TESIS DE LA UNIVERSIDAD DE ZARAGOZA
2024 188

Rodrigo Aldana López

Distributed cooperation under perception latency and network constraints

Director/es
Aragües Muñoz, Rosario
Sagüés Blázquiz, Carlos

ISSN 2254-7606

Prensas de la Universidad
Universidad Zaragoza

Universidad de Zaragoza
Servicio de Publicaciones

ISSN 2254-7606

Universidad Zaragoza
1542

Tesis Doctoral

DISTRIBUTED COOPERATION UNDER PERCEPTION LATENCY AND NETWORK CONSTRAINTS

Autor

Rodrigo Aldana López

Director/es

Aragües Muñoz, Rosario
Sagüés Blázquiz, Carlos

UNIVERSIDAD DE ZARAGOZA
Escuela de Doctorado

Programa de Doctorado en Ingeniería de Sistemas e Informática

2024

Universidad Zaragoza
1542

Tesis Doctoral

Distributed cooperation under perception latency and network constraints

Autor

Rodrigo Aldana López

Director/es

Rosario Aragües Muñoz
Carlos Sagüés Blázquiz

Escuela de Ingeniería y Arquitectura
Departamento de Informática e Ingeniería de Sistemas

2023

Ph.D. Thesis

Distributed cooperation under perception latency and network constraints

---

Rodrigo Aldana López

Supervisors: Rosario Aragüés
Carlos Sagüés

December 2023

Universidad Zaragoza

Ph.D. Thesis

Distributed cooperation under perception latency and network constraints

Rodrigo Aldana López

Ph.D. Program in Systems Engineering and Computer Science
Universidad de Zaragoza

December 2023

Supervisors:	
Rosario Aragüés	Universidad de Zaragoza, Spain
Carlos Sagüés	Universidad de Zaragoza, Spain
Examiners:	

International reviewers:	
David Gómez-Gutiérrez	Intel Labs, México
Michael Defoort	University of Valenciennes, France

This work was supported by the following agencies/projects:

- MCIN/AEI/10.13039/501100011033, by ERDF A way of making Europe and by the European Union NextGenerationEU/PRTR under projects PID2021-124137OB-I00 and TED2021-130224B-I00.
- Gobierno de Aragón under projects DGA T45-17R and DGA T45-23R.
- COMMANDIA SOE2/P1/F0638 (Interreg Sudoe Programme, ERDF).
- PGC2018-098719-B-I00 (MCIU/ AEI/ FEDER, UE)
- Universidad de Zaragoza and Banco Santander
- Consejo Nacional de Ciencia y Tecnología (CONACYT-Mexico) unders scholarship number 739841.

Resumen

Los sistemas multi-robot están adquiriendo cada vez más importancia para tareas colaborativas, permitiendo aplicaciones que un solo robot no podría manejar por sí solo. Una de estas complejas tareas es la coordinación de múltiples robots para seguir objetivos, con aplicaciones en la transmisión deportiva, la vigilancia, la cinematografía y los sistemas de transporte. Sin embargo, es esencial reconocer las limitaciones prácticas impuestas por los recursos de hardware limitados en estos robots, lo que hace que la colaboración sea aún más desafiante.

En esta tesis, nos centramos en abordar los desafíos en sistemas multi robot que operan en entornos de hardware con recursos limitados, particularmente abordando problemas relacionados con la latencia de percepción y las limitaciones de la red de comunicación. Nuestro marco para el control colaborativo de formaciones alrededor de objetivos en movimiento abarca diversos escenarios que toman en cuenta el uso de recursos. Inicialmente, exploramos escenarios en los que los robots tienen capacidades de percepción a bordo, alimentando datos sin procesar a un mecanismo de percepción, introduciendo una latencia que afecta a la auto-localización y localización del objetivo. Esto plantea un compromiso en el cual una mayor latencia puede mejorar la calidad de la medición a expensas de utilizar datos más antiguos. Además, analizamos situaciones en las que los robots están sujetos a limitaciones de energía y recursos de CPU, lo que requiere la calendarización de la latencia de percepción para aumentar el rendimiento. Por último, analizamos la comunicación entre robots a través de una red, considerando instantes de tiempo discretos asíncronos, sujetos a retrasos y a la conexión o desconexión espontánea de otros robots.

En este contexto, presentamos varias herramientas para abordar incrementalmente cada uno de los problemas previamente descritos en un entorno de control de formación multi-robot y seguimiento de objetivos. En la primera parte de esta tesis, se introducen técnicas de calendarización de latencia de percepción para el control local de un robot y para la estimación del objetivo. Desde la perspectiva del control, discutimos el desafío de mantener la estabilidad de lazo cerrado del sistema bajo la calendarización de latencia de percepción y proponemos técnicas de calendarización que preservan la estabilidad. Desde la perspectiva de la estimación del objetivo, discutimos técnicas de estimación óptima y posibles explosiones combinatorias para el problema de programación. Proponemos una solución eficiente y arbitrariamente precisa para este problema y verificamos su efectividad en simulación y datos reales.

La segunda parte de este documento estudia la herramienta principal propuesta en

este trabajo para la colaboración en cuanto al intercambio de información: el consenso dinámico. Introducimos el concepto de Consenso Dinámico Exacto (EDC por sus siglas en inglés), que requiere que se logre la convergencia a la señal de consenso dinámico de manera exacta, independientemente de si las variables de entrada están variando persistentemente. Comenzamos estudiando las soluciones de EDC en tiempo continuo y en redes de comunicación fijas. Luego extendemos progresivamente a redes abiertas donde los agentes se conectan y desconectan de la red y al caso de comunicación en tiempo discreto asíncrona bajo retrasos. En la tercera parte del documento, discutimos cómo las técnicas de EDC pueden utilizarse para mejorar la localización del objetivo en un entorno multi-robot y aplicamos estas estrategias a un problema de control de formación.

Una de las características más esenciales que distingue este trabajo del resto de la literatura es el uso de algoritmos de consenso no lineales basados en modos deslizantes de alto orden. Mostramos que estos algoritmos pueden superar a sus contrapartes lineales en presencia de ruido y a sus contrapartes de modos deslizantes de primer orden existentes bajo comunicación en tiempo discreto.

También discutimos cómo las soluciones propuestas en esta tesis se extienden más allá del contexto del control de formación. Las técnicas de calendarización de latencia de percepción pueden aplicarse a problemas genéricos de control y estimación. Además, las técnicas de EDC sirven como soluciones genéricas de cómputo distribuido de derivadas. Estas características hacen que nuestras herramientas sean atractivas para la estimación de estado distribuida general para sistemas de entrada desconocida como alternativa al filtrado de Kalman distribuido.

Parte de los resultados de esta tesis se han publicado en revistas de alto impacto: [1–3] en Automatica, [4] en Information Fusion, [5] en ISA Transactions. Varios resultados se han presentado en conferencias internacionales de alto impacto: [6] en 22th IFAC World Congress, [7] en 21th IFAC World Congress, [8] en 59th IEEE Conference on Decision and Control (CDC). Los resultados más recientes [9] se han enviado a IEEE Transactions on Automatic Control. Además de estas publicaciones, participamos en el desarrollo de otros trabajos relacionados resultantes de colaboraciones con diversos miembros del Departamento de Informática e Ingeniería de Sistemas (DIIS) de la Universidad de Zaragoza, así como colaboraciones fuera de la Universidad con instituciones en México, España, Argentina, Austria, Francia e Italia [10–27].

Summary

Multi-robot systems are becoming increasingly important for collaborative tasks, allowing for applications that a single robot could not handle alone. One such complex task is coordinating multiple robots to track targets, which finds use in sports broadcasting, surveillance, cinematography, and transportation systems. Yet, it's essential to acknowledge the practical limitations posed by constrained hardware resources in these robots, making collaboration even more challenging.

In this thesis, we focus on addressing challenges in multi-robot systems operating within resource-constrained hardware environments, particularly tackling issues related to perception latency and communication network limitations. Our framework for collaborative formation control around moving targets encompasses various resource-aware scenarios. Initially, we explore scenarios where robots possess onboard sensing capabilities, feeding raw data to a perception mechanism, introducing a latency that affects self-localization and target localization. This raises a trade-off in which an increasing latency can enhance measurement quality at the cost of using older data. Additionally, we analyze situations where robots are subject to limited power and CPU resources, requiring scheduling of perception latency to increase performance. Lastly, we analyze communication among robots through a network, considering asynchronous discrete time instants, subject to delays and spontaneous connection or disconnection of other robots.

In this context, we present several tools to incrementally tackle each of the previously described problems in a multi-robot formation control and target tracking setting. In the first part of this document, perception latency scheduling techniques are introduced for local control of a robot, and target estimation. From the control perspective, we discuss the challenge of maintaining closed-loop stability of the system under perception latency scheduling and propose stability-preserving scheduling techniques. From the target estimation perspective, we discuss optimal estimation techniques and potential combinatorial explosions for the scheduling problem. We propose an efficient and arbitrarily accurate solution for this problem and verify its effectiveness in simulation and real data.

The second part of this document studies the primary tool proposed in this work for collaboration regarding information exchange: dynamic consensus. We introduce the concept of Exact Dynamic Consensus (EDC), which requires convergence to the dynamic consensus signal to be achieved exactly, regardless if the input variables are persistently varying. We start by studying EDC solutions in continuous-time and fixed communication networks. We extend progressively to open networks where agents connect and disconnect from the network and to the case of asynchronous discrete-time communication under

delays. In the third part of the document, we discuss how EDC techniques can be used to improve target localization in a multi-robot setting and apply these strategies to a formation control problem.

One of the most essential features distinguishing this work from the rest of the literature is the usage of nonlinear consensus algorithms based on high-order sliding modes. We show that these algorithms can outperform their linear counterparts in the presence of noise and their existing first-order sliding mode counterparts under discrete-time communication.

We also discuss how the solutions proposed in this thesis extend beyond the context of formation control. The perception latency scheduling techniques can be applied to generic control and estimation problems. In addition, the EDC techniques serve as generic distributed differentiation solutions. These features make our tools appealing for general distributed state estimation for unknown input systems as an alternative to distributed Kalman filtering.

Part of the results in this thesis have been published in high-impact journals: [1–3] in Automatica, [4] in Information Fusion, [5] in ISA Transactions. Several results have been presented in high-impact international conferences: [6] in the 22th IFAC World Congress, [7] in the 21th IFAC World Congress, [8] in the 59th IEEE Conference on Decision and Control (CDC). The most recent results [9] have been submitted to IEEE Transactions on Automatic Control. In addition to these publications, we participated in the development of other related works resulting from collaborations with various members of the Department of Computer Science and Systems Engineering (DIIS) of the University of Zaragoza, as well as collaborations outside of the University with institutions in Mexico, Spain, Argentina, Austria, France, and Italy [10–27].

