optimal accuracy and communication efficiency in distributed stochastic estimation. This led to the creation of a certifiable distributed Kalman filter, which is the first of its kind to offer optimality guarantees under unknown correlations (Chapter 4). This filter allows robots to estimate the state of the system in a distributed manner while accounting for uncertainties and communication constraints. We also developed two distributed versions of the outer Löwner-John method, enabling robots to reconstruct global optimal outer ellipsoids by exchanging local ellipsoidal representations (Chapter 5). This method has applications in various domains, including multi-robot herding, where robots can use it to estimate the shape and position of the herd in a distributed way.

- **Physics-informed policy parameterization**: We presented a physics-informed policy parameterization that combines port-Hamiltonian mechanics and self-attention neural networks to learn distributed interactions in a scalable and sample-efficient manner (Chapter 6). This approach encodes the distributed nature of multi-robot systems and their physical constraints into the learning process, leading to more efficient and effective learning. Our method has shown promising results in multi-robot learning from demonstrations and multi-robot reinforcement learning. It has the potential to revolutionize the way we design and implement distributed control

policies for multi-robot systems, enabling them to learn and adapt to complex tasks in a more efficient and scalable way.

- **Control techniques for power systems**: We addressed critical challenges in power systems (Chapter 7) by developing an automatic data-driven method for learning optimal power-temperature dynamics. This method allows for the accurate characterization of power converters under varying environmental conditions, which is essential for ensuring the reliable operation of robots in real-world scenarios. We also studied Gaussian Process Regression for predicting battery profiles in photovoltaic systems, enabling the prediction of potential failures in the energy supply. Furthermore, we designed a novel nonlinear implementable control for a dual active bridge series resonant converter, which offers fast and stable energy supply for autonomous systems.

8.2 Limitations and future work

While the research conducted in the thesis has made substantial progress in distributed multi-robot control, some limitations pave the way for future research directions:

- **Robustness and applicability of Implicit Control**: Implicit Control, as explained in Chapter 2, relies on the accurate estimation of system dynamics. Future research could explore robust variations of Implicit Control to handle uncertainties and disturbances more effectively. This could involve incorporating uncertainty models into the control design to adjust the control policy in response to unexpected changes in the system dynamics. Another promising research direction is to exploit the generality of Implicit Control to foster the integration of nonlinear control techniques and learning-based models. In this sense, the neural network can be viewed as an input-nonaffine dynamical model that can be queried to compute an Implicit Control policy, regardless of the underlying complexity of the neural model. Finally, the methods derived in Chapter 3, despite promising, have not been fully integrated in a multi-robot team to achieve effective distributed input-nonaffine control. Therefore, another exciting research direction is to explore how to bring together the theoretical insights on Implicit Control, distributed optimization, consensus and dissensus to develop novel distributed input-nonaffine control techniques.

- **Extending ellipsoidal methods**: The ellipsoidal methods we developed in Chapter 4 primarily focus on stochastic estimation of linear Gaussian systems. Extending these methods to nonlinear and non-Gaussian systems would broaden their applicability to a wider range of real-world scenarios. This could involve exploring alternative geometric representations or developing new optimization techniques that can handle nonlinear and non-Gaussian distributions. On the other hand, the fundamental research on distributed ellipsoidal methods described in Chapter 5 focuses on outer Löwner-John methods; however, there exist many other ellipsoidal methods, with applications such as safe path planning or clustering, that are worth to investigate.

- **Real-world validation of physics-informed policy parameterization**: The physics-informed policy parameterization has shown promising results in simulation and initial real-world experiments, both in multi-agent reinforcement learning and imitation learning settings (Chapter 6). However, further validation on a wider range of robotic platforms and in more complex real-world scenarios is necessary to fully assess its potential and identify any limitations. This could involve conducting experiments with different types of robots, varying environmental conditions, and more complex task objectives. Complementary to this objective, it remains to be explored alternative policy parameterizations that do not rely on port-Hamiltonian systems but, still, guarantee scalability and sample efficiency by enforcing sparse structures in the policy by construction.
- **Comprehensive energy management solutions**: Our work on power systems explored in Chapter 7 has focused on specific applications that are not necessarily close to robotic platforms. Future research could explore more comprehensive energy management solutions for multi-robot systems, considering factors such as energy harvesting, wireless power transfer, and energy-aware task allocation. This could involve developing integrated energy management frameworks that optimize energy usage across the entire multi-robot system, taking into account the energy requirements of individual robots, the availability of energy sources, and the overall task objectives.

The quest for scalable and accurate distributed multi-robot control is a continuous journey. As technology progresses and new challenges surface, the ideas and contributions presented in this thesis lay the groundwork for future innovations. By persistently exploring the interplay between physics, geometry, and learning, we can unlock the full potential of multi-robot systems and pave the way for their transformative impact across diverse domains.

Bibliography

[Adil et al., 2021] Adil, M., Tavakkol, S., and Madani, R. (2021). Rapid convergence of first-order numerical algorithms via adaptive conditioning. *arXiv preprint arXiv:2103.00736*.

[Aldana-López et al., 2023] Aldana-López, R., Sebastián, E., Aragüés, R., Montijano, E., and Sagüés, C. (2023). Distributed outer approximation of the intersection of ellipsoids. *IEEE Control Systems Letters*.

[Aldana-López et al., 2019] Aldana-López, R., Gómez-Gutiérrez, D., Defoort, M., Sánchez-Torres, J. D., and Muñoz-Vázquez, A. J. (2019). A class of robust consensus algorithms with predefined-time convergence under switching topologies. *International Journal of Robust and Nonlinear Control*, 29(17):6179–6198.

[Alexopoulos et al., 2017] Alexopoulos, A., Kirsch, B., and Badreddin, E. (2017). Realization of pursuit-evasion games with unmanned aerial vehicles. In *International Conference on Unmanned Aircraft Systems*, pages 797–805.

[Alonso-Mora et al., 2018] Alonso-Mora, J., Beardsley, P., and Siegwart, R. (2018). Cooperative collision avoidance for nonholonomic robots. *IEEE Transactions on Robotics*, 34(2):404–420.

[Alonso-Mora et al., 2013] Alonso-Mora, J., Breitenmoser, A., Rufli, M., Beardsley, P., and Siegwart, R. (2013). Optimal reciprocal collision avoidance for multiple non-holonomic robots. In *Distributed autonomous robotic systems: The 10th international symposium*, pages 203–216. Springer.

[Altafini, 2012] Altafini, C. (2012). Consensus problems on networks with antagonistic interactions. *IEEE Transactions on Automatic Control*, 58(4):935–946.

[Anderson and Moore, 2012] Anderson, B. D. and Moore, J. B. (2012). *Optimal filtering*. Courier Corporation.

[Anderson et al., 2017] Anderson, J. A., Gammeter, C., Schrittwieser, L., and Kolar, J. W. (2017). Accurate calorimetric switching loss measurement for 900 V 10 mΩ SiC mosfets. *IEEE Transactions on Power Electronics*, 32(12):8963–8968.

[Annaswamy et al., 2023] Annaswamy, A. M., Johansson, K. H., Pappas, G. J., et al. (2023). Control for societal-scale challenges: Road map 2030. *IEEE Control Systems Society*.

[ANSYS, 2022] ANSYS (2022). Ansys Icepak - Cooling Simulation Software for Electronic Components. `www.ansys.com/products/electronics/ansys-icepak`. [Online; accessed 26-September-2022].

[Antonelli et al., 2008] Antonelli, G., Arrichiello, F., and Chiaverini, S. (2008). The entrapment/escorting mission. *IEEE Robotics Automation Magazine*, 15(1):22–29.

[Antonelli and Chiaverini, 2006] Antonelli, G. and Chiaverini, S. (2006). Kinematic control of platoons of autonomous vehicles. *IEEE Transactions on Robotics*, 22(6):1285–1292.

[Anurag et al., 2020] Anurag, A., Acharya, S., and Bhattacharya, S. (2020). An accurate calorimetric loss measurement method for SiC MOSFETs. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 8(2):1644–1656.

[Auletta et al., 2022] Auletta, F., Fiore, D., Richardson, M. J., and di Bernardo, M. (2022). Herding stochastic autonomous agents via local control rules and online target selection strategies. *Autonomous Robots*, pages 1–13.

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

[Azpúrua et al., 2022] Azpúrua, M. A., Pous, M., and Silva, F. (2022). Time- and frequency-domain characterization of switching losses in GaN FETs power converters. *IEEE Transactions on Power Electronics*, 37(3):3219–3232.

[Azurza Anderson et al., 2021] Azurza Anderson, J., Zulauf, G., Papamanolis, P., Hobi, S., Mirić, S., and Kolar, J. W. (2021). Three levels are not enough: Scaling laws for multilevel converters in AC/DC applications. *IEEE Transactions on Power Electronics*, 36(4):3967–3986.

[Bahman et al., 2017] Bahman, A. S., Ma, K., and Blaabjerg, F. (2017). A lumped thermal model including thermal coupling and thermal boundary conditions for high-power IGBT modules. *IEEE Transactions on Power Electronics*, 33(3):2518–2530.

[Bai et al., 2010] Bai, H., Freeman, R. A., and Lynch, K. M. (2010). Robust dynamic average consensus of time-varying inputs. In *IEEE Conference on Decision and Control*, pages 3104–3109.

[Bajcsy et al., 2019] Bajcsy, A., Herbert, S. L., Fridovich-Keil, D., Fisac, J. F., Deglurkar, S., Dragan, A. D., and Tomlin, C. J. (2019). A scalable framework for real-time multi-robot, multi-human collision avoidance. In *IEEE International Conference on Robotics and Automation*, pages 936–943.

[Barber et al., 1996] Barber, C. B., Dobkin, D. P., and Huhdanpaa, H. (1996). The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483.

[Bareiss and Van den Berg, 2015] Bareiss, D. and Van den Berg, J. (2015). Generalized reciprocal collision avoidance. *The International Journal of Robotics Research*, 34(12):1501–1514.

[Baroudi et al., 2007] Baroudi, J. A., Dinavahi, V., and Knight, A. M. (2007). A review of power converter topologies for wind generators. *Renewable Energy*, 32(14):2369–2385.

[Bastianello et al., 2020] Bastianello, N., Carli, R., Schenato, L., and Todescato, M. (2020). Asynchronous distributed optimization over lossy networks via relaxed ADMM: Stability and linear convergence. *IEEE Transactions on Automatic Control*, 66(6):2620–2635.

[Bastianello et al., 2022] Bastianello, N., Schenato, L., and Carli, R. (2022). A novel bound on the convergence rate of ADMM for distributed optimization. *Automatica*, 142:110403.

[Battilotti et al., 2021] Battilotti, S., Cacace, F., and d'Angelo, M. (2021). A stability with optimality analysis of consensus-based distributed filters for discrete-time linear systems. *Automatica*, 129:109589.

[Battistelli et al., 2012] Battistelli, G., Benavoli, A., and Chisci, L. (2012). Data-driven communication for state estimation with sensor networks. *Automatica*, 48(5):926–935.

[Battistelli et al., 2019] Battistelli, G., Chisci, L., Gao, L., and Selvi, D. (2019). Event-triggered distributed Bayes filter. In *IEEE European Control Conference*, pages 2731–2736.

[Battistelli et al., 2014] Battistelli, G., Chisci, L., Mugnai, G., Farina, A., and Graziano, A. (2014). Consensus-based linear and nonlinear filtering. *IEEE Transactions on Automatic Control*, 60(5):1410–1415.

[Battistelli et al., 2018] Battistelli, G., Chisci, L., and Selvi, D. (2018). A distributed Kalman filter with event-triggered communication and guaranteed stability. *Automatica*, 93:75–82.

[Battisti and Muradore, 2020] Battisti, T. and Muradore, R. (2020). A velocity obstacles approach for autonomous landing and teleoperated robots. *Autonomous Robots*, 44(2):217–232.