Contents

Notation

STANDARD SETS

$\mathbb{N}$	Natural numbers
$\mathbb{Z}$	Integer numbers
$\mathbb{Q}$	Rational numbers
$\mathbb{R}$	Real numbers
$\mathbb{R}_{\geq 0}$	Non-negative real numbers
$\mathbb{R}_{>0}$	Positive real numbers
$\bar{\mathbb{R}}$	$\mathbb{R} \cup \{-\infty, \infty\}$
$\mathbb{R}^n$	n-dimensional Euclidean space
$\{a_k\}_{k=k_1}^{k_2}$	Sequence $\{a_{k_1}, \ldots, a_{k_2}\}$, $k_1, k_2 \in \mathbb{Z}$

SET OPERATIONS

$\lvert S \rvert$	Cardinality of a set S
$\mathsf{len}(a)$	Length m of a sequence $a = \{a_i\}_{i=1}^{m}$
$A \times B$	Cartesian product of sets A, B
A^n	$A \times \cdots \times A$ (n times) for set A
∂S	Boundary of a set $S \subset \mathbb{R}^n$
$\mathsf{int}(S)$	Interior of a set $S \subset \mathbb{R}^n$ ($\mathbb{S} \setminus \partial\mathbb{S}$)
$\mathcal{P}(S)$	Power set of S
$\min\{\bullet\}, \max\{\bullet\}$	Minimum and maximum operators
$\inf(\bullet), \sup(\bullet)$	Infinimum and supremum operators

VECTORS AND MATRICES

bold lower-case	Vectors
bold Upper-case	matrices
$(\bullet)^\top$	Transpose
$\mathsf{tr}(\bullet)$	Trace operator
$\mathsf{vec}(\bullet)$	Vectorization operator
$\text{blockdiag}(\bullet, \dots, \bullet)$	Block diagonal operator
$\mathsf{diag}(\mathbf{x}) \in \mathbb{R}^{n \times n}$	diagonal matrix with $\mathbf{x} \in \mathbb{R}^n$ as the diagonal
$\mathsf{diag}(\mathbf{M}) \in \mathbb{R}^n$	Vector composed by the diagonal elements of $\mathbf{M} \in \mathbb{R}^{n \times n}$
$\otimes$	Kronecker product
$[\mathbf{A}]_{ij}, \mathbf{A}_{ij}$	element i, j of a matrix $\mathbf{A}$
$\|\mathbf{A}\|_F$	Frobenious norm $\left(\sqrt{\sum_{i=1}^n \sum_{j=1}^n [\mathbf{A}]_{ij}^2}\right)$
$\mathbf{A} \succ \mathbf{B}$	$\mathbf{A} - \mathbf{B}$ positive definite
$\mathbf{A} \succeq \mathbf{B}$	$\mathbf{A} - \mathbf{B}$ positive semi-definite

STANDARD VECTORS AND MATRICES

$\mathbb{1}_n$	n-dimensional vector of ones ($[1, \dots, 1]^\top \in \mathbb{R}^n$)
$\mathbb{1}$	Same as $\mathbb{1}_n$, where n is understood by context
$\mathbf{I}_n$	n-dimensional identity matrix
$\mathbf{I}$	Same as $\mathbf{I}$, where n is understood by context
$\mathbf{0}_{n \times m}$	$n \times m$ zero matrix
$\mathbf{0}$	Same as $\mathbf{0}_{n \times m}$ where n, m are understood by context

ANALYSIS

$\frac{\partial V}{\partial \mathbf{x}}$	$\left[\frac{\partial V}{\partial x_1}, \dots, \frac{\partial V}{\partial x_n}\right]$ for scalar function $V : \mathbb{R}^n \to \mathbb{R}$
$L_{\mathbf{F}} V$	Lie derivative of $V : \mathbb{R}^n \to \mathbb{R}$ with $\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^n$

FUNCTIONS OF TIME

t, k	Time variables in continuous and discrete time respectively
$x(t)$	Continuous time signal
$x[k]$	Evaluation of $x(t)$ at instants $\{\tau_k\}_{k=0}^{\infty}$
$\dot{x}(t), \ddot{x}(t)$	First and second time derivatives of $x(t)$
$x^{(\mu)}(t)$	μ-th time derivative of $x(t)$

PROBABILITY

$\mathcal{N}(\bar{\mathbf{x}}, \mathbf{P})$	Gaussian distribution with mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}$
$\mathbb{E}\{\bullet\}$	Expectation operator
$\mathsf{cov}\{\bullet, *\}$	Covariance operator $(\mathbb{E}\{(\bullet - \mathbb{E}\{\bullet\})(* - \mathbb{E}\{*\})^\top\})$
$\mathsf{cov}\{\bullet\}$	$\mathsf{cov}\{\bullet, \bullet\}$

SPECIAL FUNCTIONS AND SYMBOLS

$\text{sign}(x)$	$\begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$
$\lceil x \rfloor^\alpha$	$\lvert x \rvert^\alpha \text{sign}(x)$
$\lceil \mathbf{x} \rfloor^\alpha$	$[\lceil x_1 \rfloor^\alpha, \ldots, \lceil x_n \rfloor^\alpha]^\top$ where $\mathbf{x} = [x_1, \ldots, x_n]^\top$
$\lceil \mathbf{M} \rfloor^\alpha$	Similar to $\lceil \mathbf{x} \rfloor^\alpha$ for each element of matrix $\mathbf{M}$
$\binom{\mu}{\nu}$	Binomial coeficient

Acronyms

ANN Anytime Neural Network.

DAC Dynamic Average Consensus.

DAT Dynamic Average Tracking.

DC Dynamic Consensus.

DDP Distribution Differentiation Protocol.

DOE Discontinuous Optimal Estimation.

EDC Exact Dynamic Consensus.

EDCHO EDC of High Order.

FOSM First Order Sliding Modes.

GPS Global Positioning System.

HGO High Gain Observers.

HOSM High Order Sliding Modes.

IMU Inertial Measurement Units.

IoT Internet of Things.

LMI Linear Matrix Inequality.

MAP Mean Average Precision.

MPC Model Predictive Control.

MSE Mean Squared Error.

OMAS Open Multi-Agent Systems.

PLATE Perception LATency Aware Estimator.

RAS Resource Aware Systems.

RED Robust Exact Differentiators.

REDCHO Robust EDCHO.

RMS Root Mean Squared.

SDE Stochastic Differential Equation.

SLAM Simultaneous Localization and Mapping.

SOE Smooth-Output Estimation.

SP2 Stability Preserving Scheduling Policies.

SVO Semi-direct monocular Visual Odometry.

Chapter One

Introduction

Multi-robot systems are becoming increasingly important in our modern world. The use of these systems involve developing and deploying robots capable of interacting and collaborating with each other and their environment in order to fulfill some prescribed task. Multi-robot systems offer unique advantages, such as improved robustness, scalability, and the ability to tackle complex tasks that would be challenging or impossible for a single robot.

Formation control of multiple mobile robots for target tracking is an actively researched problem with diverse applications [28, 29]. The objective is to enable a team of robots to collaboratively detect the target's location and strategically position themselves around it to gather valuable information. This problem has garnered significant attention due to its relevance in various domains. For instance, in sports broadcasting, a team of robots can track athletes like skiers, runners, cyclists, and other sports enthusiasts, as depicted in Figure 1.1. The collected information can then be processed to create immersive videos from multiple perspectives. However, the potential applications of formation control extend beyond sports broadcasting, finding uses in automated surveillance systems [30, 31], aerial cinematography using drones [32], and intelligent transportation systems [33], among numerous other examples.

Developing autonomous multi-robot systems requires to study the fundamental capabilities of perception, control, planning, and communication [34].

Perception: It refers to the process of transforming raw sensor data into meaningful information for the robot. For example, a robot with onboard cameras captures images at regular intervals. During the perception stage, the pixel information from these images can be processed to determine the robot's position in relation to a known frame of reference. This perception problem referred to as localization, has been extensively studied in the literature [35]. One possibility is to use markers or fiducials in the environment that are detected by a camera, enabling measurement of the robot's position relative to these markers. Odometry, on the other hand, uses one or multiple cameras to measure sequential displacements from an initial position. However, odometry can suffer from error accumulation unless the robot incorporates environmental features to correct drift. To solve this problem, a map can be constructed based on previous information. In this context, Simultaneous Localization and Mapping (SLAM) is an active area of research

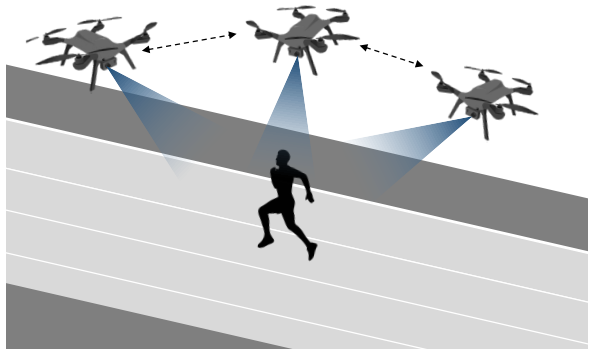

Figure 1.1: A team of drones is deployed in order to track an athlete. In this example, the robots, which produce measurements for the target using onboard sensors, coordinate by sharing information (dotted lines) in order to improve the joint tracking performance.

with numerous solutions available [36, 37]. Additionally, other sensors like radar and lidar can be integrated into these systems.

In addition to using exteroceptive sensors like vision, lidar, and radar, robots can incorporate proprioceptive sensors such as encoders to measure the positions and orientations of internal links within the robot. Other sensors in this context are Inertial Measurement Units (IMUs), which consist of accelerometers, gyroscopes, and magnetometers. Data obtained from these sensors can be used to enhance the pose estimation of the robot [34].

Another perception problem of interest based on similar sensors is object detection in the environment [38]. Object detection serves two purposes. Firstly, detected objects can be used as high-level semantic features to enhance localization [39]. Secondly, objects are detected to fulfill specific tasks such as target detection and tracking. Besides formation control, target tracking had variety of other applications, including crowd analysis and pedestrian intention prediction [40], gesture recognition [41], and traffic monitoring [42].

Planning: It refers to the process in which the robot determines its actions based on its current configuration, the map, and the desired goal. The planning output typically consists of a trajectory that enables the robot to navigate the map and reach a specific destination in the case of mobile robots. The plan can be composed of setpoints, representing locations the robot needs to reach, a path in the map, or a comprehensive time-dependent trajectory that considers velocities or other higher order dynamics. While this process generates the motion plan, it is not responsible for executing it.

Control: Once the robot has information about its own state and the planned actions, the control stage is in charge of computing and transmitting action commands to the robot's actuators. The actuators encompass various components such as servomotors, wheel motors, or propeller motors, which depend on the specific mechanisms used by the robot. The control commands are determined using the feedback provided by the error between the current robot state and the desired state outlined in the motion plan. The objective is to correct the state, ensuring the robot moves closer to its intended trajectory.

Communication: To enable cooperation among robots, some form of interaction must be established. This interaction can be passive, where robots detect each other

using the perception stage and adjust their plans accordingly. However, in many cases, explicit information exchange between robots occurs through digital communication systems. This aspect has gained significance in recent times, particularly with the emergence of Internet of Things (IoT) technologies, which have made fast and reliable wireless communication devices more affordable. As a result, networked controlled systems in numerous industrial applications have transitioned from wired sensor-actuator feedback systems to wireless network systems. In these environments, robots can collaborate by transmitting information to neighboring robots in the environment using on-board radios, leveraging the capabilities of wireless communication for enhanced cooperation.

1.1 RESOURCE AWARE SYSTEMS

In addition to the previously discussed fundamental capabilities, the overall system's performance is also influenced by the characteristics of the hardware and software employed in the platform. This aspect becomes crucial in cheap small robots, where such details can no longer be overlooked. Factors such as sensor accuracy, processing time, sampling frequency for sensors and actuators, and delays in the communication network can significantly impact system performance.

Systems that take into account these considerations are referred to as Resource Aware Systems (RAS) [43, 44]. Adopting a resource-aware perspective is essential for studying the interplay between different robot capabilities. Within the scope of this work, two specific resource constraints are of particular interest: perception latency and communication network limitations.

1.1.1 Perception latency and accuracy trade-off

The term perception latency has been defined in [45] as the time taken for the perception process to generate an output, starting from the moment when raw sensor data is retrieved. In many algorithms, the perception latency significantly impacts the quality of the perception output. For instance, in tasks such as localization, odometry, and SLAM, improving the perception quality often involves using higher image resolutions, increasing the number of features extracted from each image, or enhancing the map's size or resolution. However, these improvements generally come at the cost of increased perception latency. This trade-off has been verified experimentally for visual odometry, localization, and mapping in [46, 47]. In addition, it has been shown that deep learning models used for perception share this type of trade-off as well [48].

To illustrate this perception-latency and accuracy trade-off, a study conducted in [46] explored different parameter settings of a popular vision-based odometry solution called Semi-Direct Monocular Visual Odometry (SVO) [48]. These results are summarized in Figure 1.2, showing the relationship between perception error and perception latency. It indicates that a robot (such as a drone) can obtain information about its position in the environment with a perception latency that increases with the desired precision. In this case, if greater accuracy is required, a more significant latency is introduced, which can diminish the utility of the data for control purposes or render the measurements useless if the latency becomes substantial. Moreover, while this decreasing behaviour for the estimation error with respect to perception latency is usually expected, [46] discusses how

increasing the latency may increase the error depending on the particular environment conditions and motion of the robot.

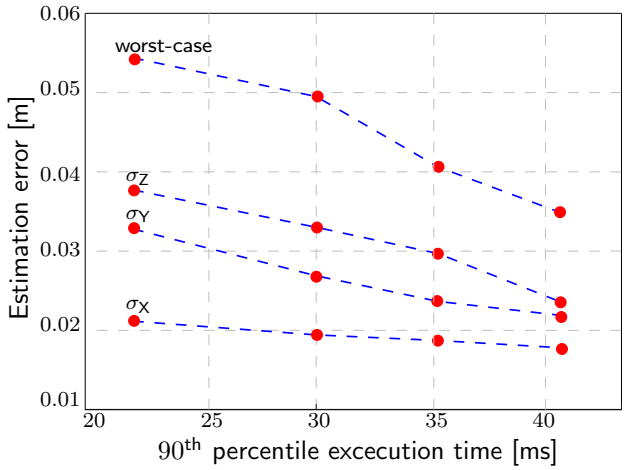

Figure 1.2: Execution time (perception latency) vs estimation error experiments for the Semi-Direct Monocular Visual Odometry (SVO) [48] used for localization on board of a multi-rotor. Standard deviations $\sigma_X, \sigma_Y, \sigma_Z$ for the estimation error for each axis is shown as well as the overall worst case error. The 90^{th} percentile execution time is shown due the variability across the experiment. Different configurations for SVO were tested, mainly varying the maximum number of features used in the algorithm. As the execution time increase, the error decreases for these experiments. Data borrowed from [46, Figure 9].

A similar trade-off appears in other forms of perception such as target or object tracking. In this case a sensor (such as a camera) takes a raw measurement from the environment (such as an image), and aims to detect in which pixels, or regions of pixels an object of interest lies in the captured image or in a common frame of reference. The impact of perception latency and quality becomes more apparent with the emergence of many state-of-the-art neural network-based object detectors like [49–53]. These solutions tend to follow scaling laws, leveraging the increasing network parameters to train on larger datasets and achieve better generalization, enhancing object detection quality. As expected, Figure 1.3 demonstrates how object detection accuracy increases with neural network inference time for various popular architectures and configurations.

Therefore, the trade-off between perception latency and precision is a critical concern in RAS. The most common approach to address this challenge is to establish a fixed sampling step for the perception process, which is often dictated by the control task considering the mechanical speed and physical capabilities of the robot. This fixed sampling step imposes a real-time constraint on the perception process, requiring it to produce an output with a predetermined configuration to ensure stability guarantees in the controller stage. This approach has been used repeatedly in the robotics literature, e.g. for multi-rotor control using onboard vision sensors [18, 55, 56]

In certain cases, a known computing budget may be available for the perception process to adapt. For instance, if higher-priority tasks are running on the computing platform, the remaining computing resources can be estimated to allocate an appropriate

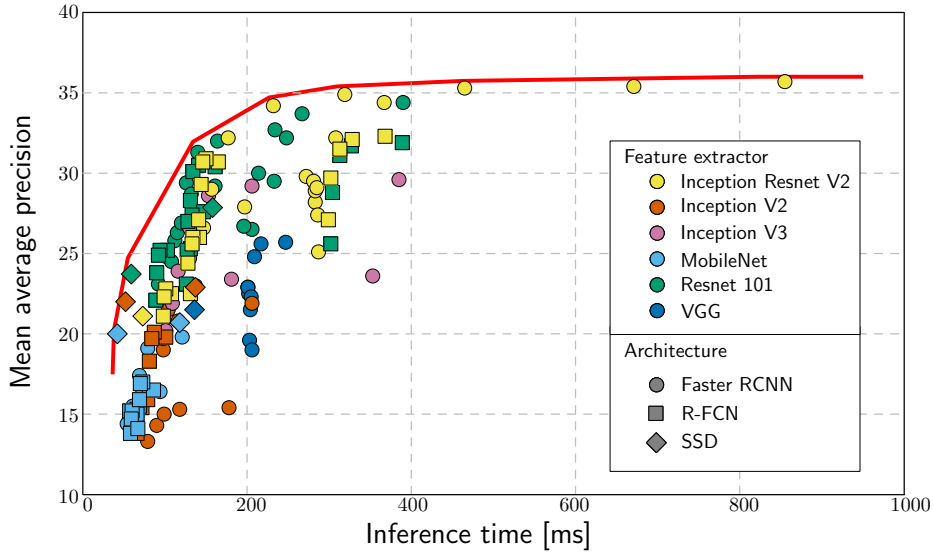

Figure 1.3: Precision vs inference time experiments for different object detectors. Different feature extractors and neural network architectures are tested (See [54] for a detailed exposition for these variations). The mean average precision (MAP) is a composite metric based on several other standard accuracy metrics, where a bigger MAP means a better result. The results suggest that a bigger MAP can be obtained with diminishing returns using configurations with bigger inference time. Data borrowed from [54, Figure 11].

perception process that fits within the budget. In this context, Anytime Neural Networks (ANN) may be employed for neural-network-based perception methods. These networks can produce successive output versions, gradually improving the quality until a desired level is achieved. Such networks can be constructed using multiple hidden layers with multiple early outputs, as demonstrated in [57, 58]. Recurrent iterative networks offer another approach to obtaining ANN, where the network's output is used as its input in subsequent steps for further refinement [59].

Perception latency scheduling: Recent advancements in reconfigurable algorithms, such as ANN with variable perception latency, suggest that control performance enhancement via dynamically adjusting the perception latency based on the current system demands may be possible. In specific scenarios, the required update rate for perception may vary [60]. For instance, a stationary robot in a relatively static environment may not need frequent perception updates compared to a fast-moving robot in a dynamic environment. Moreover, some tasks may not prioritize precision, allowing for perception latency savings. Hence, adjusting the perception latency according to the particular situation can be beneficial.

Furthermore, it is essential to consider that perception is just one of the tasks running on the robot's computing platform. Limited processing power is shared among various tasks, including control, planning, communication, and higher-level functions like learning. Therefore, scheduling the perception latency to optimize CPU load becomes crucial

for achieving autonomy in resource-constrained robots.

Moreover, sensors directly consume additional power, as observed with radar or lidar devices. Considering the limited battery capacity carried by robots, it is energetically sensible to maintain a small number of sensing events. Additionally, sensors typically share I/O buses with other devices, and limiting the number of transaction events helps prevent bus congestion.

1.1.2 Network induced constraints

To enable information exchange between multiple robots, a wireless communication network is required, implying the use of radio devices and the implementation of specific communication protocols. Various examples of communication protocols and network stacks are employed for multi-robot systems and sensor networks, such as IEEE 802 specifications for Bluetooth [61], ZigBee [62], WiFi [63], or proprietary protocols.

However, as described in [64], these communication networks introduce certain imperfections that deviate from the expected nominal behavior. Several issues arising from the network itself can be observed, including:

- **Time delays:** Information requires a certain amount of time to travel from one network node to another. This delay is caused by physical travel delay, protocol and software overhead, and re-transmissions due to errors or collisions between communication packets.
- **Clock synchronization issues among network nodes:** In practical scenarios, synchronization among nodes is imperfect. Even slight clock differences can result in significant clock drifting over time. Moreover, this lack of synchronization hinders agents from exchanging information periodically across the network.
- **Time-varying packet transmission/sampling intervals:** Despite the intention to transmit messages periodically within a single node, the protocol overhead, network stack implementation, and re-transmissions due to errors lead to aperiodic or asynchronous message transmission to other agents.
- **Arbitrary connection and disconnection of nodes:** In many real-world applications, the network configuration cannot be assumed to be static. Nodes may connect or disconnect from the network based on distance constraints between robots moving in the environment. Additionally, depending on the task at hand, it may be beneficial to merge or split the network accordingly.

Robust open multi-robot systems: The challenges outlined earlier highlight the need for robust solutions to address the imperfections inherent in communication systems. In this regard, the concept of Open Multi-Agent Systems (OMAS) [65] refers to a collection of multiple agents that interact through a communication network, subject to the random arrival and departure of agents and transmissions. Extending this concept to develop algorithms for open multi-robot systems, resilient to time delays and irregular transmissions becomes highly desirable in the context of this thesis.

Distributed, decentralized, and one-hop solutions: When robots use their local sensors to gather information from the environment, there is distributed information across the network. One approach to leverage this distributed information is to designate

a leader robot or incorporate a central computing platform that receives all data and obtains a centralized estimation of the environment. However, due to the potential issues mentioned earlier, solutions that distribute decision-making among the robots in the network are preferred over relying solely on a single leader robot or central computer. The reason for favoring distributed decision-making is that the communication link connecting the leader with the rest of the network, or the leader's computational system itself, may fail [66]. In contrast, each node in the network can act as a central node, aiming to receive information from all other agents through a multi-hop communication setting. However, this approach, known as *flooding*, also presents challenges. Firstly, information can be further delayed due to multi-hop communication. Additionally, the network can become congested with additional messages from each node to every other node. This renders the flooding solution non-scalable concerning the network size, as the number of communication links grows quadratically. Furthermore, similar to a centralized framework, this approach requires that each node has a unique identifier, which restricts its use in an OMAS setting. As a result, a preferable RAS solution is a distributed and decentralized framework, that uses single-hop communication [67].

1.1.3 Resource aware framework for formation control

To provide a unified perspective on the contributions of this thesis, we envision a practical scenario where a group of robots is assigned to track a specific target, subject to the resource constraints mentioned before. These robots are equipped with perception sensors and wireless communication capabilities. They aim to work together to detect the target's location and position themselves accordingly, even if the target follows a complex and dynamic trajectory.

We establish a shared architecture that underlies the majority of the findings presented in this thesis, representing an abstract depiction of the capabilities of each robot involved, as depicted in Figure 1.4. The properties of each component within this framework may vary depending on the assumptions made about the actual real-world environment of interest. The features of this model are described as follows:

- **Hybrid system:** We assume that the robot have continuous-time state space dynamics. In contrast, we assume that the robot is equipped with a computing platform that acts in a discrete-time setting for decision making. Henceforth, the overall system is of hybrid nature.

- **Adaptive perception modes:** The robot is equipped with various sensors (cameras, lidars, radars, IMUs) that capture raw samples to obtain information about the robot's state and the environment. This raw data is processed using a localization module (e.g., SLAM algorithm) for robot position measurement and an object detection module for obtaining information about the target of interest. The perception latency of these methods can be configured in real-time. This can be achieved through a bank of interchangeable perception modules that can be swapped online, or by anytime perception methods.