[Bertsekas, 1999] Bertsekas, D. P. (1999). *Nonlinear Programming*. Athena Scientific.

[Bertsekas, 2011] Bertsekas, D. P. (2011). Incremental proximal methods for large scale convex optimization. *Mathematical programming*, 129(2):163–195.

[Bettini et al., 2022] Bettini, M., Kortvelesy, R., Blumenkamp, J., and Prorok, A. (2022). VMAS: a vectorized multi-agent simulator for collective robot learning. *arXiv preprint arXiv:2207.03530*.

[Bin et al., 2022] Bin, M., Notarnicola, I., and Parisini, T. (2022). Stability, linear convergence, and robustness of the wang-elia algorithm for distributed consensus optimization. In *IEEE Conference on Decision and Control*, pages 1610–1615.

[Bizyaeva et al., 2022] Bizyaeva, A., Franci, A., and Leonard, N. E. (2022). Nonlinear opinion dynamics with tunable sensitivity. *IEEE Transactions on Automatic Control*.

[Blankenstein et al., 2002] Blankenstein, G., Ortega, R., and Van Der Schaft, A. J. (2002). The matching conditions of controlled lagrangians and ida-passivity based control. *International Journal of Control*, 75(9):645–665.

[Bloembergen et al., 2015] Bloembergen, D., Tuyls, K., Hennes, D., and Kaisers, M. (2015). Evolutionary dynamics of multi-agent learning: A survey. *Journal of Artificial Intelligence Research*, 53:659–697.

[Bogert and Doshi, 2018] Bogert, K. and Doshi, P. (2018). Multi-robot inverse reinforcement learning under occlusion with estimation of state transitions. *Artificial Intelligence*, 263:46–73.

[Bohm et al., 2004] Bohm, C., Hauck, T., Rudnyi, E., and Korvink, J. (2004). Compact electro-thermal models of semiconductor devices with multiple heat sources. In *International Conference on Thermal and Mechanical Simulation and Experiments in Microelectronics and Microsystems*, pages 101–104.

[Boldrer et al., 2023] Boldrer, M., Serra-Gomez, A., Lyons, L., Alonso-Mora, J., and Ferranti, L. (2023). Rule-based lloyd algorithm for multi-robot motion planning and control with safety and convergence guarantees. *arXiv preprint arXiv:2310.19511*.

[Boyd et al., 1994] Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V. (1994). *Linear matrix inequalities in system and control theory*. SIAM.

[Boyd et al., 2006] Boyd, S., Ghosh, A., Prabhakar, B., and Shah, D. (2006). Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530.

[Bredies et al., 2022] Bredies, K., Chenchene, E., and Naldi, E. (2022). Graph and distributed extensions of the douglas-rachford method. *arXiv preprint arXiv:2211.04782*.

[Brito et al., 2021] Brito, B., Everett, M., How, J. P., and Alonso-Mora, J. (2021). Where to go next: Learning a subgoal recommendation policy for navigation in dynamic environments. *IEEE Robotics and Automation Letters*, 6(3):4616–4623.

[Brito et al., 2019] Brito, B., Floor, B., Ferranti, L., and Alonso-Mora, J. (2019). Model predictive contouring control for collision avoidance in unstructured dynamic environments. *IEEE Robotics and Automation Letters*, 4(4):4459–4466.

[Brock and Khatib, 1999] Brock, O. and Khatib, O. (1999). High-speed navigation using the global dynamic window approach. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 341–346.

[Buffolo et al., 2024] Buffolo, M., Favero, D., Marcuzzi, A., Santi, C. D., Meneghesso, G., Zanoni, E., and Meneghini, M. (2024). Review and outlook on gan and sic power devices: Industrial state-of-the-art, applications, and perspectives. *IEEE Transactions on Electron Devices*, pages 1–12.

[Bullo et al., 2009] Bullo, F., Cortés, J., and Martinez, S. (2009). *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*, volume 27. Princeton University Press.

[Bürger et al., 2013] Bürger, M., Notarstefano, G., and Allgöwer, F. (2013). A polyhedral approximation framework for convex and robust distributed optimization. *IEEE Transactions on Automatic Control*, 59(2):384–395.

[Buşoniu et al., 2010] Buşoniu, L., Babuška, R., and De Schutter, B. (2010). Multi-agent reinforcement learning: An overview. *Innovations in multi-agent systems and applications-1*, pages 183–221.

[Carnevale et al., 2023a] Carnevale, G., Bastianello, N., Carli, R., and Notarstefano, G. (2023a). Distributed consensus optimization via ADMM-tracking gradient. In *IEEE Conference on Decision and Control*, pages 290–295.

[Carnevale et al., 2023b] Carnevale, G., Bastianello, N., Notarstefano, G., and Carli, R. (2023b). ADMM-Tracking Gradient for Distributed Optimization over Asynchronous and Unreliable Networks. *arXiv:2309.14142*.

[Carnevale et al., 2024] Carnevale, G., Fabiani, F., Fele, F., Margellos, K., and Notarstefano, G. (2024). Tracking-based distributed equilibrium seeking for aggregative games. *IEEE Transactions on Automatic Control*.

[Carnevale et al., 2023c] Carnevale, G., Notarnicola, I., Marconi, L., and Notarstefano, G. (2023c). Triggered gradient tracking for asynchronous distributed optimization. *Automatica*, 147:110726.

[Cathcart et al., 2023] Cathcart, C., Santos, M., Park, S., and Leonard, N. E. (2023). Proactive opinion-driven robot navigation around human movers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4052–4058.

[Cattivelli and Sayed, 2010] Cattivelli, F. S. and Sayed, A. H. (2010). Diffusion strategies for distributed Kalman filtering and smoothing. *IEEE Transactions on Automatic Control*, 55(9):2069–2084.

[Chalupka et al., 2013] Chalupka, K., Williams, C. K., and Murray, I. (2013). A framework for evaluating approximation methods for gaussian process regression. *Journal of Machine Learning Research*, 14(1).

[Chang et al., 2023] Chang, C.-W., Zhao, X., Phukan, R., Burgos, R., Uicich, S., Asfaux, P., and Dong, D. (2023). Thermal consideration and design for a 200-kw sic-based high-density three-phase inverter in more electric aircraft. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 11(6):5910–5929.

[Chen et al., 2019] Chen, C., Liu, Y., Kreiss, S., and Alahi, A. (2019). Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. In *IEEE International Conference on Robotics and Automation*, pages 6015–6022.

[Chen et al., 2018] Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31.

[Chen et al., 1990] Chen, S., Billings, S. A., and Grant, P. (1990). Non-linear system identification using neural networks. *International Journal of Control*, 51(6):1191–1214.

[Cheng et al., 2020] Cheng, T., Dah-Chuanlu, D., and Siwakoti, Y. P. (2020). Electro-thermal average modeling of a boost converter considering device self-heating. *IEEE Applied Power Electronics Conference and Exposition*, pages 2854–2859.

[Chipade and Panagou, 2019] Chipade, V. S. and Panagou, D. (2019). Herding an adversarial swarm in an obstacle environment. In *IEEE Conference on Decision and Control*, pages 5859–5864.

[Choset et al., 2005] Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., and Burgard, W. (2005). *Principles of robot motion: theory, algorithms, and implementations*. MIT press.

[Christensen et al., 2021] Christensen, H., Amato, N., Yanco, H., Mataric, M., Choset, H., Drobnis, A., Goldberg, K., Grizzle, J., Hager, G., Hollerbach, J., et al. (2021). A roadmap for US robotics–from internet to robotics 2020 edition. *Foundations and Trends® in Robotics*, 8(4):307–424.

[Chung et al., 2011] Chung, T. H., Hollinger, G. A., and Isler, V. (2011). Search and pursuit-evasion in mobile robotics: A survey. *Autonomous robots*, 31:299–316.

[Cisneros-Velarde et al., 2020] Cisneros-Velarde, P., Chan, K. S., and Bullo, F. (2020). Polarization and fluctuations in signed social networks. *IEEE Transactions on Automatic Control*, 66(8):3789–3793.

[Coday and Pilawa-Podgurski, 2023] Coday, S. and Pilawa-Podgurski, R. C. N. (2023). Characterization and modeling of ceramic capacitor losses under large signal operating conditions. *IEEE Open Journal of Power Electronics*, 4:24–33.

[Cong et al., 2020] Cong, X., Zhang, C., Jiang, J., Zhang, W., and Jiang, Y. (2020). A hybrid method for the prediction of the remaining useful life of lithium-ion batteries with accelerated capacity degradation. *IEEE Transactions on Vehicular Technology*, 69(11):12775–12785.

[Cortes, 2008] Cortes, J. (2008). Discontinuous dynamical systems. *IEEE Control Systems Magazine*, 28(3):36–73.

[Cortés et al., 2008] Cortés, P., Kazmierkowski, M. P., Kennel, R. M., Quevedo, D. E., and Rodríguez, J. (2008). Predictive control in power electronics and drives. *IEEE Transactions on industrial electronics*, 55(12):4312–4324.

[Cruz-Duarte et al., 2021] Cruz-Duarte, J. M., Avina-Cervantes, J. G., Garcia-Perez, A., Andrade-Lucio, J. A., Correa, R., and Morega, A. M. (2021). Novel calorimetric approach for thermal analysis of microelectronic devices. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 11(9):1442–1451.

[Cui et al., 2023] Cui, K., Li, M., Fabian, C., and Koeppl, H. (2023). Scalable task-driven robotic swarm control via collision avoidance and learning mean-field control. In *IEEE International Conference on Robotics and Automation*, pages 1192–1199.

[Cuomo et al., 2022] Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., and Piccialli, F. (2022). Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):88.

[Das and Moura, 2016] Das, S. and Moura, J. M. (2016). Consensus+Innovations distributed Kalman filter with optimized gains. *IEEE Transactions on Signal Processing*, 65(2):467–481.

[Dasari et al., 2020] Dasari, S., Ebert, F., Tian, S., Nair, S., Bucher, B., Schmeckpeper, K., Singh, S., Levine, S., and Finn, C. (2020). Robonet: Large-scale multi-robot learning. In *Conference on Robot Learning*, pages 885–897.

[De Doncker et al., 1991] De Doncker, R. W., Divan, D. M., and Kheraluwala, M. H. (1991). A three-phase soft-switched high-power-density DC/DC converter for high-power applications. *IEEE Transactions on Industry Applications*, 27(1):63–73.

[de Groot et al., 2023] de Groot, O., Ferranti, L., Gavrila, D., and Alonso-Mora, J. (2023). Scenario-based motion planning with bounded probability of collision. *arXiv preprint arXiv:2307.01070*.

[de Groot et al., 2024] de Groot, O., Ferranti, L., Gavrila, D., and Alonso-Mora, J. (2024). Topology-driven parallel trajectory optimization in dynamic environments. *arXiv preprint arXiv:2401.06021*.

[De Witt et al., 2020] De Witt, C. S., Gupta, T., Makoviichuk, D., Makoviychuk, V., Torr, P. H., Sun, M., and Whiteson, S. (2020). Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*.

[Deits and Tedrake, 2015] Deits, R. and Tedrake, R. (2015). Computing large convex regions of obstacle-free space through semidefinite programming. In *Algorithmic Foundations of Robotics XI: Selected Contributions of the Eleventh International Workshop on the Algorithmic Foundations of Robotics*, pages 109–124. Springer.

[Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

[Deng et al., 2012] Deng, Z., Zhang, P., Qi, W., Liu, J., and Gao, Y. (2012). Sequential covariance intersection fusion Kalman filter. *Information Science*, 189:293–309.

[Deplano et al., 2022] Deplano, D., Franceschelli, M., and Giua, A. (2022). Dynamic max-consensus with local self-tuning. *IFAC-PapersOnLine*, 55(13):127–132.