- **Consensus-based information fusion:** In the right section of Figure 1.4, the object detection component generates positional information for the target. This information may be incomplete due to occlusions or may be inaccurate. To address these limitations, agents communicate via radio and employ information fusion

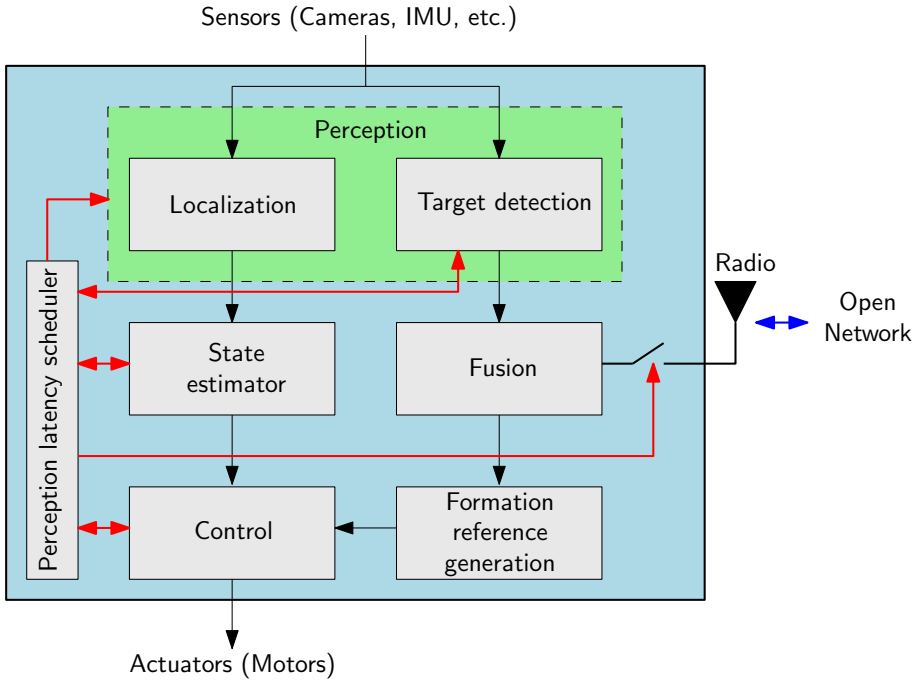

Figure 1.4: Individual robot architecture for multi-robot formation control encompassing all the features of interest in this thesis. The blue box consists on the software components that run in the robot computing platform, with the green box being the perception stage. Outside of the blue box are the components equipped on-board of the robot with which it interacts with the environment: sensors, actuators and a radio to communicate with other agents.

based on consensus techniques. The obtained positional information establishes a formation reference for the robots, allowing them to position themselves relative to the target within a shared frame of reference. Simplifying the planning stage, the robots track a fixed displacement relative to the target's position.

- **Asynchronous communication over open networks:** The radio depicted in the rightmost part of Figure 1.4 allows the robot to communicate with the rest of the robots in the network. However, we assume that the network is open in the sense that it allows spontaneous connection or disconnection of robots, and that only asynchronous discrete-time communication is possible.

- **Perception latency aware estimation and control:** In the left part of Figure 1.4, a state estimator that uses position measurements from the perception stage to estimate the robot's current state is included. This estimated state and the current reference information from the planning or fusion stages are used in the control stage. The control stage computes commands for the robot's actuators, considering both the estimated state and the desired reference information. These commands are sent to the actuators, enabling the robot to execute the desired environmental

movements and responses.

- **Perception latency scheduling:** In addition to the traditional control input connected to the actuators, we introduce another control input or virtual actuator in our system. This virtual actuator corresponds to the perception latency used in the perception stage, which can be adaptively chosen. The adaptation of the perception latency is achieved through a perception latency scheduler, whose objective is to maintain system stability, ensure satisfactory tracking results, and optimize resource usage. It is worth noting that the perception latency scheduler also can adjust the control law employed by the actuators.

1.2 LITERATURE REVIEW

There are some important gaps in the literature related to the challenges of perception latency scheduling, network constraints and resource management in the context of collaborative formation tracking of multi-robots. In this section, we provide a concise overview of the main issues with current solutions and highlight notable works in each area. For a more detailed discussion on related work, please refer to the corresponding chapters.

Perception latency schedulers for control: The literature contains various works that highlight a similar trade-off between latency and accuracy in different contexts, including communication in networked control systems and perception in robot systems [46, 68, 69]. Some studies have explored scheduling sampling instants using dynamic programming [58], event-triggered techniques [70, 71], delay-based analysis [72], and model predictive controllers [46], among others. However, there is a lack of consideration for the interplay between latency adaptation, resource usage, and the performance of control or estimation tasks in general. For example, as will be explored later in this thesis, the stability of the overall closed-loop system may be compromised under arbitrary scheduling.

Perception latency schedulers for estimation: In the context of target trajectory estimation with vision sensors, frame-skipping techniques have been employed to reduce computational load. Some works have proposed event-triggered [60, 73] and machine learning-based [74] approaches for this purpose. While these methods demonstrate promising experimental performance in specific scenarios, they heavily rely on heuristic rules, making it difficult to provide formal guarantees on resource usage and estimation quality for target tracking in general cases.

The multi-sensor scheduling problem is relevant in the perception scheduling context [75, 76]. It involves scheduling multiple sensors to measure the state of a dynamical system for control or estimation, considering a given cost function. This problem often faces a combinatorial explosion due to the number of available sensors and the time horizon for the cost function. There exist solutions for this problem in the literature, such as looking only for periodic solutions [77–79], using pruning of schedules [80], applying greedy solutions [81], or tailored made methods taking advantage the structure of the cost function [82, 83]. However, these solutions are often very conservative or cannot be extended for different performance metrics to be optimized, such as the RAS-based costs we consider in this thesis.

Exact consensus under persistently varying input information: Average consensus algorithms have become increasingly popular building blocks for more complicated distributed one-hop algorithms such as distributed sensor fusion or distributed state es-

timation. When the input information to be fused across the agents change in time, dynamic consensus algorithms are employed, aiming to track time-varying average signals. The most popular approach for this problem is to rely on linear consensus [84] updated at each agent based on information shared between neighbors. An extensive survey of these techniques can be found in [67].

One of the most important drawbacks of these methods that draws the attention, is that no exact convergence can be guaranteed under persistently varying input signals. In fact, asymptotic convergence for the consensus error towards the origin is often highlighted only when the inputs become constant asymptotically as well. In the general case, error bounds are provided based on bounds of derivatives of the input signals. There are algorithms in the literature achieving exact convergence even under persistently varying inputs such as in [85], being one of the most prominent works in this context. However, in order to achieve such exact convergence, the introduction of sliding mode control terms need to be used in the consensus algorithm for continuous-time analysis, leading to the introduction of chattering in actual discrete-time implementations [86].

Distributed state estimation: There are various strategies in the literature to fuse local Kalman filter estimations through fusion centers [87, 88] or decentralized static consensus methods [89], linear dynamic consensus filters [90, 91], and covariance intersection-based consensus filters [92–94]. While most of these methods work in either continuous-time or discrete-time, this thesis aims to incorporate a distributed state-estimator for a hybrid system with continuous-time dynamics and discrete-time measurements due to adaptive latency in the perception stage. However, applying standard filtering techniques to this system for generating robot trajectories to track the target may lead to a lack of asymptotic closed-loop stability, as discussed later in subsequent chapters.

Moreover, most of the distributed state estimation literature relies on Kalman filtering theory. However, recent research has highlighted nonlinear estimation techniques for linear systems using differentiation methods. For instance, in [95], the significance of numerical exact differentiation for unknown-input observer design was discussed. In this context, Robust Exact Differentiators (RED) [96] have been developed as non-linear observers capable of achieving exact differentiation without noise for a specific class of input signals with bounded high-order derivatives. In the presence of noise, the RED demonstrates optimally shaped error bounds. Despite these advancements, there is a gap in the literature as no distributed differentiator has been proposed that generalizes the capabilities of the RED.

Asynchronous communication: Most existing distributed consensus algorithms assume either continuous-time or discrete-time communication. In the case of discrete-time communication, it is often assumed that communication instants are synchronized across the network [97–99]. However, there are works that allow asynchronous communication intentionally [100–103]. Asynchronous communication can also be understood in terms of communication delays [104, 105]. It is important to note that most of these approaches use linear protocols, which have significant precision drawbacks as discussed earlier. Event-triggered algorithms, such as the one proposed in [106], also produce asynchronous communication instants. Despite this, it is assumed that these instants are a controlled variable rather than arbitrary imperfections in the network, which limits their applicability for the considered RAS context in this work.

1.3 OBJECTIVES

As noted in the previous section, there is a notable gap in existing research and solutions concerning the collaboration of robots in a setting that considers various challenges such as perception latency, resource usage, and asynchronous communication. The main objective of this work is to advance the development of such problems in the context of the architecture of Figure 1.4. For this purpose, the following objectives need to be addressed:

- Develop perception latency-aware scheduling solutions for control systems and target estimation, aiming for an improved resource usage.
- Develop novel information fusion techniques based on distributed, decentralized, and one-hop dynamic average consensus.
- Develop perception latency-aware information fusion techniques for formation control.
- Analyze the performance of the proposed algorithms under network constraints such as asynchronous communication and delays.

To fulfill these objectives, a formal framework will be employed, ensuring stability and robustness, and proving a computational complexity analysis. The formal guarantees and analysis in this thesis provide several practical insights as well, which will be important when considering the implementation of RAS systems such as the one depicted in Figure 1.4.

1.4 CONTRIBUTIONS

The main contribution of this thesis is a comprehensive framework that addresses the challenges of perception latency scheduling, network constraints and resource management in the context of collaborative formation tracking of multi-robots. The developed solution provides a practical framework and highlights the many challenges to be further explored in future research.

This thesis follows an incremental approach, presenting results from simpler to more complex settings. A modular philosophy is adopted, addressing individual modules of Figure 1.4 separately for clarity and focus. By gradually integrating these modules, a comprehensive understanding of the proposed solution is achieved. Table 1.1 categorizes scenarios based on the presence or absence of perception latency affecting sensing (horizontal axis) and the communication capabilities of the robots (vertical axis). The first column and row represent mono-robot scenarios without communication or perception latency and thus is not of interest in this work.

In this context, the contributions of this thesis are enlisted as follows:

- We provide the first-of-its-kind solutions for perception latency scheduling:
  - **Stability Preserving Scheduling Policies (SP2):** A novel framework for perception latency scheduling for control, ensuring closed-loop stability.

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="2">Perception</td></tr>
<tr><td>Ideal</td><td>Adaptive latency</td></tr>
<tr><td rowspan="4">Communications</td><td>No</td><td>Standard mono-robot</td><td>Control: SP2 (Chapter 2)
Estimation: PLATE (Chapter 3)</td></tr>
<tr><td>Continous-time</td><td>EDCHO (Chapter 4)</td><td rowspan="2">Perception latency aware formation control (Chapter 7)</td></tr>
<tr><td>OMAS</td><td>REDCHO (Chapter 5)
DDP (Chapter 6)</td></tr>
<tr><td>Asynchronous (Discrete-time)</td><td colspan="2">Asynchronous EDC (Chapter 8)</td></tr>
</table>

Table 1.1: Organization of the chapters in this thesis, according to how resource constraints in perception and communications are handled.

  - **Perception LATency Aware Estimator (PLATE):** A novel estimation framework for target tracking compatible with modern perception methods using deep learning. Efficient approximate near-optimal solutions for perception latency scheduling are presented, addressing the combinatorial explosion challenge.

  While these results are presented in the context of a single-agent system, they can be applied to multi-robot scenarios.

- As a basis for information fusion algorithms, we provide novel solutions for dynamic consensus, introducing the concept of Exact Dynamic Consensus (EDC) for precise convergence in finite time, even under persistently varying information sources of information. Our framework effectively reduces chattering compared to existing solutions in the literature. Three main algorithms are presented:
  - **EDC of High Order (EDCHO):** The first EDC protocol that features high-order sliding modes to reduce chattering.
  - **Robust EDCHO (REDCHO):** An extension of EDCHO designed to work in Open Multi-Agent Systems (OMAS), accommodating spontaneous connections or disconnections of agents within the network.
  - **Distributed Differentiation Protocol (DDP):** An extension of REDCHO capable of accurately computing derivatives of the consensus signal, even in the presence of measurement noise.
  - **Asynchronous EDC:** We propose a new consensus protocol under asynchronous discrete communication and OMAS. As a result, it can be directly

applied in conjunction with other results in this thesis, with or without perception latency scheduling, allowing to communicate only when the perception stage produce new information.

- **Perception latency aware formation control:** We introduce an estimator design compatible with our consensus solutions, enabling information fusion for target tracking under arbitrary perception latency schedules. This novel approach allows for perception latency scheduling at each robot, marking the first incorporation of perception latency scheduling effects in a multi-robot context.

All the methods and modules presented in this work come with a rigorous formal analysis including stability, robustness and computational complexity. Most proofs are based on tools from control theory such as hybrid and switching systems analysis, stochastic system analysis and optimal filtering, as well as Lyapunov and non-linear system analysis.

1.4.1 Research output

The results presented in this thesis are a compilation of several peer-reviewed publications resulting from the work in this Ph.D.:

Journal articles:

1. R. Aldana-Lopez, R. Aragues, and C. Sagues, "Quasi-exact dynamic consensus under asynchronous communication and symmetric delays," *Submitted to IEEE Transactions on Automatic Control* [9].

2. R. Aldana-Lopez, R. Aragues, and C. Sagues, "Perception-latency aware distributed target tracking," *Information Fusion*, p. 101857, 2023 [4].

3. R. Aldana-Lopez, R. Aragues, and C. Sagues, "PLATE: A perception-latency aware estimator," *ISA Transactions*, vol. 142, pp. 716–730, 2023 [5].

4. R. Aldana-Lopez, R. Aragues, and C. Sagues, "Latency vs precision: Stability preserving perception scheduling," *Automatica*, vol. 155, p. 111123, 2023 [3].

5. R. Aldana-Lopez, R. Aragues, and C. Sagues, "REDCHO: Robust exact dynamic consensus of high order," *Automatica*, vol. 141, p. 110320, 2022 [2].

6. R. Aldana-Lopez, R. Aragues, and C. Sagues, "EDCHO: High order exact dynamic consensus," *Automatica*, vol. 131, p. 109750, 2021 [1].

Conference articles:

1. R. Aldana-Lopez, R. Aragues, and C. Sagues, "Distributed differentiation with noisy measurements for exact dynamic consensus," *IFAC-PapersOnLine*, 2023. 22th IFAC World Congress [6].

2. R. Aldana-Lopez, R. Aragues, and C. Sagues, "Attention vs. precision: latency scheduling for uncertainty resilient control systems," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 5697–5702, 2020 [8].

3. R. Aldana-Lopez, R. Aragues, and C. Sagues, "EDC: Exact dynamic consensus," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 2921–2926, 2020. 21th IFAC World Congress [7].

In addition to these publications, other related works were developed during the course of the Ph.D. between 2019 and 2023, which were not included in the body of this thesis for the sake of clarity and conciseness. These works resulted from collaborations with various members of the Department of Computer Science and Systems Engineering (DIIS) of the University of Zaragoza, as well as collaborations outside of the University with institutions in Mexico, Spain, Argentina, Austria, France, and Italy. These collaborative efforts cover related topics on control theory, estimation, distributed systems and communication systems. Moreover, they have contributed to a broader scope of research and enriched the overall findings and impact of the Ph.D. project.

Other journal articles:

1. R. Aldana-Lopez, M. Aranda, R. Aragues, and C. Sagues, "Robust affine formation tracking," *Submitted to IEEE Transactions on Automatic Control* [27].

2. I. Perez-Salesa, R. Aldana-Lopez, and C. Sagues, "Remote estimation with bounded uncertainty under dynamic event-triggered communication," *Submitted to IEEE Transactions on Systems, Man, and Cybernetics* [26].

3. R. Aldana-Lopez, E. Sebastian, R. Aragues, E. Montijano, and C. Sagues, "Distributed outer approximation of the intersection of ellipsoids," *IEEE Control Systems Letters*, pp. 1–1, 2023 [107].

4. D. Gomez-Gutierrez, R. Aldana-Lopez, R. Seeber, M. T. Angulo, and L. Fridman, "An arbitrary-order exact differentiator with predefined convergence time bound for signals with exponential growth bound," *Automatica*, vol. 153, p. 110995, 2023 [11].

5. I. Perez-Salesa, R. Aldana-Lopez, and C. Sagues, "Precise dynamic consensus under event-triggered communication," *Machines*, vol. 11, no. 2, 2023 [10].

6. R. Aldana-Lopez, R. Seeber, D. Gomez-Gutierrez, M. T. Angulo, and M. Defoort, "A redesign methodology generating predefined-time differentiators with bounded time-varying gains," *International Journal of Robust and Nonlinear Control*, pp. 1–16, 2022 [12].

7. R. Aldana-Lopez, D. Gomez-Gutierrez, R. Aragues, and C. Sagues, "Dynamic consensus with prescribed convergence time for multileader formation tracking," *IEEE Control Systems Letters*, vol. 6, pp. 3014–3019, 2022 [13].

8. A. Ramirez-Perez, R. Aldana-Lopez, O. Longoria-Gandara, J. Valencia-Velasco, L. Pizano-Escalante, and R. Parra-Michel, "Modular arithmetic cpm for sdr platforms," *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1–1, 2022 [14].

9. H. Haimovich, R. Seeber, R. Aldana-Lopez, and D. Gomez-Gutierrez, "Differentiator for noisy sampled signals with best worst-case accuracy," *IEEE Control Systems Letters*, pp. 1–1, 2021 [15].

10. R. Aldana-Lopez, D. Gomez-Gutierrez, M. A. Trujillo, M. Navarro-Gutierrez, J. Ruiz-Leon, and H. M. Becerra, "A predefined-time first-order exact differentiator based on time-varying gains," *International Journal of Robust and Nonlinear Control*, vol. 31, pp. 5510–5522, 2021 [16].

11. R. Aldana-Lopez, D. Gomez-Gutierrez, E. Jimenez-Rodriguez, J. D. Sanchez-Torres, and M. Defoort, "Generating new classes of fixed-time stable systems with predefined upper bound for the settling time," *International Journal of Control*, vol. 0, no. 0, pp. 1–13, 2021 [17].

12. L. Campos-Macias, R. Aldana-Lopez, R. de la Guardia, J. I. Parra-Vilchis, and D. Gomez-Gutierrez, "Autonomous navigation of mavs in unknown cluttered environments," *J. Field Rob.*, vol. 38, no. 2, pp. 307–326, 2021 [18].

13. M. Trujillo, R. Aldana-Lopez, D. Gomez-Gutierrez, M. Defoort, J. Ruiz-Leon, and H. M. Becerra, "Autonomous and non-autonomous fixed-time leader-follower consensus for second-order multi-agent systems," *Nonlinear Dyn.*, vol. 102, pp. 1–18, 12 2020 [19].

14. R. Aldana-Lopez, D. Gomez-Gutierrez, E. Jimenez-Rodriguez, J. Sanchez-Torres, and A. Loukianov, "On predefined-time consensus protocols for dynamic networks," *Journal of the Franklin Institute*, vol. 357, no. 16, pp. 11880–11899, 2020. Finite-Time Stability Analysis and Synthesis of Complex Dynamic Systems [20].

15. J. D. Sanchez-Torres, A. J. Munoz-Vazquez, M. Defoort, R. Aldana-Lopez, and D. Gomez-Gutierrez, "Predefined-time integral sliding mode control of second-order systems," *International Journal of Systems Science*, vol. 51, no. 16, pp. 3425–3435, 2020 [21].

16. J. Valencia-Velasco, O. Longoria-Gandara, R. Aldana-Lopez, and L. Pizano-Escalante, "Low-complexity maximum-likelihood detector for IoT BLE devices," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4737–4745, 2020 [22].

Other conference articles:

1. I. Perez-Salesa, R. Aldana-Lopez, and C. Sagues, "Event-triggered consensus for continuous-time distributed estimation," *IFAC-PapersOnLine*, 2023. 22th IFAC World Congress [25].

2. I. Perez-Salesa, R. Aldana-Lopez, and C. Sagues, "Event-based visual tracking in dynamic environments," in *ROBOT2022: Fifth Iberian Robotics Conference* (D. Tardioli, V. Matellan, G. Heredia, M. F. Silva, and L. Marques, eds.), (Cham), pp. 175–186, Springer International Publishing, 2023 [24].

3. E. Jimenez-Rodriguez, R. Aldana-Lopez, J. D. Sanchez-Torres, D. Gomez-Gutierrez, and A. G. Loukianov, "Consistent discretization of a class of predefined-time stable systems," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 628–633, 2020. 21st IFAC World Congress [23].

1.5 DOCUMENT ORGANIZATION

The document is conceptually divided in three parts.

1. The first part, (Chapters 2 and 3) deals with the blocks *State estimator, Control* and *Perception latency scheduler* in Figure 1.4. Perception latency scheduling techniques are introduced for local control of a robot and for target detection and tracking.

2. The second part (Chapters 4, 5 and 6) deals with the *Fusion* block in Figure 1.4 by discussing dynamic (average) consensus techniques, serving as a basis for information fusion. We introduce novel techniques for dynamic consensus, discussing robustness in open networks and distributed differentiation techniques.

3. The third part (Chapters 7 and 8) deals with the interplay between perception latency scheduling techniques presented in the first part and consensus ideas in the second part. Adaptation of the methods for both estimation and consensus is performed to accommodate for the complete architecture. We present a solution for distributed information fusion under perception latency scheduling, and an extension of our dynamic consensus methods for open networks with asynchronous communications.

Chapter Two

Perception Latency Scheduling In Control

In this chapter, we explore the perception latency scheduling problem for control, addressing its complexity, main challenges, and proposing practical algorithms and formal results in this context. Following the modular philosophy presented in Chapter 1, we start by analysing the problem with a single robot. We consider an architecture of the form depicted in Figure 2.1, incorporating the perception stage for self-localization of the robot, a state estimator, and a controller. The primary objective is for the robot to track a reference. This reference may originate from the specific single-robot application or result from distributed observations of a target by multiple robots, as discussed in subsequent chapters.

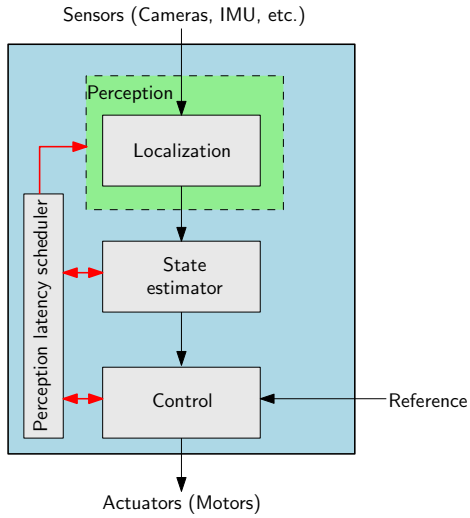

Figure 2.1: Model for the individual robot architecture with perception latency scheduler for control.

Now, we explain the particular perception latency model we will use throughout this

thesis. First, consider a robot model of the following form:

$$\mathrm{d}\mathbf{x}(t) = (\mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}[k])\,\mathrm{d}t + \mathrm{d}\mathbf{w}(t) \tag{2.1}$$

for intervals $t \in [\tau_k, \tau_{k+1})$ where $\mathbf{x}(t) \in \mathbb{R}^n$ is the robot state (e.g. position and velocity for a particle robot), $\mathbf{u}[k] \in \mathbb{R}^{n_\mathbf{u}}$ is a zero-order hold input (e.g. representing the actuator forces induced to the system by the robot) and a sequence of instants $\tau = \{\tau_k\}_{k=0}^{\infty}$ with $\tau_k \in \mathbb{R}$, $\tau_k < \tau_{k+1}$ and $\tau_0 \equiv 0$. The matrices $\mathbf{A}, \mathbf{B}$ are of appropriate size.

We adopt a linear model represented by (2.1), as it facilitates the derivation of various results in closed form, enhancing the understanding of the perception latency scheduling problem. To account for potential non-modeled nonlinear dynamics or disturbances, we introduce the n-dimensional stochastic process $\mathbf{w}(t)$. Consequently, (2.1) transforms into a Stochastic Differential Equation (SDE).