[Desouky and Schwartz, 2010] Desouky, S. F. and Schwartz, H. M. (2010). Learning in n-pursuer n-evader differential games. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 4069–4074.

[Dong et al., 2020] Dong, H., Weng, C.-Y., Guo, C., Yu, H., and Chen, I.-M. (2020). Real-time avoidance strategy of dynamic obstacles via half model-free detection and tracking with 2d lidar for mobile robots. *IEEE/ASME transactions on mechatronics*, 26(4):2215–2225.

[Donge et al., 2023] Donge, V. S., Lian, B., Lewis, F. L., and Davoudi, A. (2023). Accelerated reinforcement learning via dynamic mode decomposition. *IEEE Transactions on Control of Network Systems*.

[Du et al., 2017] Du, S. L., Sun, X. M., Cao, M., and Wang, W. (2017). Pursuing an evader through cooperative relaying in multi-agent surveillance networks. *Automatica*, 83:155–161.

[Duvenaud, 2014] Duvenaud, D. (2014). *Automatic model construction with Gaussian processes*. PhD thesis, University of Cambridge.

[Eberly, 2011] Eberly, D. (2011). Distance from a point to an ellipse, an ellipsoid, or a hyperellipsoid. *Geometric Tools, LLC*.

[El Mejdoubi et al., 2015] El Mejdoubi, A., Oukaour, A., Chaoui, H., Gualous, H., Sabor, J., and Slamani, Y. (2015). State-of-charge and state-of-health lithium-ion batteries' diagnosis according to surface temperature variation. *IEEE Transactions on Industrial Electronics*, 63(4):2391–2402.

[Eleffendi and Johnson, 2016] Eleffendi, M. A. and Johnson, C. M. (2016). Application of kalman filter to estimate junction temperature in IGBT power modules. *IEEE Transactions on Power Electronics*, 31(2):1576–1587.

[Engel et al., 2012] Engel, S. P., Soltau, N., Stagge, H., and De Doncker, R. W. (2012). Dynamic and balanced control of three-phase high-power dual-active bridge DC-DC converters in DC-grid applications. *IEEE Transactions on Power Electronics*, 28(4):1880–1889.

[Entzminger et al., 2024] Entzminger, C., Qiao, W., Qu, L., and Hudgins, J. L. (2024). Automated extraction of low-order thermal model with controllable error bounds for sic mosfet power modules. *IEEE Transactions on Power Electronics*, 39(1):538–551.

[Eppenberger et al., 2020] Eppenberger, T., Cesari, G., Dymczyk, M., Siegwart, R., and Dubé, R. (2020). Leveraging stereo-camera data for real-time dynamic obstacle detection and tracking. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 10528–10535.

[Erickson and Maksimovic, 2007] Erickson, R. W. and Maksimovic, D. (2007). *Fundamentals of power electronics*. Springer Science & Business Media.

[Everett et al., 2021] Everett, M., Chen, Y. F., and How, J. P. (2021). Collision avoidance in pedestrian-rich environments with deep reinforcement learning. *IEEE Access*, 9:10357–10377.

[Fagnani and Zampieri, 2009] Fagnani, F. and Zampieri, S. (2009). Average consensus with packet drop communication. *SIAM Journal on Control and Optimization*, 48(1):102–133.

[Fan et al., 2020] Fan, T., Long, P., Liu, W., and Pan, J. (2020). Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios. *The International Journal of Robotics Research*, 39(7):856–892.

[Feng et al., 2012] Feng, W., Lee, F. C., and Mattavelli, P. (2012). Simplified optimal trajectory control (SOTC) for LLC resonant converters. *IEEE Transactions on Power Electronics*, 28(5):2415–2426.

[Ferrer et al., 2013] Ferrer, G., Garrell, A., and Sanfeliu, A. (2013). Robot companion: A social-force based approach with human awareness-navigation in crowded environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1688–1694.

[Filippov, 1988] Filippov, A. F. (1988). *Differential equations with discontinuous righthand sides*, volume 18 of *Mathematics and its Applications (Soviet Series)*. Kluwer Academic Publishers Group, Dordrecht.

[Fiorini and Shiller, 1993] Fiorini, P. and Shiller, Z. (1993). Motion planning in dynamic environments using the relative velocity paradigm. In *IEEE International Conference on Robotics and Automation*, pages 560–565.

[Fiorini and Shiller, 1998] Fiorini, P. and Shiller, Z. (1998). Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7):760–772.

[Foster, 1924] Foster, R. M. (1924). A Reactance Theorem. *Bell System Technical Journal*, 3(2):259–267.

[Franceschelli and Gasparri, 2013] Franceschelli, M. and Gasparri, A. (2013). Gossip-based centroid and common reference frame estimation in multiagent systems. *IEEE Transactions on Robotics*, 30(2):524–531.

[Franceschelli and Gasparri, 2016] Franceschelli, M. and Gasparri, A. (2016). Multi-stage discrete time dynamic average consensus. In *IEEE Conference on Decision and Control*, pages 897–903.

[Franceschelli and Gasparri, 2019] Franceschelli, M. and Gasparri, A. (2019). Multi-stage discrete time and randomized dynamic average consensus. *Automatica*, 99:69–81.

[Franceschelli et al., 2013] Franceschelli, M., Gasparri, A., Giua, A., and Seatzu, C. (2013). Decentralized estimation of laplacian eigenvalues in multi-agent systems. *Automatica*, 49(4):1031–1036.

[Franceschelli et al., 2020] Franceschelli, M., Pilloni, A., and Gasparri, A. (2020). Multi-agent coordination of thermostatically controlled loads by smart power sockets for electric demand side management. *IEEE Transactions on Control Systems Technology*, 29(2):731–743.

[Franchi et al., 2010] Franchi, A., Stegagno, P., Rocco, M. D., and Oriolo, G. (2010). Distributed target localization and encirclement with a multi-robot system. In *IFAC Proceedings Volumes*, volume 43, pages 151–156.

[Fu et al., 2020] Fu, A., Zhang, J., and Boyd, S. (2020). Anderson accelerated douglas–rachford splitting. *SIAM Journal on Scientific Computing*, 42(6):A3560–A3583.

[Furieri et al., 2022a] Furieri, L., Galimberti, C. L., Zakwan, M., and Ferrari-Trecate, G. (2022a). Distributed neural network control with dependability guarantees: a compositional port-hamiltonian approach. In *Learning for Dynamics and Control Conference*, pages 571–583. PMLR.

[Furieri et al., 2022b] Furieri, L., Galimberti, C. L., Zakwan, M., and Ferrari-Trecate, G. (2022b). Distributed neural network control with dependability guarantees: a compositional port-hamiltonian approach. In *Learning for Dynamics and Control Conference*, pages 571–583.

[Gama et al., 2022] Gama, F., Li, Q., Tolstaya, E., Prorok, A., and Ribeiro, A. (2022). Synthesizing decentralized controllers with graph neural networks and imitation learning. *IEEE Transactions on Signal Processing*, 70:1932–1946.

[Gao et al., 2018] Gao, S., Song, R., and Li, Y. (2018). Cooperative control of multiple nonholonomic robots for escorting and patrolling mission based on vector field. *IEEE Access*, 6:41883–41891.

[Garin and Schenato, 2010] Garin, F. and Schenato, L. (2010). A survey on distributed estimation and control applications using linear consensus algorithms. In *Networked control systems*, pages 75–107. Springer.

[George and Freeman, 2019] George, J. and Freeman, R. A. (2019). Robust dynamic average consensus algorithms. *IEEE Transactions on Automatic Control*, 64(11):4615–4622.

[Gharbi et al., 2023] Gharbi, I., Kuckling, J., Ramos, D. G., and Birattari, M. (2023). Show me what you want: Inverse reinforcement learning to automatically design robot swarms by demonstration. *arXiv preprint arXiv:2301.06864*.

[Ghosh et al., 1996] Ghosh, B., Muthukrishnan, S., and Schultz, M. H. (1996). First and second order diffusive methods for rapid, coarse, distributed load balancing. In *ACM symposium on Parallel algorithms and architectures*, pages 72–81.

[Graves and Schmidhuber, 2005] Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

[Grewal and Andrews, 2014] Grewal, M. S. and Andrews, A. P. (2014). *Kalman filtering: Theory and Practice with MATLAB*. John Wiley & Sons.

[Guacci et al., 2020] Guacci, M., Azurza Anderson, J., Pally, K. L., Bortis, D., Kolar, J. W., Kasper, M. J., Sanchez, J., and Deboy, G. (2020). Experimental characterization of silicon and gallium nitride 200 V power semiconductors for modular-multi-level converters using advanced measurement techniques. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 8(3):2238–2254.

[Haarnoja et al., 2018a] Haarnoja, T., Ha, S., Zhou, A., Tan, J., Tucker, G., and Levine, S. (2018a). Learning to walk via deep reinforcement learning. *arXiv preprint arXiv:1812.11103*.

[Haarnoja et al., 2018b] Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018b). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR.

[Haarnoja et al., 2018c] Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. (2018c). Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.

[Han et al., 2013] Han, D., Mo, Y., Wu, J., Sinopoli, B., and Shi, L. (2013). Stochastic event-triggered sensor scheduling for remote state estimation. In *IEEE Conference on Decision and Control*, pages 6079–6084.

[Han et al., 2020] Han, R., Chen, S., and Hao, Q. (2020). Cooperative multi-robot navigation in dynamic environment with deep reinforcement learning. In *IEEE International Conference on Robotics and Automation*, pages 448–454.

[Han et al., 2022] Han, R., Chen, S., Wang, S., Zhang, Z., Gao, R., Hao, Q., and Pan, J. (2022). Reinforcement learned distributed multi-robot navigation with reciprocal velocity obstacle shaped rewards. *IEEE Robotics and Automation Letters*, 7(3):5896–5903.

[Hauck and Bohm, 2000] Hauck, T. and Bohm, T. (2000). Thermal RC-network approach to analyze multichip power packages. In *IEEE Semiconductor Thermal Measurement and Management Symposium*, pages 227–234.

[Hauswirth et al., 2016] Hauswirth, A., Bolognani, S., Hug, G., and Dörfler, F. (2016). Projected gradient descent on riemannian manifolds with applications to online power system optimization. In *Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 225–232.

[Heintzman et al., 2021] Heintzman, L., Hashimoto, A., Abaid, N., and Williams, R. K. (2021). Anticipatory planning and dynamic lost person models for human-robot search and rescue. In *IEEE International Conference on Robotics and Automation*, pages 8252–8258.

[Henrion and Lasserre, 2004] Henrion, D. and Lasserre, J.-B. (2004). Solving nonconvex optimization problems. *IEEE Control Systems Magazine*, 24(3):72–83.

[Henrion and Lasserre, 2011] Henrion, D. and Lasserre, J.-B. (2011). Inner approximations for polynomial matrix inequalities and robust stability regions. *IEEE Transactions on Automatic Control*, 57(6):1456–1467.

[Henrion et al., 2001] Henrion, D., Tarbouriech, S., and Arzelier, D. (2001). LMI approximations for the radius of the intersection of ellipsoids: Survey. *Journal of Optimization Theory and Applications*, 108(1):1–28.

[Herrera et al., 2013] Herrera, L., Li, C., Yao, X., Jiao, D., and Wang, J. (2013). FPGA based real time electro-thermal modeling of power electronic converters. *IEEE Applied Power Electronics Conference and Exposition*, pages 1725–1729.

[Ho and Ermon, 2016] Ho, J. and Ermon, S. (2016). Generative adversarial imitation learning. *Advances in neural information processing systems*, 29.

[Hu and Biela, 2023] Hu, A. and Biela, J. (2023). Fast and accurate data sheet based analytical turn-on switching loss model for a SiC MOSFET and Schottky Diode half-bridge. In *European Conference on Power Electronics and Applications*, pages 1–11.

[Hu et al., 2014] Hu, J., Hu, J., Lin, H., Li, X., Jiang, C., Qiu, X., and Li, W. (2014). State-of-charge estimation for battery management system using optimized support vector machine for regression. *Journal of Power Sources*, 269:682–693.

[Hu et al., 2011] Hu, J., Xie, L., and Zhang, C. (2011). Diffusion Kalman filtering based on covariance intersection. *IEEE Transactions on Signal Processing*, 60(2):891–902.

[Huang et al., 2024] Huang, K., Pu, S., and Nedić, A. (2024). An accelerated distributed stochastic gradient method with momentum. *arXiv:2402.09714*.

[Instruments, 2018] Instruments, T. (2018). CLA math library user guide.

[Iqbal and Sha, 2019] Iqbal, S. and Sha, F. (2019). Actor-attention-critic for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 2961–2970. PMLR.

[Jahn et al., 2017] Jahn, A., Alitappeh, R. J., Saldaña, D., Pimenta, L. C. A., Santos, A. G., and Campos, M. F. M. (2017). Distributed multi-robot coordination for dynamic perimeter surveillance in uncertain environments. In *IEEE International Conference on Robotics and Automation*, pages 273–278.

[Jakovetić et al., 2014] Jakovetić, D., Xavier, J., and Moura, J. M. (2014). Fast distributed gradient methods. *IEEE Transactions on Automatic Control*, 59(5):1131–1146.

[Jekel et al., 2018] Jekel, C. F., Venter, G., Venter, M. P., Stander, N., and Haftka, R. T. (2018). Similarity measures for identifying material parameters from hysteresis loops using inverse analysis. *International Journal of Material Forming*, 12:355–378.

[Jeung and Lee, 2018] Jeung, Y.-C. and Lee, D.-C. (2018). Voltage and current regulations of bidirectional isolated dual-active-bridge DC-DC converters based on a double-integral sliding mode control. *IEEE Transactions on Power Electronics*, 34(7):6937–6946.

[Jia et al., 2015] Jia, P., MirTabatabaei, A., Friedkin, N. E., and Bullo, F. (2015). Opinion dynamics and the evolution of social power in influence networks. *SIAM review*, 57(3):367–397.

[Jiahao et al., 2022] Jiahao, T. Z., Pan, L., and Hsieh, M. A. (2022). Learning to swarm with knowledge-based neural ordinary differential equations. In *IEEE International Conference on Robotics and Automation*, pages 6912–6918.

[Jiang et al., 2018] Jiang, W., Wen, G., Peng, Z., Huang, T., and Rahmani, A. (2018). Fully distributed formation-containment control of heterogeneous linear multiagent systems. *IEEE Transactions on Automatic Control*, 64(9):3889–3896.

[John, 2014] John, F. (2014). Extremum problems with inequalities as subsidiary conditions. In *Traces and emergence of nonlinear programming*, pages 197–215. Springer.

[Julier and Uhlmann, 2017] Julier, S. and Uhlmann, J. K. (2017). General decentralized data fusion with covariance intersection. In *Handbook of multisensor data fusion*, pages 339–364. CRC Press.

[Julier and Uhlmann, 1997] Julier, S. J. and Uhlmann, J. K. (1997). A non-divergent estimation algorithm in the presence of unknown correlations. In *IEEE ACC*, volume 4, pages 2369–2373.

[Julier and Uhlmann, 2001] Julier, S. J. and Uhlmann, J. K. (2001). General decentralized data fusion with covariance intersection. *Handbook of multisensor data fusion*.

[Kalman, 1960] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *ASME Journal of Basic Engineering*, 82(1):35–45.

[Kamal et al., 2013] Kamal, A. T., Farrell, J. A., and Roy-Chowdhury, A. K. (2013). Information weighted consensus filters and their application in distributed camera networks. *IEEE Transactions on Automatic Control*, 58(12):3112–3125.

[Katyal et al., 2020] Katyal, K. D., Hager, G. D., and Huang, C.-M. (2020). Intent-aware pedestrian prediction for adaptive crowd navigation. In *IEEE International Conference on Robotics and Automation*, pages 3277–3283.

[Khalil, 2014] Khalil, H. K. (2014). *Nonlinear Control*. Pearson Education.

[Khan et al., 2020] Khan, A., Tolstaya, E., Ribeiro, A., and Kumar, V. (2020). Graph policy gradients for large scale robot control. In *Conference on Robot Learning*, pages 823–834.

[Khatib, 1986] Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–98.

[Kheraluwala et al., 1992] Kheraluwala, M., Gascoigne, R. W., Divan, D. M., and Baumann, E. D. (1992). Performance characterization of a high-power dual active bridge DC-to-DC converter. *IEEE Transactions on Industry Applications*, 28(6):1294–1301.

[Kia et al., 2019] Kia, S. S., Van Scoy, B., Cortes, J., Freeman, R. A., Lynch, K. M., and Martinez, S. (2019). Tutorial on dynamic average consensus: The problem, its applications, and the algorithms. *IEEE Control Systems Magazine*, 39(3):40–72.

[Kingma and Ba, 2015] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference for Learning Presentations*.

[Koch et al., 2019] Koch, D., Araujo, S., and Kallfass, I. (2019). Accuracy analysis of calorimetric loss measurement for benchmarking wide bandgap power transistors under soft-switching operation. *IEEE Workshop on Wide Bandgap Power Devices and Applications in Asia*, pages 6–11.

[Kohlhepp et al., 2021] Kohlhepp, B., Kübrich, D., and Dürbaum, T. (2021). Switching loss measurement – a thermal approach applied to GaN-half-bridge configuration keywords. *European Conference on Power Electronics and Applications*, pages 1–10.

[Kohonen, 1997] Kohonen, T. (1997). Exploration of very large databases by self-organizing maps. In *IEEE International Conference on Neural Networks*, volume 1, pages 1–6.

[Kokiopoulou and Frossard, 2008] Kokiopoulou, E. and Frossard, P. (2008). Polynomial filtering for fast convergence in distributed consensus. *IEEE Transactions on Signal Processing*, 57(1):342–354.

[Krismer and Kolar, 2009] Krismer, F. and Kolar, J. W. (2009). Accurate power loss model derivation of a high-current dual active bridge converter for an automotive application. *IEEE Transactions on Industrial Electronics*, 57(3):881–891.

[Kuba et al., 2021] Kuba, J. G., Chen, R., Wen, M., Wen, Y., Sun, F., Wang, J., and Yang, Y. (2021). Trust region policy optimisation in multi-agent reinforcement learning. In *International Conference on Learning Representations*.

[Kuprat et al., 2023] Kuprat, J., Debbadi, K., Schaumburg, J., Liserre, M., and Langwasser, M. (2023). Thermal digital twin of power electronics modules for online thermal parameter identification. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, pages 1–1.

[Laloya et al., 2016] Laloya, E., Lucía, O., Sarnago, H., and Burdío, J. M. (2016). Heat management in power converters: From state of the art to future ultrahigh efficiency systems. *IEEE Transactions on Power Electronics*, 31(11):7896–7908.

[Lasserre, 2015] Lasserre, J. B. (2015). A generalization of Löwner-John's ellipsoid theorem. *Mathematical Programming*, 152:559–591.

[Lee, 2003] Lee, J. M. (2003). *Introduction to Smooth Manifolds*. Springer.

[Li et al., 2020a] Li, G., Jiang, B., Zhu, H., Che, Z., and Liu, Y. (2020a). Generative attention networks for multi-agent behavioral modeling. In *AAAI Conference on Artificial Intelligence*, volume 34, pages 7195–7202.

[Li et al., 2020b] Li, Q., Gama, F., Ribeiro, A., and Prorok, A. (2020b). Graph neural networks for decentralized multi-robot path planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 11785–11792. IEEE.

[Li et al., 2021a] Li, Q., Lin, W., Liu, Z., and Prorok, A. (2021a). Message-aware graph attention networks for large-scale multi-robot path planning. *IEEE Robotics and Automation Letters*, 6(3):5533–5540.

[Li et al., 2019a] Li, R., Wu, X., Yang, S., and Sheng, K. (2019a). Dynamic on-state resistance test and evaluation of gan power devices under hard- and soft-switching conditions by double and multiple pulses. *IEEE Transactions on Power Electronics*, 34(2):1044–1053.

[Li et al., 2015] Li, W., Jia, Y., and Du, J. (2015). Event-triggered Kalman consensus filter over sensor networks. *IET Control Theory & Applications*, 10(1):103–110.

[Li et al., 2021b] Li, W., Zeng, X., Hong, Y., and Ji, H. (2021b). Distributed consensus-based solver for semi-definite programming: An optimization viewpoint. *Automatica*, 131:109737.

[Li and Bhat, 2009] Li, X. and Bhat, A. K. (2009). Analysis and design of high-frequency isolated dual-bridge series resonant DC/DC converter. *IEEE Transactions on Power Electronics*, 25(4):850–862.

[Li, 1993] Li, Y. (1993). Centering, trust region, reflective techniques for nonlinear minimization subject to bounds. Technical report, Cornell University.

[Li et al., 2019b] Li, Z., Shi, W., and Yan, M. (2019b). A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates. *IEEE Transactions on Signal Processing*, 67(17):4494–4506.

[Licitra et al., 2018] Licitra, R. A., Bell, Z. I., Doucette, E. A., and Dixon, W. E. (2018). Single agent indirect herding of multiple targets: A switched adaptive control approach. *IEEE Control System Letters*, 2(1):127–132.

[Likas et al., 2003] Likas, A., Vlassis, N., and Verbeek, J. J. (2003). The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461.

[Lin et al., 2012] Lin, H.-T., Liang, T.-J., and Chen, S.-M. (2012). Estimation of battery state of health using probabilistic neural network. *IEEE Transactions on Industrial Informatics*, 9(2):679–685.

[Lin et al., 2020] Lin, W., Niu, X., Gao, X., and Tao, G. (2020). Adaptive regulation of discrete-time nonaffine systems with parametric uncertainty. *IEEE Transactions on Automatic Control*, 66(5):2365–2371.

[Lippi et al., 2022] Lippi, M., Furchì, A., Marino, A., and Gasparri, A. (2022). An adaptive distributed protocol for finite-time infimum or supremum dynamic consensus. *IEEE Control Systems Letters*, 7:401–406.

[Liu and Sun, 2021] Liu, C. and Sun, S. (2021). Event-triggered optimal and suboptimal distributed Kalman consensus filters for sensor networks. *Journal of the Franklin Institute*, 358(9):5163–5183.

[Liu et al., 2012] Liu, D., Pang, J., Zhou, J., and Peng, Y. (2012). Data-driven prognostics for lithium-ion battery based on gaussian process regression. In *IEEE Prognostics and System Health Management Conference*, pages 1–5.

[Liu and Chen, 2019] Liu, J. and Chen, Z. (2019). Remaining useful life prediction of lithium-ion batteries based on health indicator and gaussian process regression model. *IEEE Access*, 7:39474–39484.

[Liu and Morse, 2011] Liu, J. and Morse, A. S. (2011). Accelerated linear iterations for distributed averaging. *Annual Reviews in Control*, 35(2):160–165.

[Liu et al., 2011] Liu, J., Mou, S., Morse, A. S., Anderson, B. D., and Yu, C. (2011). Deterministic gossiping. *Proceedings of the IEEE*, 99(9):1505–1524.

[Liu et al., 2019a] Liu, K., Hu, X., Wei, Z., Li, Y., and Jiang, Y. (2019a). Modified gaussian process regression models for cyclic capacity prediction of lithium-ion batteries. *IEEE Transactions on Transportation Electrification*, 5(4):1225–1236.

[Liu et al., 2019b] Liu, K., Li, Y., Hu, X., Lucu, M., and Widanage, W. D. (2019b). Gaussian process regression with automatic relevance determination kernel for calendar aging prediction of lithium-ion batteries. *IEEE Transactions on Industrial Informatics*, 16(6):3767–3777.