Perception mechanism model: The robot is equipped with sensors such as IMU, cameras, range sensors, GPS, etc. that produce *raw measurements* which, in order to be used, they must be processed. For example, if both a camera and IMU sensor are employed, their raw measurements can be processed through SLAM (Simultaneous Localization and Mapping) to determine the robot's position and orientation. In general, for each raw measurement, the system can choose between D different perception methods to produce a *processed measurement* for the position of the robot through a perception process. Thus, processed measurements are available to be used by the system only at, perhaps non-uniform, instants $\{\tau_k\}_{k=0}^{\infty}$ with $\tau_0 = 0$. Processed measurements are represented by $\mathbf{C}\mathbf{x}[k]$ with some constant matrix $\mathbf{C} \in \mathbb{R}^{n_\mathbf{z} \times n}$ with $(\mathbf{A}, \mathbf{C})$ observable. Each method has a different perception-latency in $\{\Delta^1, \ldots, \Delta^D\}$. Hence, if method $p_k \in \{1, \ldots, D\}$ is chosen at $t = \tau_k$, a new measurement $\mathbf{z}[k] = \mathbf{C}\mathbf{x}[k] + \mathbf{v}[k]$ is available at $t = \tau_k + \Delta^{p_k}$ where $\mathbf{v}[k]$ is a noise modeling the accuracy of the perception method. Note that at $t = \tau_k + \Delta^{p_k}$ the robot is ready to take a new measurement, thus we set $\tau_{k+1} = \tau_k + \Delta^{p_k}$.

The perception-latency and accuracy relation for the perception method is modeled by the covariance matrix $\mathbf{R}^{p_k} = \mathsf{cov}\{\mathbf{v}[k]\}$ which capture the different noise levels of the accuracy of methods $1, \ldots, D$. This corresponds to the actual practical behavior of usual perception algorithms when different configurations are available [45–47, 68]. The values of the latency Δ^{p_k} and noise level $\mathbf{R}^{p_k}$ for a concrete perception method p_k can be obtained offline through a statistical analysis of its performance. In practice, it is expected that a longer latency Δ^{p_k} results in a more concentrated distribution $\mathcal{N}(\mathbf{0}, \mathbf{R}^{p_k})$. For instance, in [68], $\mathbf{R}^{p_k}$ is modeled to be inversely proportional to the latency, which is also supported by the experimental data in [46, 47]. However, to allow our methods to be tailored to any latency-precision model, we do not assume a particular relation between the latency Δ^{p_k} and its corresponding noise level $\mathbf{R}^{p_k}$ but that these might be different between perception methods.

Any (perhaps infinite) sequence of perception modes p is referred to as a *perception schedule*. We refer to a rule that generates a perception schedule p based on state information as a *scheduling policy*.

There are two main characteristics of a schedule p which have a direct impact in the resource usage of the computing platform onboard of the robot, and are independent of the actual trajectories for the state $\mathbf{x}(t)$. These are:

- **Attention:** the attention $\mathsf{att}(p; \mathcal{I})$ of a perception schedule p for an interval of time $\mathcal{I} \subset \mathbb{R}_{\geq 0}$ corresponds to the amount of processed measurements generated by p in

such interval. Keeping this number of sampling events small is desirable both from an energetic point of view and to minimize the use of I/O buses when using the sensors.

- **CPU load:** consider that the latency Δ^{p_k} is not used exclusively for perception, but also that the method frees the computing unit for an interval of length $(1-f^{p_k})\Delta^{p_k}$ with $f^{p_k} \in [0,1]$. Thus, the perception CPU load in the interval $\mathcal{I} \subset \mathbb{R}_{\geq 0}$ is

$$\mathsf{load}(p;\mathcal{I}) = \frac{1}{T_{\mathcal{I}}} \sum_{k=0}^{\alpha-1} f^{p_k} \Delta^{p_k}$$

  where $\alpha = \mathsf{att}(p;\mathcal{I})$ and $T_{\mathcal{I}} = \sup(\mathcal{I}) - \inf(\mathcal{I})$.

Regarding the tracking performance, the goal is to minimize the error $\mathbf{x}(t) - \mathbf{x}_{\mathsf{ref}}(t)$ with respect to some exogenous reference $\mathbf{x}_{\mathsf{ref}}(t)$. For the sake of simplicity, we set $\mathbf{x}_{\mathsf{ref}}(t) = \mathbf{0}$ during this chapter. However, the extension to general time-varying references is usually straightforward by the use of appropriate feedforward terms in the controller $\mathbf{u}(t)$ and by designing over the error system instead. In this sense, one interpretation for (2.1) is that it corresponds to the error dynamics for the robot state, which we want to keep as close to the origin as possible.

Consequently, in this chapter we propose a performance metric that captures both tracking control performance and resource usage, such as CPU load and attention, to be used as a figure of merit for effectively evaluating the proposed perception scheduling methods. Additionally, the chapter aims to propose perception latency scheduling policies with the objective of optimizing the performance metric. The contributions of this chapter were also published in [3, 8].

2.1 RELATED WORK

The perception latency and accuracy trade-off has been studied for state estimation in [68]. In addition, [69] studied another related type of latency trade-off arising from the quality of a communication channel in networked systems. Nonetheless, these approaches are mainly focused on the resulting estimation quality rather than the overall performance of a closed-loop system using such an estimation framework. In [46], a control-estimator co-design is proposed in a periodic setting, where the relationship between latency and estimator quality is modeled and taken into account in a Model Predictive Control (MPC) strategy. Thus, more emphasis is given to the feasibility of an optimization program rather than on ensuring some form of stability outside the sliding window used in the MPC, which is particularly important when switching between different perception modes.

The problem of perception latency scheduling is quite close to the notion of scheduling sampling instants for which some works have used event-triggered and self-triggered sampling [70]. In these sampling schemes, a state-dependent triggering rule for sampling is obtained by employing sufficient stability conditions. In particular, [71] studies variable sampling intervals to model information exchange instants in networked control systems and schedule them accordingly using an event-triggered approach. However, these methods do not use a latency model in which shorter sampling intervals lead to poor state estimates as in the latency-precision trade-off.

Another approach is to model the effect of variable sampling intervals as a time-varying delay [72]. However, these works mostly study robustness against arbitrary sampling sequences. On the other hand, as mentioned before, variable sampling problems can be studied using switching systems theory. Switching systems stability results may be divided into two categories. The first one studies stability under arbitrary switching signals. Examples of this type of analysis can be found in [108, 109] for continuous-time and in [110, 111] for discrete-time. The switching signal can be considered a state-dependent input in the second category. Examples of the study of stability under state-dependant switching signals can be found in [112, 113] for continuous-time systems and in [113–117] for discrete-time systems. However, as we show later, the connection between stability and optimality may not be evident, particularly with the cost function considered in this chapter. To this regard, the works [114, 118] deal with the optimality of state-dependent switching signals for some particular forms of cost functions that do not extend to the cost function modeling the latency-precision trade-off as we propose.

2.2 PROBLEM STATEMENT

In this section, we introduce the formal problem statement for perception latency scheduling. First, in line with the linear model in (2.1), we assume Gaussian probabilistic models for the initial conditions and disturbance process. This is, $\mathbf{x}(0) \sim \mathcal{N}(\mathbf{x}_0, \mathbf{P}_0)$ and $\mathbf{w}(t)$ is a Wiener process with covariance function $\mathbb{E}\{\mathbf{w}(s)\mathbf{w}(r)^\top\} = \mathbf{W}\min(s, r)$ with $\mathbf{W}$ positive semi-definite [119, Page 63].

Due to the the delay introduced by the perception mechanism, the only available information at $t = \tau_k$ is $\{\mathbf{z}[0], \dots, \mathbf{z}[k-1]\}$. Hence, a control is designed as

$$\mathbf{u}[k] = \mathbf{L}^{p_k}\hat{\mathbf{x}}[k|k-1]$$

where

$$\hat{\mathbf{x}}[k|k-1] = \mathbb{E}\left\{\mathbf{x}[k] \;\middle|\; \mathbf{z}[0], \dots, \mathbf{z}[k-1]\right\}$$

is the conditional mean of $\mathbf{x}[k]$ using all available measurements. A recursive expression for $\hat{\mathbf{x}}[k|k-1]$ in the form of a filter is given in Appendix A. Moreover, the gains $\mathbf{L}^1, \dots, \mathbf{L}^D$ are assumed to be given, which may have been designed for each latency by separate, provided that the pair $(\mathbf{A}, \mathbf{B})$ is controllable.

Under this conditions, the goal of this chapter is the following:

Problem 2.1 (Perception scheduling in control)**.** *Let an interval of interest $[0, T_f]$ and penalties $0 \leq r^1, \dots, r^D$, one for each perception method. Thus, given $\lambda_{\mathbf{x}}, \lambda_r, T_f > 0$ and $\mathbf{Q}, \mathbf{Q}_f$ positive semi-definite, the problem is to find a perception schedule $p = \{p_k\}_{k=0}^{\infty}$ such that*

$$\mathcal{J}(p) = \frac{\lambda_r}{T_f}\sum_{k=0}^{\alpha-1} r^{p_k} + \lambda_{\mathbf{x}}\mathbb{E}\left\{\frac{1}{T_f}\int_0^{T_f} \mathbf{x}(t)^\top\mathbf{Q}\mathbf{x}(t)\ \mathrm{d}t + \mathbf{x}(T_f)^\top\mathbf{Q}_f\mathbf{x}(T_f)\right\} \tag{2.2}$$

with $\alpha = \mathsf{att}(p; [0, T_f])$, is minimized for system (2.1). Moreover, if there exists a stabilizing perception schedule for system (2.1) under arbitrary T_f, then p must induce $\lim_{t\to\infty} \mathbb{E}\{\mathbf{x}(t)\} = 0$.

As usual in minimum variance control problems [119, Page 172] the persistent introduction of uncertainty originated from disturbances in the model, and measurement noise prevents a linear controller from making $\mathbf{x}(t)$ converge to the origin in the mean-squared error sense. Hence, to concentrate our efforts on building a schedule p, it is more practical to ensure convergence of $\mathbb{E}\{\mathbf{x}(t)\}$ towards the origin, and deal with the effect of the second order moment of $\mathbf{x}(t)$ during the finite interval of interest $[0, T_f]$ by minimizing (2.2).

The cost in (2.2) is meant to model the latency-precision trade-off we described before in the following way. The second term in (2.2) is composed of an expected quadratic penalty for cost $\mathbf{x}(t)$ of (2.1), which penalizes deviations from the origin. On the other hand, the first term in (2.2) is the accumulation of the penalties $r^{p_0}, r^{p_1}, \dots$ over the interval $[0, T_f]$. If $r^1 = \cdots = r^D = 1$, this term is proportional to the attention of p. On the other hand, using $r^{p_k} = f^{p_k}\Delta^{p_k}$ then the last term corresponds to the CPU load over $[0, T_f]$.

Consider the last requirement in Problem 2.1. Note that the cost in (2.2) only penalizes the schedule during the interval $[0, T_f]$, which can be used to penalize a transient response. In practice, it may be beneficial to pose a similar problem once $[0, T_f]$ has elapsed in a moving horizon fashion. However, this strategy won't ensure asymptotic stability for $\mathbb{E}\{\mathbf{x}(t)\}$ on the long run for arbitrary values of $\lambda_{\mathbf{x}}, \lambda_r, T_f, \mathbf{Q}, \mathbf{Q}_f$ since the individual terms in (2.2) are often conflicting. As an example, take $\lambda_{\mathbf{x}} = 0$ and $\lambda_r > 0$ in which the system's error is not penalized and thus, minimizing (2.2) won't ensure stability regardless of T_f or how often the optimization problem is solved. Hence, a connection between optimality and stability is not evident in general problem setting.

Our strategy is based on the following observation. Using Proposition A.2 in Appendix A, it is obtained that $\bar{\mathbf{x}}[k] := \mathbb{E}\{\mathbf{x}[k]\}$ is given by

$$\bar{\mathbf{x}}[k+1] = \mathbf{\Lambda}(\Delta^{p_k})\bar{\mathbf{x}}[k], \quad \bar{\mathbf{x}}[0] = \bar{\mathbf{x}}_0 \tag{2.3}$$

where

$$\mathbf{\Lambda}(\Delta^{p_k}) := \exp(\mathbf{A}\Delta^{p_k}) + \int_0^{\Delta^{p_k}} \exp(\mathbf{A}\tau)\mathrm{d}\tau\mathbf{B}\mathbf{L}^{p_k}.$$

As a result, (2.3) is a switched system which switches between matrices

$$\mathbf{\Lambda}(\Delta^1), \dots, \mathbf{\Lambda}(\Delta^D)$$

according to the perception schedule p as the switching signal. Hence, the asymptotic stability character of Problem 2.1 is tied to the stability of the switching system in (2.3). In this context, the outline of our solution has 3 main components:

- First, in Section 2.3 we analyze some results in the literature regarding stabilizing switching signals for discrete-time systems. We provide an extension of the results found in the literature to enable the construction of multiple scheduling policy candidates for optimizing (2.2). These scheduling policies are all stabilizing for (2.3) when an admissibility condition is attained.
- Second, for completeness, in Section 2.4 we provide a new algorithm to check if a scheduling policy candidate is admissible.
- Third, in Section 2.5 we provide a new sub-optimal algorithm for Problem 2.1 based on the multiple scheduling policies previously constructed. We analyze its theoretical performance and discuss some heuristics and approximations.

For the sake of readability, all proofs for the formal results are placed at the end of this chapter in Section 2.8.

2.3 STABILITY PRESERVING PERCEPTION SCHEDULES

In this section, we develop scheduling policies which ensure (2.3) is stable. In [115] an interesting framework is proposed which provides sufficient and necessary conditions for asymptotic stability of discrete-time switching linear systems as (2.3), and generates state-dependant switching laws based on such conditions. In the following, we analyze some related ideas and propose a framework suitable for Problem 2.1. This is required since not only a single switching rule is needed, but several switching rule possibilities have to be available as the search space for the optimization of (2.2). First, let us introduce the following concept:

Definition 2.2. *A set of schedules is any set of the form* $\Gamma = \{\gamma^1, \ldots, \gamma^{|\Gamma|}\}$ *with* $|\Gamma| < \infty$ *where* $\gamma^i \in \{1, \ldots, D\}^{\mathsf{len}(\gamma^i)}$ *are individual schedules with* $\mathsf{len}(\gamma^i) < \infty,\ \forall i \in \{1, \ldots, |\Gamma|\}$.

We aim to construct stabilizing perception schedules by piecing together individual finite-length schedules found in a set of schedules. Up to this point, sets of schedules are arbitrary and can be constructed randomly. However, stabilizing switching signals cannot arise for any set of schedules. In the following, we study which sets of schedules induce stability. Consider the following objects: the hyper-ellipse

$$\mathbb{S}_0 := \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^\top \mathbf{M}_0 \mathbf{x} \leq 1\}$$

for some arbitrary positive definite matrix $\mathbf{M}_0$ and

$$\mathbb{S}_\gamma := \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} = \mathbf{\Lambda}(\Delta^{\gamma_0})^{-1} \cdots \mathbf{\Lambda}(\Delta^{\gamma_{\ell-1}})^{-1}\mathbf{y}, \mathbf{y} \in \mathbb{S}_0\}$$

where $\gamma := \{\gamma_k\}_{k=0}^{\ell-1}$ is a schedule of finite length $\mathsf{len}(\gamma) = \ell$. Thus, by construction, for any $\bar{\mathbf{x}}_0 \in \mathbb{S}_\gamma$, then $\bar{\mathbf{x}}[\mathsf{len}(\gamma)] \in \mathbb{S}_0$ for such schedule γ. This is verified as:

$$\begin{aligned}
&\bar{\mathbf{x}}[\mathsf{len}(\gamma)] = \mathbf{\Lambda}^\gamma \bar{\mathbf{x}}_0 \in \mathbf{\Lambda}^\gamma \mathbb{S}_\gamma \\
&= \{\mathbf{z} \in \mathbb{R}^n : \mathbf{z} = \mathbf{\Lambda}^\gamma \mathbf{x},\ \mathbf{x} = (\mathbf{\Lambda}^\gamma)^{-1}\mathbf{y},\ \mathbf{y}^\top \mathbf{M}_0 \mathbf{y} \leq 1\} \\
&= \{\mathbf{z} \in \mathbb{R}^n : \mathbf{z} = \mathbf{\Lambda}^\gamma (\mathbf{\Lambda}^\gamma)^{-1}\mathbf{y},\ \mathbf{y}^\top \mathbf{M}_0 \mathbf{y} \leq 1\} \\
&= \{\mathbf{z} \in \mathbb{R}^n : \mathbf{z}^\top \mathbf{M}_0 \mathbf{z} \leq 1\} = \mathbb{S}_0
\end{aligned}$$

with $\mathbf{\Lambda}^\gamma := \mathbf{\Lambda}(\Delta^{\gamma_{\ell-1}}) \cdots \mathbf{\Lambda}(\Delta^{\gamma_0})$ as the multiplication chain in (2.3) for the schedule γ. Moreover, note that $\mathbb{S}_\gamma$ has the shape of an hyper-ellipse defined by the positive definite matrix $\mathbf{M}_\gamma = (\mathbf{\Lambda}^\gamma)^\top \mathbf{M}_0 (\mathbf{\Lambda}^\gamma)$ as

$$\mathbb{S}_\gamma \equiv \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^\top \mathbf{M}_\gamma \mathbf{x} \leq 1\}.$$

The key idea is the following. Consider Γ to be a set of schedules γ. Thus, if the interior of the set

$$\mathbb{S}_\Gamma^\star := \bigcup_{\gamma \in \Gamma} \mathbb{S}_\gamma$$

contains $\mathbb{S}_0$, it means that for each initial condition, there exists a schedule $\gamma \in \Gamma$ which contracts $\mathbb{S}_\Gamma^\star$ into itself after $\mathsf{len}(\gamma)$ steps. This is, if $\bar{\mathbf{x}}_0 \in \mathbb{S}_\Gamma^\star$, thus

$$\bar{\mathbf{x}}[\mathsf{len}(\gamma)] \in \mathbb{S}_0 \subset \mathsf{int}(\mathbb{S}_\Gamma^\star)$$

for some $\gamma \in \Gamma$.

Definition 2.3 (Admissible set of schedules)**.** *A finite set of schedules Γ is admissible if $\mathbb{S}_0 \subset \mathit{int}(\mathbb{S}_\Gamma^\star)$.*

As we show in a subsequent result, if Γ is admissible, a contracting schedule can be found $\forall \bar{\mathbf{x}} \in \mathbb{R}^n$ as:

$$\gamma^\star(\bar{\mathbf{x}}; \Gamma) := \arg\min_{\gamma \in \Gamma} \{r \in \mathbb{R} : r = \bar{\mathbf{x}}^\top \mathbf{M}_\gamma \bar{\mathbf{x}}\} \tag{2.4}$$

Using this switching rule for various admissible sets of schedules $\Gamma^1, \ldots, \Gamma^m$, the following strategy can be used online to compute a stabilizing perception method at each $t = \tau_k$ given $\bar{\mathbf{x}}[k]$.

Definition 2.4 (Stability Preserving Scheduling)**.** *The SP^2 (Stability Preserving Scheduling Policy) strategy is defined by Algorithm 2.2.*

Remark 2.5. *A similar rule to* (2.4) *was studied in [115] for a particular fixed set of schedules Γ constructed through a combinatorial approach. This is, all possible combinations of elements in $\{1, \ldots, D\}$ up to the first length at which admissibility is attained [115, Algorithm 1]. As a result, a single scheduling policy is considered regardless of the choice of* (2.2)*, and with no additional degrees of freedom to improve performance. Unlike the previous approach, we introduce an extra degree of freedom by constructing multiple sets of schedules $\Gamma^1, \ldots, \Gamma^m$. Then, as explained in detail in Section 2.5, a switching law of the form* (2.4) *is used for each set of schedules as scheduling policy candidates to optimize for* (2.2)*. This makes our approach more beneficial for the perception scheduling problem since intuitively, given an initial condition $\bar{\mathbf{x}}[0]$, the approach with a single set of schedules will result in a fixed schedule p, regardless of the cost* (2.2) *whereas our method can adapt p to the cost.*

Algorithm 2.1 Construction of a set of schedules

Input: ℓ, D.
Output: Γ.

$\Gamma = \varnothing$
repeat
  if $(\{1, \ldots, D\}^\ell \setminus \Gamma = \varnothing)$ **then**
    # If all schedules have been evaluated
    Increase ℓ
  end if
  # Generate a random sequence over $\{1, \ldots, D\}$ with random length $\ell' \leq \ell$.
  $\ell' \leftarrow \texttt{randomSample}(\{1, \ldots, \ell\})$
  $\Gamma \leftarrow \Gamma \bigcup \texttt{randomSample}(\{1, \ldots, D\}^{\ell'} \setminus \Gamma)$
until $\mathbb{S}_0 \subset \mathsf{int}(\mathbb{S}_\Gamma^\star)$ # Using Algorithm 2.3

Algorithm 2.2 SP^2 strategy

Input: $\bar{\mathbf{x}}_0$.
Output: Perception schedule p.

1: $\gamma \leftarrow \varnothing$.
2: $i \leftarrow 0$
3: **for** each instant $\tau_k, k = 0, 1, \dots$ **do**
4: **if** $(i < \mathsf{len}(\gamma))$ **then**
5: $p_k \leftarrow \gamma_i$ #For chosen schedule γ, traverse its elements $\gamma_0, \dots, \gamma_{\mathsf{len}(\gamma)}$.
6: $i \leftarrow i + 1$
7: **else**
8: $\Gamma \leftarrow$ any set of schedules chosen from the admissible options $\Gamma^1, \dots, \Gamma^m$.
9: $\gamma \leftarrow \gamma^\star(\bar{\mathbf{x}}[k]; \Gamma)$ from (2.4). #Once previous schedule has been consumed, choose a new schedule
10: $i \leftarrow 0$
11: **end if**
12: **end for**

In this sense, an alternative for the combinatorial approach is given in Algorithm 2.1 in which random schedules are appended to Γ. These schedules are generated with length up to a given value ℓ, unless all schedules are contained in Γ, in which case ℓ is increased. This procedure terminates once Γ is admissible and, similarly to the combinatorial approach, this must happen if there exists a stabilizing schedule for (2.3). When compared to the combinatorial approach, the advantage of using Algorithm 2.1 is two-fold. First, depending on the initial value of the length ℓ, longer sequences can be tested earlier. This is complicated using the combinatorial approach due to the combinatorial explosion involved in such a strategy. Note that even if the combinatorial approach is used starting from an initial length $\ell > 1$ to test longer sequences earlier, this strategy will neglect short sequences, which may also be useful for rapid decision-making if available. Hence, Algorithm 2.1 covers short and long sequences from the beginning. Moreover, Algorithm 2.1 allows us to generate multiple admissible sets of schedules, namely $\Gamma^1, \dots, \Gamma^m$ useful for our optimization strategy.

However, since in line 8 of Algorithm 2.2 we allow to change the set of schedules Γ to any admissible one between the m options $\{\Gamma^1, \dots, \Gamma^m\}$, the scheduling policy in (2.4) may be different each time in line 9 of Algorithm 2.2 is executed. The following results show that even in this case, Algorithm 2.2 manages to stabilize (2.3).

Theorem 2.6. *Assume that there exists at least one admissible set of schedules and that the SP^2 strategy is used in* (2.3)*. Thus* $\lim_{k\to\infty} \bar{\mathbf{x}}[k] = 0$.

Checking if a set of schedules is admissible is not a trivial task. [115, Remark 6] proposes to check if there is at least one $\gamma \in \Gamma$ such that $\mathbb{S}_0 \subset \mathsf{int}(\mathbb{S}_\gamma)$ as a sufficient condition for admissibility. However, this is a very conservative condition that won't be attained in general, even when Γ is admissible. Motivated by this, in the following section, we provide a novel analysis to check if a set of schedules is admissible.

2.4 NON-CONSERVATIVE ADMISSIBILITY CHECKING

The basis of a non-conservative admissibility checking is formulated as a nonlinear program as follows.

Theorem 2.7. *Let*

$$R := \min_{\bar{\mathbf{x}} \in \partial \mathbb{S}_\Gamma^\star} \bar{\mathbf{x}}^\top \mathbf{M}_0 \bar{\mathbf{x}}. \tag{2.5}$$

Then, $\mathbb{S}_0 \subset \mathbb{S}_\Gamma^\star$ *if and only if* $R > 1$.