[Liu et al., 2020] Liu, K., Shang, Y., Ouyang, Q., and Widanage, W. D. (2020). A data-driven approach with uncertainty quantification for predicting future capacities and remaining useful life of lithium-ion battery. *IEEE Transactions on Industrial Electronics*, 68(4):3170–3180.

[Liu et al., 2015a] Liu, Q., Wang, Z., He, X., and Zhou, D.-H. (2015a). Event-based distributed filtering with stochastic measurement fading. *IEEE Transactions on Industrial Informatics*, 11(6):1643–1652.

[Liu et al., 2015b] Liu, Q., Wang, Z., He, X., and Zhou, D.-H. (2015b). Event-based recursive distributed filtering over wireless sensor networks. *IEEE Transactions on Automatic Control*, 60(9):2470–2475.

[Liu et al., 2023] Liu, S., Chang, P., Huang, Z., Chakraborty, N., Hong, K., Liang, W., McPherson, D. L., Geng, J., and Driggs-Campbell, K. (2023). Intention aware robot crowd navigation with attention-based interaction graph. In *IEEE International Conference on Robotics and Automation*, pages 12015–12021.

[Lo et al., 2023] Lo, Y. L., de Witt, C. S., Sokota, S., Foerster, J. N., and Whiteson, S. (2023). Cheap talk discovery and utilization in multi-agent reinforcement learning. In *The Eleventh International Conference on Learning Representations*.

[Long et al., 2020] Long, N. K., Sammut, K., Sgarioto, D., Garratt, M., and Abbass, H. A. (2020). A comprehensive review of shepherding as a bio-inspired swarm-robotics guidance approach. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 4(4):523–537.

[Long et al., 2018] Long, P., Fan, T., Liao, X., Liu, W., Zhang, H., and Pan, J. (2018). Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning. In *IEEE International Conference on Robotics and Automation*, pages 6252–6259.

[Long et al., 2017] Long, P., Liu, W., and Pan, J. (2017). Deep-learned collision avoidance policy for distributed multiagent navigation. *IEEE Robotics and Automation Letters*, 2(2):656–663.

[Long et al., 2019] Long, Q., Zhou, Z., Gupta, A., Fang, F., Wu, Y., and Wang, X. (2019). Evolutionary population curriculum for scaling multi-agent reinforcement learning. In *International Conference on Learning Representations*.

[Lowe et al., 2017] Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30.

[Lu et al., 2018] Lu, H., Freund, R. M., and Nesterov, Y. (2018). Relatively smooth convex optimization by first-order methods, and applications. *SIAM Journal on Optimization*, 28(1):333–354.

[Ma et al., 2016] Ma, K., He, N., Liserre, M., and Blaabjerg, F. (2016). Frequency-domain thermal modeling and characterization of power semiconductor devices. *IEEE Transactions on Power Electronics*, 31(10):7183–7193.

[Mahdavi et al., 1997] Mahdavi, J., Emaadi, A., Bellar, M., and Ehsani, M. (1997). Analysis of power electronic converters using the generalized state-space averaging approach. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 44(8):767–770.

[Mahdavi et al., 1997] Mahdavi, J., Emaadi, A., Bellar, M. D., and Ehsani, M. (1997). Analysis of power electronic converters using the generalized state-space averaging approach. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 44(8):767–770.

[Mahler, 2007] Mahler, R. P. (2007). *Statistical multisource-multitarget information fusion*, volume 685. Artech House Norwood, MA, USA.

[Mai et al., 2023] Mai, V. S., La, R. J., Zhang, T., and Battou, A. (2023). Distributed optimization with global constraints using noisy measurements. *IEEE Transactions on Automatic Control*.

[Malyuta et al., 2021] Malyuta, D., Yu, Y., Elango, P., and Açıkmeşe, B. (2021). Advances in trajectory optimization for space vehicle control. *Annual Reviews in Control*, 52:282–315.

[Marino et al., 2023] Marino, A., Pacchierotti, C., and Giordano, P. R. (2023). On the stability of gated graph neural networks. *arXiv preprint arXiv:2305.19235*.

[Marquardt, 1963] Marquardt, D. (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441.

[Martinez-Baselga et al., 2023] Martinez-Baselga, D., Riazuelo, L., and Montano, L. (2023). Improving robot navigation in crowded environments using intrinsic rewards. In *IEEE International Conference on Robotics and Automation*, pages 9428–9434.

[Martinez-Baselga et al., 2024] Martinez-Baselga, D., Sebastián, E., Montijano, E., Riazuelo, L., Sagüés, C., and Montano, L. (2024). AVOCADO: Adaptive optimal collision avoidance driven by opinion. *arXiv preprint arXiv:2407.00507*.

[Martinez-Piazuelo et al., 2022] Martinez-Piazuelo, J., Diaz-Garcia, G., Quijano, N., and Giraldo, L. F. (2022). Discrete-time distributed population dynamics for optimization and control. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(11):7112–7122.

[Matarić, 1997] Matarić, M. J. (1997). Reinforcement learning in the multi-robot domain. *Autonomous Robots*, 1(4):73–83.

[MathWorks®, 2020a] MathWorks®(2020a). *Simulink®Developing S-Functions*. MathWorks®.

[MathWorks®, 2020b] MathWorks®(2020b). *Simulink®User's Guide*. MathWorks®.

[Matignon et al., 2012] Matignon, L., Jeanpierre, L., and Mouaddib, A.-I. (2012). Coordinated multi-robot exploration under communication constraints using decentralized markov decision processes. In *AAAI Conference on Artificial Intelligence*, volume 26, pages 2017–2023.

[Matignon et al., 2007] Matignon, L., Laurent, G. J., and Le Fort-Piat, N. (2007). Hysteretic q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent teams. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 64–69.

[Mavrogiannis et al., 2022] Mavrogiannis, C., Balasubramanian, K., Poddar, S., Gandra, A., and Srinivasa, S. S. (2022). Winding through: Crowd navigation via topological invariance. *IEEE Robotics and Automation Letters*, 8(1):121–128.

[Mavrogiannis et al., 2023] Mavrogiannis, C., Baldini, F., Wang, A., Zhao, D., Trautman, P., Steinfeld, A., and Oh, J. (2023). Core challenges of social robot navigation: A survey. *ACM Transactions on Human-Robot Interaction*, 12(3):1–39.

[McLurkin and Yamins, 2005] McLurkin, J. and Yamins, D. (2005). Dynamic task assignment in robot swarms. In *Robotics: Science and Systems*, volume 8. Cambridge, USA.

[Meng et al., 2015] Meng, D., Fazel, M., and Mesbahi, M. (2015). Proximal alternating direction method of multipliers for distributed optimization on weighted graphs. In *IEEE Conference on Decision and Control*, pages 1396–1401.

[Meselhi et al., 2022] Meselhi, M., Sarker, R., Essam, D., and Elsayed, S. (2022). A decomposition approach for large-scale non-separable optimization problems. *Applied Soft Computing*, 115:108168.

[Meza et al., 2012] Meza, C., Biel, D., Jeltsema, D., and Scherpen, J. M. A. (2012). Lyapunov-based control scheme for single-phase grid-connected PV central inverters. *IEEE Transactions on Control Systems Technology*, 20(2):520–529.

[Mian et al., 2013] Mian, N. S., Fletcher, S., Longstaff, A. P., and Myers, A. (2013). Efficient estimation by FEA of machine tool distortion due to environmental temperature perturbations. *Precision Engineering*, 37(2):372–379.

[Montijano et al., 2014a] Montijano, E., Martínez, S., and Sagues, C. (2014a). Distributed robust consensus using ransac and dynamic opinions. *IEEE Transactions on Control Systems Technology*, 23(1):150–163.

[Montijano et al., 2012] Montijano, E., Montijano, J. I., and Sagues, C. (2012). Chebyshev polynomials in distributed consensus applications. *IEEE Transactions on Signal Processing*, 61(3):693–706.

[Montijano et al., 2017] Montijano, E., Montijano, J. I., and Sagues, C. (2017). Fast distributed algebraic connectivity estimation in large scale networks. *Journal of the Franklin Institute*, 354(13):5421–5442.

[Montijano et al., 2014b] Montijano, E., Montijano, J. I., Sagüés, C., and Martínez, S. (2014b). Robust discrete time dynamic average consensus. *Automatica*, 50(12):3131–3138.

[Montijano et al., 2013] Montijano, E., Priolo, A., Gasparri, A., and Sagues, C. (2013). Distributed entrapment for multi-robot systems with uncertainties. In *IEEE Conference on Decision and Control*, pages 403–408.

[Moradian and Kia, 2018] Moradian, H. and Kia, S. S. (2018). On robustness analysis of a dynamic average consensus algorithm to communication delay. *IEEE Transactions on Control of Network Systems*, 6(2):633–641.

[Mordatch and Abbeel, 2018] Mordatch, I. and Abbeel, P. (2018). Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

[Morgan et al., 2014] Morgan, D., Chung, S.-J., and Hadaegh, F. Y. (2014). Model predictive control of swarms of spacecraft using sequential convex programming. *Journal of Guidance, Control, and Dynamics*, 37(6):1725–1740.

[Munikoti et al., 2023] Munikoti, S., Agarwal, D., Das, L., Halappanavar, M., and Natarajan, B. (2023). Challenges and opportunities in deep reinforcement learning with graph neural networks: A comprehensive review of algorithms and applications. *IEEE Transactions on Neural Networks and Learning Systems*.

[Nedic, 2020] Nedic, A. (2020). Distributed gradient methods for convex machine learning problems in networks: Distributed optimization. *IEEE Signal Processing Magazine*, 37(3):92–101.

[Nedić and Liu, 2018] Nedić, A. and Liu, J. (2018). Distributed optimization for control. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:77–103.

[Nedić and Olshevsky, 2014] Nedić, A. and Olshevsky, A. (2014). Distributed optimization over time-varying directed graphs. *IEEE Transactions on Automatic Control*, 60(3):601–615.

[Nedić et al., 2018] Nedić, A., Olshevsky, A., and Rabbat, M. G. (2018). Network topology and communication-computation tradeoffs in decentralized optimization. *Proceedings of the IEEE*, 106(5):953–976.

[Nedic et al., 2017] Nedic, A., Olshevsky, A., and Shi, W. (2017). Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4):2597–2633.

[Nesterov and Nemirovskii, 1994] Nesterov, Y. and Nemirovskii, A. (1994). *Interior-point polynomial algorithms in convex programming*. SIAM.

[Ng et al., 2000] Ng, A. Y., Russell, S., et al. (2000). Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning*, volume 1, page 2.

[Nghiem et al., 2023] Nghiem, T. X., Drgoňa, J., Jones, C., Nagy, Z., Schwan, R., Dey, B., Chakrabarty, A., Di Cairano, S., Paulson, J. A., Carron, A., et al. (2023). Physics-informed machine learning for modeling and control of dynamical systems. *arXiv preprint arXiv:2306.13867*.

[Nguyen et al., 2023] Nguyen, D. T. A., Nguyen, D. T., and Nedić, A. (2023). Accelerated AB/push-pull methods for distributed optimization over time-varying directed networks. *IEEE Transactions on Control of Network Systems*.

[Niehsen, 2002] Niehsen, W. (2002). Information fusion based on fast covariance intersection filtering. In *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002.*, volume 2, pages 901–904 vol.2.

[Nolfi, 2002] Nolfi, S. (2002). Power and the limits of reactive agents. *Neurocomputing*, 42(1-4):119–145.

[Nosrati et al., 2012] Nosrati, S., Shafiee, M., and Menhaj, M. B. (2012). Dynamic average consensus via nonlinear protocols. *Automatica*, 48(9):2262–2270.

[Notarnicola et al., 2019] Notarnicola, I., Franceschelli, M., and Notarstefano, G. (2019). A duality-based approach for distributed min–max optimization. *IEEE Transactions on Automatic Control*, 64(6):2559–2566.

[Notarnicola and Notarstefano, 2019] Notarnicola, I. and Notarstefano, G. (2019). Constraint-coupled distributed optimization: a relaxation and duality approach. *IEEE Transactions on Control of Network Systems*, 7(1):483–492.

[Notarstefano et al., 2019] Notarstefano, G., Notarnicola, I., Camisa, A., et al. (2019). Distributed optimization for smart cyber-physical networks. *Foundations and Trends® in Systems and Control*, 7(3):253–383.

[Oetinger and Sawodny, 2015] Oetinger, D. and Sawodny, O. (2015). Using model order reduction for disturbance feed forward control based on transient thermal finite element models. In *IEEE Conference on Control Applications*, pages 1486–1491.

[Olfati-Saber, 2005] Olfati-Saber, R. (2005). Distributed Kalman filter with embedded consensus filters. In *IEEE Conference on Decision and Control*, pages 8179–8184.

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

[Olfati-Saber, 2007a] Olfati-Saber, R. (2007a). Distributed Kalman filtering for sensor networks. In *IEEE Conference on Decision and Control*, pages 5492–5498.

[Olfati-Saber, 2007b] Olfati-Saber, R. (2007b). Distributed Kalman filtering for sensor networks. In *IEEE Conference on Decision and Control*, pages 5492–5498.

[Olfati-Saber, 2009] Olfati-Saber, R. (2009). Kalman-consensus filter: Optimality, stability, and performance. In *IEEE Conference on Decision and Control*, pages 7036–7042.

[Olfati-Saber et al., 2007] Olfati-Saber, R., Fax, J. A., and Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233.

[Olfati-Saber and Shamma, 2005] Olfati-Saber, R. and Shamma, J. S. (2005). Consensus filters for sensor networks and distributed sensor fusion. In *IEEE Conference on Decision and Control*, pages 6698–6703. IEEE.

[Oruganti and Lee, 1985a] Oruganti, R. and Lee, F. C. (1985a). Resonant power processors, part i—state plane analysis. *IEEE Transactions on industry applications*, 1(6):1453–1460.

[Oruganti and Lee, 1985b] Oruganti, R. and Lee, F. C. (1985b). Resonant power processors, part ii-methods of control. *IEEE Transactions on Industry Applications*, 1(6):1461–1471.

[Oruganti et al., 1988] Oruganti, R., Yang, J. J., and Lee, F. C. (1988). Implementation of optimal trajectory control of series resonant converter. *IEEE Transactions on Power Electronics*, 3(3):318–327.

[Ourari et al., 2022] Ourari, R., Cui, K., Elshamanhory, A., and Koeppl, H. (2022). Nearest-neighbor-based collision avoidance for quadrotors via reinforcement learning. In *IEEE International Conference on Robotics and Automation*, pages 293–300.

[Pajovic et al., 2017] Pajovic, M., Sahinoglu, Z., Wang, Y., Orlik, P. V., and Wada, T. (2017). Online data-driven battery voltage prediction. In *IEEE International Conference on Industrial Informatics*, pages 827–834.

[Papamanolis et al., 2021] Papamanolis, P., Guillod, T., Krismer, F., and Kolar, J. W. (2021). Transient calorimetric measurement of ferrite core losses up to 50 MHz. *IEEE Transactions on Power Electronics*, 36(3):2548–2563.

[Parikh and Boyd, 2014] Parikh, N. and Boyd, S. (2014). Block splitting for distributed optimization. *Mathematical Programming Computation*, 6(1):77–102.

[Parise and Ozdaglar, 2019] Parise, F. and Ozdaglar, A. (2019). A variational inequality framework for network games: Existence, uniqueness, convergence and sensitivity analysis. *Games and Economic Behavior*, 114:47–82.

[Parnika et al., 2021] Parnika, P., Diddigi, R. B., Danda, S. K. R., and Bhatnagar, S. (2021). Attention actor-critic algorithm for multi-agent constrained co-operative reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*.

[Patwardhan et al., 2022] Patwardhan, A., Murai, R., and Davison, A. J. (2022). Distributing collaborative multi-robot planning with gaussian belief propagation. *IEEE Robotics and Automation Letters*, 8(2):552–559.

[Pavlović et al., 2012] Pavlović, Z., Oliver, J. A., Alou, P., García, O., and Cobos, J. A. (2012). Bidirectional dual active bridge series resonant converter with pulse modulation. In *IEEE Applied Power Electronics Conference and Exposition*, pages 503–508.

[Peng et al., 2021] Peng, B., Rashid, T., Schroeder de Witt, C., Kamienny, P.-A., Torr, P., Böhmer, W., and Whiteson, S. (2021). FACMAC: Factored multi-agent centralised policy gradients. *Advances in Neural Information Processing Systems*, 34:12208–12221.

[Peng et al., 2016] Peng, X., Zhang, S., and Lei, X. (2016). Multi-target trapping in constrained environments using gene regulatory network-based pattern formation. In *International Journal of Advanced Robotic Systems*, volume 13, pages 5859–5864.

[Petti and Fraichard, 2005] Petti, S. and Fraichard, T. (2005). Safe motion planning in dynamic environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2210–2215.

[Pierson and Schwager, 2017] Pierson, A. and Schwager, M. (2017). Controlling noncooperative herds with robotic herders. *IEEE Transactions on Robotics*, 34(2):517–525.

[Pierson and Schwager, 2018] Pierson, A. and Schwager, M. (2018). Controlling noncooperative herds with robotic herders. *IEEE Transactions on Robotics*, 34(2):517–525.

[Poddar et al., 2023] Poddar, S., Mavrogiannis, C., and Srinivasa, S. S. (2023). From crowd motion prediction to robot navigation in crowds. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 6765–6772.

[Pokle et al., 2019] Pokle, A., Martín-Martín, R., Goebel, P., Chow, V., Ewald, H. M., Yang, J., Wang, Z., Sadeghian, A., Sadigh, D., Savarese, S., et al. (2019). Deep local trajectory replanning and control for robot navigation. In *2019 International Conference on Robotics and Automation*, pages 5815–5822.

[Qin et al., 2023] Qin, J., Qin, J., Qiu, J., Liu, Q., Li, M., and Ma, Q. (2023). SRL-ORCA: A socially aware multi-agent mapless navigation algorithm in complex dynamic scenes. *IEEE Robotics and Automation Letters*, 9(1):143–150.

[Qu and Li, 2017] Qu, G. and Li, N. (2017). Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 5(3):1245–1260.

[Qu and Li, 2019] Qu, G. and Li, N. (2019). Accelerated distributed nesterov gradient descent. *IEEE Transactions on Automatic Control*, 65(6):2566–2581.

[Qu et al., 2020] Qu, G., Wierman, A., and Li, N. (2020). Scalable reinforcement learning of localized policies for multi-agent networked systems. In *Learning for Dynamics and Control*, pages 256–266. PMLR.

[Qu et al., 2022] Qu, G., Wierman, A., and Li, N. (2022). Scalable reinforcement learning for multiagent networked systems. *Operations Research*, 70(6):3601–3628.

[Queralta et al., 2020] Queralta, J. P., Taipalmaa, J., Pullinen, B. C., Sarker, V. K., Gia, T. N., Tenhunen, H., Gabbouj, M., Raitoharju, J., and Westerlund, T. (2020). Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision. *IEEE Access*, 8:191617–191643.

[Quinonero-Candela and Rasmussen, 2005] Quinonero-Candela, J. and Rasmussen, C. E. (2005). A unifying view of sparse approximate gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959.

[Ramachandran et al., 2017] Ramachandran, P., Zoph, B., and Le, Q. V. (2017). Searching for activation functions. *arXiv preprint arXiv:1710.05941*.

[Rasmussen and Nickisch, 2010] Rasmussen, C. E. and Nickisch, H. (2010). Gaussian processes for machine learning (GPML) toolbox. *The Journal of Machine Learning Research*, 11:3011–3015.

[Reinhardt et al., 2015] Reinhardt, M., Noack, B., Arambel, P. O., and Hanebeck, U. D. (2015). Minimum covariance bounds for the fusion under unknown correlations. *IEEE Signal Processing Letters*, 22(9):1210–1214.

[Roche et al., 2025] Roche, J., Sebastián, E., and Montijano, E. (2025). Curriculum imitation learning for distributed multi-robot policies. In *IEEE International Conference on Robotics and Automation*, pages 1–6. under review.

[Rockafellar, 1970] Rockafellar, R. T. (1970). *Convex analysis*. Princeton Mathematical Series. Princeton University Press, Princeton, N. J.

[Rodríguez and Cortés, 2012] Rodríguez, J. and Cortés, P. (2012). *Predictive control of power converters and electrical drives*, volume 40. John Wiley & Sons.

[Rogers et al., 2023] Rogers, D. J., Bruford, J., Ristic-Smith, A., Ali, K., Palmer, P., and Shelton, E. (2023). A comparison of the hard-switching performance of 650 v power transistors with calorimetric verification. *IEEE Open Journal of Power Electronics*, 4:764–775.

[Romao et al., 2021] Romao, L., Margellos, K., Notarstefano, G., and Papachristodoulou, A. (2021). Subgradient averaging for multi-agent optimisation with different constraint sets. *Automatica*, 131:109738.

[Rudin, 1976] Rudin, W. (1976). *Principles of Mathematical Analysis*. McGraw-Hill, Inc., 3rd edition.

[Rufli et al., 2013] Rufli, M., Alonso-Mora, J., and Siegwart, R. (2013). Reciprocal collision avoidance with motion continuity constraints. *IEEE Transactions on Robotics*, 29(4):899–912.

[Sadamoto et al., 2020] Sadamoto, T., Chakrabortty, A., and Imura, J.-i. (2020). Fast online reinforcement learning control using state-space dimensionality reduction. *IEEE Transactions on Control of Network Systems*, 8(1):342–353.

[Saggini et al., 2007] Saggini, S., Stefanutti, W., Tedeschi, E., and Mattavelli, P. (2007). Digital deadbeat control tuning for DC-DC converters using error correlation. *IEEE Transactions on Power Electronics*, 22(4):1566–1570.

[Sahinoglu et al., 2017] Sahinoglu, G. O., Pajovic, M., Sahinoglu, Z., Wang, Y., Orlik, P. V., and Wada, T. (2017). Battery state-of-charge estimation based on regular/recurrent gaussian process regression. *IEEE Transactions on Industrial Electronics*, 65(5):4311–4321.

[Sanders et al., 1991] Sanders, S. R., Noworolski, J. M., Liu, X. Z., and Verghese, G. C. (1991). Generalized averaging method for power conversion circuits. *IEEE Transactions on Power Electronics*, 6(2):251–259.

[Sanders et al., 1991] Sanders, S. R., Noworolski, J. M., Liu, X. Z., and Verghese, G. C. (1991). Generalized averaging method for power conversion circuits. *IEEE Transactions on power Electronics*, 6(2):251–259.

[Sanz-Alcaine et al., 2023] Sanz-Alcaine, J. M., Jose Perez-Cebolla, F., Bernal-Ruiz, C., Arruti, A., Aizpuru, I., and Sanchez, J. (2023). Loss measurement of low RDS devices through thermal modelling - the advantage of not turning it fully on. In *European Conference on Power Electronics and Applications*, pages 1–7.

[Sanz-Alcaine et al., 2024] Sanz-Alcaine, J. M., Sebastián, E., Perez-Cebolla, F. J., Arruti, A., Bernal-Ruiz, C., and Aizpuru, I. (2024). Estimation of semiconductor power losses through automatic thermal modeling. *IEEE Transactions on Power Electronics*.

[Sanz-Alcaine et al., 2021] Sanz-Alcaine, J. M., Sebastián, E., Sanz-Gorrachategui, I., Bernal-Ruiz, C., Bono-Nuez, A., Pajovic, M., and Orlik, P. V. (2021). Online voltage prediction using gaussian process regression for fault-tolerant photovoltaic standalone applications. *Neural Computing and Applications*, 33:16577–16590.

[Saravanos et al., 2023] Saravanos, A. D., Li, Y., and Theodorou, E. A. (2023). Distributed hierarchical distribution control for very-large-scale clustered multi-agent systems. In *Robotics: Science and Systems*.

[Schiestl et al., 2020] Schiestl, M., Losch, A., Incurvati, M., and Starz, R. (2020). Accurate losses multipoint non adiabatic calorimetric measurement technique for WBG power converters. *PCIM Europe Conference Proceedings*, pages 1330–1337.

[Schulman et al., 2015] Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR.

[Scutari and Sun, 2019] Scutari, G. and Sun, Y. (2019). Distributed nonconvex constrained optimization over time-varying digraphs. *Mathematical Programming*, 176:497–544.

[Sebastián et al., 2025] Sebastián, E., Atanasov, N., Montijano, E., and Sagüés, C. (2025). Learning linear distributed quadratic regulators for nonlinear multi-agent games. In *International Conference on Machine Learning*, pages 1–12. under review.

[Sebastián et al., 2023a] Sebastián, E., Duong, T., Atanasov, N., Montijano, E., and Sagüés, C. (2023a). Learning to identify graphs from node trajectories in multi-robot networks. In *IEEE International Symposium on Multi-Robot and Multi-Agent Systems*, pages 142–148.

[Sebastián et al., 2023b] Sebastián, E., Duong, T., Atanasov, N., Montijano, E., and Sagüés, C. (2023b). LEMURS: Learning distributed multi-robot interactions. In *IEEE International Conference on Robotics and Automation*, pages 7713–7719.

[Sebastian et al., 2023] Sebastian, E., Duong, T., Atanasov, N., Montijano, E., and Sagues, C. (2023). Physics-informed multi-agent reinforcement learning for distributed multi-robot problems. *arXiv preprint arXiv:2401.00212*.

[Sebastián et al., 2024] Sebastián, E., Franceschelli, M., Gasparri, A., Montijano, E., and Sagüés, C. (2024). Accelerated alternating direction method of multipliers gradient tracking for distributed optimization. *IEEE Control Systems Letters*.

[Sebastián and Montijano, 2021] Sebastián, E. and Montijano, E. (2021). Multi-robot implicit control of herds. In *IEEE International Conference on Robotics and Automation*, pages 1601–1607.

[Sebastián et al., 2021a] Sebastián, E., Montijano, E., Oyarbide, E., Bernal, C., and Gálvez, R. (2021a). Nonlinear implementable control of a dual active bridge series resonant converter. *IEEE Transactions on Industrial Electronics*, 69(5):5111–5121.

[Sebastián et al., 2021b] Sebastián, E., Montijano, E., and Sagüés, C. (2021b). All-in-one: Certifiable optimal distributed kalman filter under unknown correlations. In *IEEE Conference on Decision and Control*, pages 6578–6583. IEEE.

[Sebastián et al., 2021c] Sebastián, E., Montijano, E., and Sagüés, C. (2021c). All-in-one: Certifiable optimal distributed kalman filter under unknown correlations. In *IEEE Conference on Decision and Control*, pages 6578–6583.

[Sebastián et al., 2022a] Sebastián, E., Montijano, E., and Sagüés, C. (2022a). Adaptive multirobot implicit control of heterogeneous herds. *IEEE Transactions on Robotics*, 38(6):3622–3635.

[Sebastián et al., 2022b] Sebastián, E., Montijano, E., and Sagüés, C. (2022b). Multi-robot implicit control of massive herds. In *Iberian Robotics conference*, pages 448–459. Springer.

[Sebastián et al., 2023a] Sebastián, E., Montijano, E., and Sagüés, C. (2023a). ECO-DKF: Event-triggered and certifiable optimal distributed Kalman filter under unknown correlations. *IEEE Transactions on Automatic Control*.

[Sebastián et al., 2023b] Sebastián, E., Montijano, E., and Sagüés, C. (2023b). ECO-DKF: Event-triggered and certifiable optimal distributed kalman filter under unknown correlations. *arXiv preprint arXiv:2311.02406*.

[Sebastián et al., 2023c] Sebastián, E., Montijano, E., Sagüés, C., Franceschelli, M., and Gasparri, A. (2023c). Accelerated multi-stage discrete time dynamic average consensus. *IEEE Control Systems Letters*, 7:2731–2736.

[Sechilariu et al., 2012] Sechilariu, M., Wang, B., and Locment, F. (2012). Building integrated photovoltaic system with energy storage and smart grid communication. *IEEE Transactions on Industrial Electronics*, 60(4):1607–1618.

[Seltzer et al., 2012] Seltzer, D., Bloomquist, D., Zane, R., and Maksimovic, D. (2012). Gain-scheduled control of multi angle phase shift modulated dual active bridge series resonant DC/DC converters. In *IEEE Workshop on Control and Modeling for Power Electronics*, pages 1–7.

[Seltzer et al., 2011] Seltzer, D., Corradini, L., Bloomquist, D., Zane, R., and Maksimović, D. (2011). Small signal phasor modeling of dual active bridge series resonant DC/DC converters with multi-angle phase shift modulation. In *IEEE Energy Conversion Congress and Exposition*.

[Semnani et al., 2020] Semnani, S. H., Liu, H., Everett, M., De Ruiter, A., and How, J. P. (2020). Multi-agent motion planning for dense and dynamic environments via deep reinforcement learning. *IEEE Robotics and Automation Letters*, 5(2):3221–3226.

[Serra-Gómez et al., 2023] Serra-Gómez, Á., Zhu, H., Brito, B., Böhmer, W., and Alonso-Mora, J. (2023). Learning scalable and efficient communication policies for multi-robot collision avoidance. *Autonomous Robots*, pages 1–23.

[Shah et al., 2020] Shah, K., Ballard, G., Schmidt, A., and Schwager, M. (2020). Multidrone aerial surveys of penguin colonies in antarctica. *Science Robotics*, 5(47):1–12.

[Shahjalal et al., 2020] Shahjalal, M., Ahmed, M. R., Lu, H., Bailey, C., and Forsyth, A. J. (2020). An analysis of the thermal interaction between components in power converter applications. *IEEE Transactions on Power Electronics*, 35(9):9082–9094.

[Shi et al., 2014] Shi, D., Chen, T., and Shi, L. (2014). On set-valued Kalman filtering and its application to event-based state estimation. *IEEE Transactions on Automatic Control*, 60(5):1275–1290.

[Shi et al., 2020] Shi, G., Hönig, W., Yue, Y., and Chung, S.-J. (2020). Neural-swarm: Decentralized close-proximity multirotor control using learned interactions. In *IEEE International Conference on Robotics and Automation*, pages 3241–3247. IEEE.

[Shi et al., 2015] Shi, W., Ling, Q., Wu, G., and Yin, W. (2015). Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966.

[Shorinwa et al., 2024] Shorinwa, O., Halsted, T., Yu, J., and Schwager, M. (2024). Distributed Optimization Methods for Multi-robot Systems: Part 2-A Survey. *IEEE Robotics & Automation Magazine*.

[Sinopoli et al., 2004] Sinopoli, B., Schenato, L., Franceschetti, M., Poolla, K., Jordan, M. I., and Sastry, S. S. (2004). Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, 49(9):1453–1464.

[Slotine et al., 1991] Slotine, J.-J. E., Li, W., et al. (1991). *Applied nonlinear control*, volume 199. Prentice Hall Englewood Cliffs, NJ.

[Song et al., 2021] Song, H., Varava, A., Kravchenko, O., Kragic, D., Wang, M. Y., Pokorny, F. T., and Hang, K. (2021). Herding by caging: a formation-based motion planning framework for guiding mobile agents. *Autonomous Robots*, pages 1–19.

[Soto et al., 2020] Soto, F., Wang, J., Ahmed, R., and Demirci, U. (2020). Medical micro/nanorobots in precision medicine. *Advanced science*, 7(21):2002203.

[Stilwell et al., 2005] Stilwell, D. J., Bishop, B. E., and Sylvester, C. A. (2005). Redundant manipulator techniques for partially decentralized path planning and control of a platoon of autonomous vehicles. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(4):842–848.

[Strömbom et al., 2014a] Strömbom, D., Mann, R. P., Wilson, A. M., Hailes, S., Morton, A. J., Sumpter, D. J., and King, A. J. (2014a). Solving the shepherding problem: heuristics for herding autonomous, interacting agents. *Journal of the royal society interface*, 11(100):20140719.

[Strömbom et al., 2014b] Strömbom, D., Mann, R. P., Wilson, A. M., Hailes, S., Morton, A. J., Sumpter, D. J., and King, A. J. (2014b). Solving the shepherding problem: heuristics for herding autonomous, interacting agents. *Journal of the royal society interface*, 11(100).

[Suh et al., 2007] Suh, Y. S., Nguyen, V. H., and Ro, Y. S. (2007). Modified Kalman filter for networked monitoring systems employing a send-on-delta method. *Automatica*, 43(2):332–338.

[Sundhar Ram et al., 2010] Sundhar Ram, S., Nedić, A., and Veeravalli, V. V. (2010). Distributed stochastic subgradient projection algorithms for convex optimization. *Journal of optimization theory and applications*, 147:516–545.

[Sutton, 2018] Sutton, R. S. (2018). Reinforcement learning: An introduction. *A Bradford Book*.

[Tai et al., 2018] Tai, L., Zhang, J., Liu, M., and Burgard, W. (2018). Socially compliant navigation through raw depth inputs with generative adversarial imitation learning. In *IEEE International Conference on Robotics and Automation*, pages 1111–1117.

[Talebi and Werner, 2019] Talebi, S. P. and Werner, S. (2019). Distributed kalman filtering and control through embedded average consensus information fusion. *IEEE Transactions on Automatic Control*, 64(10):4396–4403.

[Tavakoli and Pantic, 2018] Tavakoli, R. and Pantic, Z. (2018). Analysis, design, and demonstration of a 25-kW dynamic wireless charging system for roadway electric vehicles. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 6(3):1378–1393.

[Technologies, 2010] Technologies, A. (2010). Agilent InfiniiVision 7000B Series Oscilloscopes user's guide.

[Technologies, 2018] Technologies, I. (2018). EiceDRIVER™ 2EDF7275K dual-channel isolated gate driver IC, Infineon Technologies. Datasheet, Infineon Technologies.

[Technologies, 2021] Technologies, I. (2021). ISC030N10NM6 - 100V OptiMOS™ 6 Power-Transistor. Datasheet, Infineon Technologies.

[Todorov et al., 2012] Todorov, E., Erez, T., and Tassa, Y. (2012). MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033.

[Tolstaya et al., 2020] Tolstaya, E., Gama, F., Paulos, J., Pappas, G., Kumar, V., and Ribeiro, A. (2020). Learning decentralized controllers for robot swarms with graph neural networks. In *Conference on Robot Learning*, pages 671–682.

[Tolstaya et al., 2021] Tolstaya, E., Paulos, J., Kumar, V., and Ribeiro, A. (2021). Multi-robot coverage and exploration using spatial graph neural networks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 8944–8950.

[Trimpe and Campi, 2015] Trimpe, S. and Campi, M. C. (2015). On the choice of the event trigger in event-based estimation. In *IEEE International Conference on Event-based Control, Communication, and Signal Processing*, pages 1–8.

[Trimpe and D'Andrea, 2014] Trimpe, S. and D'Andrea, R. (2014). Event-based state estimation with variance-based triggering. *IEEE Transactions on Automatic Control*, 59(12):3266–3281.

[Ubl and Hale, 2019] Ubl, M. and Hale, M. T. (2019). Totally asynchronous distributed quadratic programming with independent stepsizes and regularizations. In *IEEE Conference on Decision and Control*, pages 7423–7428.

[Van Den Berg et al., 2011] Van Den Berg, J., Guy, S. J., Lin, M., and Manocha, D. (2011). Reciprocal n-body collision avoidance. In *Robotics Research: The 14th International Symposium ISRR*, pages 3–19. Springer.

[Van den Berg et al., 2008] Van den Berg, J., Lin, M., and Manocha, D. (2008). Reciprocal velocity obstacles for real-time multi-agent navigation. In *IEEE International Conference on Robotics and Automation*, pages 1928–1935.

[Van Der Broeck et al., 2018] Van Der Broeck, C. H., Ruppert, L. A., Hinz, A., Conrad, M., and De Doncker, R. W. (2018). Spatial electro-thermal modeling and simulation of power electronic modules. *IEEE Transactions on Industry Applications*, 54(1):404–415.

[Van Der Schaft et al., 2014] Van Der Schaft, A., Jeltsema, D., et al. (2014). Port-hamiltonian systems theory: An introductory overview. *Foundations and Trends® in Systems and Control*, 1(2-3):173–378.

[van der Schaft, 2004] van der Schaft, A. J. (2004). Port-Hamiltonian systems: network modeling and control of nonlinear physical systems. In *Advanced dynamics and control of structures and machines*, pages 127–167. Springer.

[Van Scoy and Lessard, 2022] Van Scoy, B. and Lessard, L. (2022). A universal decomposition for distributed optimization algorithms. *IEEE Control Systems Letters*, 6:3044–3049.

[Vandenberghe and Boyd, 1996] Vandenberghe, L. and Boyd, S. (1996). Semidefinite programming. *SIAM review*, 38(1):49–95.

[Varagnolo et al., 2015] Varagnolo, D., Zanella, F., Cenedese, A., Pillonetto, G., and Schenato, L. (2015). Newton-raphson consensus for distributed convex optimization. *IEEE Transactions on Automatic Control*, 61(4):994–1009.

[Vasca and Iannelli, 2012] Vasca, F. and Iannelli, L. (2012). *Dynamics and Control of Switched Electronic Systems Advanced Perspectives for Modeling, Simulation and Control of Power Converters.* Springer, London, 1st edition.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

[Vazquez et al., 2010] Vazquez, S., Lukic, S. M., Galvan, E., Franquelo, L. G., and Carrasco, J. M. (2010). Energy storage systems for transport and grid applications. *IEEE Transactions on Industrial Electronics*, 57(12):3881–3895.

[Vesanto et al., 1999] Vesanto, J., Himberg, J., Alhoniemi, E., Parhankangas, J., et al. (1999). Self-organizing map in Matlab: the SOM toolbox. In *Matlab DSP conference*, volume 99, pages 16–17. Finland.

[Vesentini et al., 2024] Vesentini, F., Muradore, R., and Fiorini, P. (2024). A survey on velocity obstacle paradigm. *Robotics and Autonomous Systems*, page 104645.

[VISHAY, 2021] VISHAY (2021). IHTH1500TZEB101M5A - High Current Through-Hole Inductor, High Temperature Series. Datasheet, VISHAY.

[Vorperian and Cuk, 1982] Vorperian, V. and Cuk, S. (1982). A complete dc analysis of the series resonant converter. In *1982 IEEE Power Electronics Specialists conference*, pages 85–100. IEEE.

[Wagstaff et al., 2001] Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S., et al. (2001). Constrained k-means clustering with background knowledge. In *International Conference on Machine Learning*, volume 1, pages 577–584.

[Walke et al., 2023] Walke, H. R., Black, K., Zhao, T. Z., Vuong, Q., Zheng, C., Hansen-Estruch, P., He, A. W., Myers, V., Kim, M. J., Du, M., et al. (2023). Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pages 1723–1736. PMLR.

[Wang et al., 2022] Wang, B., Xie, J., and Atanasov, N. (2022). DARL1N: Distributed multi-agent reinforcement learning with one-hop neighbors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

[Wang et al., 2017] Wang, L., Ames, A. D., and Egerstedt, M. (2017). Safety barrier certificates for collisions-free multirobot systems. *IEEE Transactions on Robotics*, 33(3):661–674.

[Wang et al., 2021] Wang, X., Chen, Y., and Zhu, W. (2021). A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4555–4576.

[Wattenberg et al., 2023] Wattenberg, M., Lorenz, O. G., and Sanchez, J. (2023). A multi-kilowatt low-profile gan inverter for light electric vehicles and high-power tools. In *IEEE Applied Power Electronics Conference and Exposition*, pages 1443–1450.

[Wei and Ozdaglar, 2013] Wei, E. and Ozdaglar, A. (2013). On the O(1/k) convergence of asynchronous distributed alternating direction method of multipliers. In *IEEE Global Conference on Signal and Information Processing*, pages 551–554.

[Weimer et al., 2022] Weimer, J., Koch, D., Nitzsche, M., Haarer, J., Roth-Stielow, J., and Kallfass, I. (2022). Miniaturization and thermal design of a 170 W AC/DC battery charger utilizing GaN power devices. *IEEE Open Journal of Power Electronics*, 3:13–25.

[Wendell and Hurter Jr, 1976] Wendell, R. E. and Hurter Jr, A. P. (1976). Minimization of a non-separable objective function subject to disjoint constraints. *Operations Research*, 24(4):643–657.

[Westenbroek et al., 2020] Westenbroek, T., Fridovich-Keil, D., Mazumdar, E., Arora, S., Prabhu, V., Sastry, S. S., and Tomlin, C. J. (2020). Feedback linearization for uncertain systems via reinforcement learning. In *IEEE International Conference on Robotics and Automation*, pages 1364–1371.

[Wilson et al., 2020] Wilson, S., Glotfelter, P., Wang, L., Mayya, S., Notomista, G., Mote, M., and Egerstedt, M. (2020). The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems. *IEEE Control Systems Magazine*, 40(1):26–44.

[Wipf and Nagarajan, 2007] Wipf, D. and Nagarajan, S. (2007). A new view of automatic relevance determination. *Advances in neural information processing systems*, 20.

[Wu et al., 2012] Wu, J., Jia, Q.-S., Johansson, K. H., and Shi, L. (2012). Event-based sensor data scheduling: Trade-off between communication rate and estimation quality. *IEEE Transactions on Automatic Control*, 58(4):1041–1046.

[Xiao et al., 2007] Xiao, C., Chen, G., and Odendaal, W. G. H. (2007). Overview of power loss measurement techniques in power electronics systems. *IEEE Transactions on Industry Applications*, 43(3):657–664.

[Xie et al., 2021] Xie, Z., Xin, P., and Dames, P. (2021). Towards safe navigation through crowded dynamic environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4934–4940.

[Xin and Khan, 2019] Xin, R. and Khan, U. A. (2019). Distributed heavy-ball: A generalization and acceleration of first-order methods with gradient tracking. *IEEE Transactions on Automatic Control*, 65(6):2627–2633.

[Yang and Matni, 2021] Yang, F. and Matni, N. (2021). Communication topology co-design in graph recurrent neural network based distributed control. In *IEEE Conference on Decision and Control*, pages 3619–3626.

[Yang and Carlone, 2022] Yang, H. and Carlone, L. (2022). Certifiably optimal outlier-robust geometric perception: Semidefinite relaxations and scalable global optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):2816–2834.

[Yang et al., 2020] Yang, H., Shi, J., and Carlone, L. (2020). Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 37(2):314–333.

[Yang et al., 2023] Yang, Q., Nabih, A., Zhang, R., Li, Q., and Zhang, Y. (2023). A converter based switching loss measurement method for WBG device. In *IEEE Applied Power Electronics Conference and Exposition*, pages 8–13.

[Yang et al., 2019] Yang, T., Yi, X., Wu, J., Yuan, Y., Wu, D., Meng, Z., Hong, Y., Wang, H., Lin, Z., and Johansson, K. H. (2019). A survey of distributed optimization. *Annual Reviews in Control*, 47:278–305.

[Yang et al., 2018] Yang, Y., Luo, R., Li, M., Zhou, M., Zhang, W., and Wang, J. (2018). Mean field multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 5571–5580. PMLR.

[You and Xie, 2012] You, K. and Xie, L. (2012). Kalman filtering with scheduled measurements. *IEEE Transactions on Signal Processing*, 61(6):1520–1530.

[Yu et al., 2022] Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y., Bayen, A., and Wu, Y. (2022). The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35:24611–24624.

[Zambaldi et al., 2018] Zambaldi, V., Raposo, D., Santoro, A., Bapst, V., Li, Y., Babuschkin, I., Tuyls, K., Reichert, D., Lillicrap, T., Lockhart, E., et al. (2018). Deep reinforcement learning with relational inductive biases. In *International conference on learning representations*.

[Zhan et al., 2025] Zhan, W., Dong, C., Sebastián, E., and Atanasov, N. (2025). Image-to-automata learning for robot task planning. In *International Conference on Machine Learning*, pages 1–12. under review.

[Zhang and Jia, 2017] Zhang, C. and Jia, Y. (2017). Distributed Kalman consensus filter with event-triggered communication: Formulation and stability analysis. *Journal of the Franklin Institute*, 354(13):5486–5502.

[Zhang et al., 2015] Zhang, C., Wang, J., Li, S., Wu, B., and Qian, C. (2015). Robust control for PWM-based DC-DC buck power converters with uncertainty via sampled-data output feedback. *IEEE Transactions on Power Electronics*, 30(1):504–515.

[Zhang et al., 2023] Zhang, S., Garg, K., and Fan, C. (2023). Neural graph control barrier functions guided distributed collision-avoidance multi-agent control. In *Conference on Robot Learning*, pages 2373–2392. PMLR.

[Zhang et al., 2015] Zhang, Y., Wang, C., Li, N., and Chambers, J. (2015). Diffusion Kalman filter based on local estimate exchanges. In *IEEE International Conference on Digital Signal Processing*, pages 828–832.

[Zhao et al., 2017] Zhao, B., Wang, G., and Hurley, W. G. (2017). Analysis and performance of LCLC resonant converters for high-voltage high-frequency applications. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 5(3):1272–1286.

[Zhao et al., 2023] Zhao, Z., Zhao, Z., and Zhang, P. (2023). A new method for identifying industrial clustering using the standard deviational ellipse. *Scientific Reports*, 13(1):578.

[Zhou et al., 2019] Zhou, S., Phielipp, M. J., Sefair, J. A., Walker, S. I., and Amor, H. B. (2019). Clone swarms: Learning to predict and control multi-robot systems by imitation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4092–4099.

[Zhu et al., 2015] Zhu, D., Lv, R., Cao, X., and Yang, S. X. (2015). Multi-AUV hunting algorithm based on bio-inspired neural network in unknown environments. *International Journal of Advanced Robotic Systems*, 12(11):166–178.

[Zhu et al., 2021] Zhu, H., Claramunt, F. M., Brito, B., and Alonso-Mora, J. (2021). Learning interaction-aware trajectory predictions for decentralized multi-robot motion planning in dynamic environments. *IEEE Robotics and Automation Letters*, 6(2):2256–2263.

[Zhu et al., 2018] Zhu, J., Zou, W., and Zhu, Z. (2018). Learning evasion strategy in pursuit-evasion by deep Q-network. In *International Conference on Pattern Recognition*, pages 67–72.

[Zhu and Martínez, 2010] Zhu, M. and Martínez, S. (2010). Discrete-time dynamic average consensus. *Automatica*, 46(2):322–329.

[Zhu et al., 2013] Zhu, S., Chen, C., Li, W., Yang, B., and Guan, X. (2013). Distributed optimal consensus filter for target tracking in heterogeneous sensor networks. *IEEE Transactions on Cybernetics*, 43(6):1963–1976.