Hence, under the light of Theorem 2.7, studying if Γ is admissible is equivalent to solving the nonlinear program in (2.5) and checking if $R > 1$. Note that the attempt to solve numerically for local minima in (2.5) may fail since the condition $R > 1$ is only useful when the global minimum is considered. To the best of our knowledge, there is no previous work which solves for the global minimum of a non-convex program of the form (2.5). Thus, the following efforts are dedicated to this task.

Note that most of the theory presented up to now remains the same even when $\mathbb{S}_0$ is not a hyper-ellipse. An arbitrary $C^\star$ set as in Definition B.1 in Appendix B can be used instead of a hyper-ellipse. For example, polyhedral sets are also considered in [115]. However, note that the set $\partial \mathbb{S}_\Gamma^\star$ may be non-convex, implying that there might be many local minima for (2.5), which complicates the analysis. Despite this, we show that for the case where $\mathbb{S}_0$ is a hyper-ellipse, we can compute the global minimum of (2.5). The strategy is the following: we split the nonlinear program in (2.5) into subprograms of the following form:

$$\begin{aligned} &\min_{\bar{\mathbf{x}} \in \mathbb{R}^n} \bar{\mathbf{x}}^\top \mathbf{M}_0 \bar{\mathbf{x}} \\ &\text{s.t. : } h_\gamma(\bar{\mathbf{x}}) := \bar{\mathbf{x}}^\top \mathbf{M}_\gamma \bar{\mathbf{x}} - 1 = 0, \quad \forall \gamma \in \Gamma' \in \mathcal{P}(\Gamma) \end{aligned} \tag{2.6}$$

Where Γ' is any subset of Γ, i.e. Γ' is in the power set $\mathcal{P}(\Gamma)$. Intuitively, the sub-program in (2.6) is useful since due to the shape of $\partial \mathbb{S}_\Gamma^\star$, the global optimum of (2.5) must lie either in the boundary of a single hyper ellipse $\mathbb{S}_\gamma$ for some $\gamma \in \Gamma$ which has the form $h_\gamma(\bar{\mathbf{x}}) = 0$, or in the intersection of the boundaries of multiple hyper-ellipses. As an example, consider $\Gamma = \{p, q\}$ with arbitrary schedules p, q. Hence, the global optimum of (2.5) must be the global optimum of (2.6) with either $\Gamma' = \{p\}$, $\Gamma' = \{q\}$, which correspond to checking points in $\partial \mathbb{S}_p, \partial \mathbb{S}_q$ by separate or $\Gamma' = \{p, q\}$ which corresponds to checking points in $\partial \mathbb{S}_p \cap \partial \mathbb{S}_q$ where the two constraints $h_p(\bar{\mathbf{x}}) = 0, h_q(\bar{\mathbf{x}}) = 0$ are active. Therefore, the solution for (2.5) comes from (2.6) for some $\Gamma' \in \mathcal{P}(\Gamma) = \{\varnothing, \{p\}, \{q\}, \{p, q\}\}$. This idea is formalized in the following:

Corollary 2.8. *Let* $\mathcal{X}_{\Gamma'}$ *be the set of all critical points (local minima candidates)* $\bar{\mathbf{x}}^*$ *of* (2.6) *such that* $\bar{\mathbf{x}}^* \notin \mathbb{S}_\gamma$ *for any* $\gamma \in \Gamma \setminus \Gamma'$ *and any* $\Gamma' \in \mathcal{P}(\Gamma)$. *Thus, the global minimum* R *of* (2.5) *can be obtained as*

$$R = \min \left\{ r = \bar{\mathbf{x}}^\top \mathbf{M}_0 \bar{\mathbf{x}} : \bar{\mathbf{x}} \in \bigcup_{\Gamma' \in \mathcal{P}(\Gamma)} \mathcal{X}_{\Gamma'} \right\} \tag{2.7}$$

Corollary 2.8 implies that the solution of (2.5) can be obtained by solving programs of the form (2.6), where critical points of (2.6) are rejected if such points are contained in other hyper-ellipses not involved in (2.6) (written as $\bar{\mathbf{x}}^* \notin \mathbb{S}_\gamma, \forall \gamma \in \Gamma \setminus \Gamma'$ in Corollary

2.8). Moreover, all programs of the form (2.6) involve from one hyper-ellipse at the time, to all combinations of $\mathbb{S}_\gamma, \forall \gamma \in \Gamma$.

Remark 2.9. *Note that if $|\Gamma'| = n$ in (2.6), the set of all points $\bar{\mathbf{x}}$ with $\bar{\mathbf{x}}^\top \mathbf{M}_\gamma \bar{\mathbf{x}} = 1, \forall \gamma \in \Gamma'$ contains only isolated points for almost all $\{\mathbf{M}_\gamma\}_{\gamma \in \Gamma}$. Moreover, the set of equations $\bar{\mathbf{x}}^\top \mathbf{M}_\gamma \bar{\mathbf{x}} = 1$ are n polynomial equations with n variables which have n^n complex solutions due to the Bezout's Theorem [120, Theorem 4.14]. Homotopy solvers such as the one in [121] can track all such real solutions for polynomial systems or obtained explicitly if $n \leq 2$. Hence, all reals solutions can be checked directly to build the set $\mathcal{X}_{\Gamma'}$. In contrast, the case with $|\Gamma'| > n$ can be ignored since those solutions are either contained in the $|\Gamma'| = n$ case or are nonexistent.*

In general, an optimization program of the form (2.6) can be solved by the Lagrange multipliers method. However, depending on the problem constraints, in this case, given in the form $h_\gamma(\bar{\mathbf{x}}) = 0$, not all local minima can be found through this method. Such local minima are often called non-regular [122, Page 279]. Due to the fact that we look for the global optimum of (2.6), we are forced to analyze both regular and non-regular points. Formally, critical points in $\mathcal{X}_{\Gamma'}$ for (2.6) in which $|\Gamma'| < n$ will be classified into regular or non-regular points for their analysis as follows:

Definition 2.10. *[122, Page 279] A critical point $\bar{\mathbf{x}}$ for (2.6) is said to be regular if the vectors*

$$\nabla h_\gamma(\bar{\mathbf{x}}) = 2\mathbf{M}_\gamma \bar{\mathbf{x}}, \quad \gamma \in \Gamma'$$

are linearly independent.

In the next section, we use the Lagrange multiplier method to find the list of critical regular points for (2.6). Then, we use a tailored analysis for the non-regular ones.

2.4.1 Regular points analysis

We start the study of (2.6) with the characterization of regular points by means of the Lagrange multiplier theorem in [122, Proposition 3.1.1].

Theorem 2.11. *Let $\boldsymbol{\lambda} := \{\lambda_\gamma\}_{\gamma \in \Gamma'}$ be a solution to the system of equations*

$$\operatorname{tr}(\mathbf{G}(\boldsymbol{\lambda})^\dagger (\mathbf{M}_\gamma - \mathbf{M}_\nu)) = 0, \quad \gamma, \nu \in \Gamma' \tag{2.8a}$$

$$\det \mathbf{G}(\boldsymbol{\lambda}) = 0 \tag{2.8b}$$

where

$$\mathbf{G}(\boldsymbol{\lambda}) := \mathbf{M}_0 + \sum_{\gamma \in \Gamma'} \lambda_\gamma \mathbf{M}_\gamma \tag{2.9}$$

and $\mathbf{G}(\boldsymbol{\lambda})^\dagger$ is the adjugate matrix of $\mathbf{G}(\boldsymbol{\lambda})$ [123, Page 22]. Thus, all regular critical points $\bar{\mathbf{x}}^*$ of program (2.6) are in the kernel of $\mathbf{G}(\boldsymbol{\lambda})$ for some $\boldsymbol{\lambda}$. This is, there exists a solution $\boldsymbol{\lambda}$ of (2.8) such that $\mathbf{G}(\boldsymbol{\lambda})\bar{\mathbf{x}}^* = 0$ and have

$$(\bar{\mathbf{x}}^*)^\top \mathbf{M}_0 (\bar{\mathbf{x}}^*) = -\sum_{\gamma \in \Gamma'} \lambda_\gamma.$$

Remark 2.12. *Note that the number of equations in* (2.8) *are evidently more than* $|\Gamma'|$. *However, most of them are linearly dependent. They can be arranged as* $|\Gamma'|$ *independent equations with* $|\Gamma'|$ *variables by using* $|\Gamma'|-1$ *equations of the form* (2.8a) *in addition to* (2.8b). *These* $|\Gamma'|$ *equations are independent for most choices of* $\{\mathbf{M}_\gamma\}_{\gamma\in\Gamma}$. *Moreover, note that all equations in* (2.8) *are polynomials in* $\boldsymbol{\lambda}$ *and therefore, the number of solutions can be counted by means of Bezout's theorem and obtained similarly as pointed out in Remark 2.9 using the algorithm in [121].*

Moreover, the nullity (dimension of kernel) of the singular matrix $\mathbf{G}(\boldsymbol{\lambda})$ may be of dimension up to n. However, among the possible singular matrices, ones with nullity greater than one lie in a set of measure zero (with respect to the possible choices of $\{\mathbf{M}_\gamma\}_{\gamma\in\Gamma}$). This fact is evidenced in the following result.

Corollary 2.13. *Let* $\boldsymbol{\lambda}$ *satisfy* (2.8), *assume that* $\mathbf{G}(\boldsymbol{\lambda})$ *has nullity greater than 1, and let* $\mathbf{G}_{ij}(\boldsymbol{\lambda})$ *be the sub-matrix obtained by deleting the* i*-th row and* j*-th column of* $\mathbf{G}(\boldsymbol{\lambda})$. *Then,* $\boldsymbol{\lambda}$ *complies with*

$$\det\, \mathbf{G}_{ij}(\boldsymbol{\lambda}) = 0, \quad \forall (i,j) \in \{1,\dots n\}^2 \tag{2.10}$$

As a result of Corollary 2.13, the case in which $\mathbf{G}(\boldsymbol{\lambda})$ has nullity greater than one is is not common in a general setting since $\boldsymbol{\lambda} \in \mathbb{R}^{|\Gamma'|}$ would have to comply $|\Gamma'|-1$ equations of the type (2.8a) in addition to n^2 equations of the type (2.10). Henceforth, we ignore such cases in our analysis. Now, for $\mathbf{G}(\boldsymbol{\lambda})$ with nullity 1, let $\mathbf{g}$ be any vector in the kernel of $\mathbf{G}(\boldsymbol{\lambda})$. Thus, as a result of Theorem 2.11 the critical regular points for such $\boldsymbol{\lambda}$ are parallel to $\mathbf{g}$, and can be uniquely computed by scaling $\mathbf{g}$ in order to comply $(\bar{\mathbf{x}}^*)^\top \mathbf{M}_0 \bar{\mathbf{x}}^* = -\sum_{\gamma\in\Gamma'} \lambda_\gamma$ as:

$$\bar{\mathbf{x}}^* = \pm \mathbf{g} \sqrt{\frac{-\sum_{\gamma\in\Gamma'} \lambda_\gamma}{\mathbf{g}^\top \mathbf{M}_0 \mathbf{g}}} \tag{2.11}$$

and therefore, the condition $\bar{\mathbf{x}}^* \in \mathbb{S}_\gamma$ for $\gamma \in \Gamma \setminus \Gamma'$ can be checked by computing if $(\bar{\mathbf{x}}^*)^\top \mathbf{M}_\gamma (\bar{\mathbf{x}}^*) > 1$ in order to build the set $\mathcal{X}_{\Gamma'}$ as required in Corollary 2.8.

2.4.2 Non-regular points analysis

Non-regular points for program (2.6) are points $\bar{\mathbf{x}}$ which comply $\bar{\mathbf{x}}^\top \mathbf{M}_\gamma \bar{\mathbf{x}} = 1, \forall \gamma \in \Gamma'$ and the vectors $\{\mathbf{M}_\gamma \bar{\mathbf{x}}\}_{\gamma\in\Gamma'}$ linearly dependent. In general, linearly dependent vectors are rare among all possible vectors, and hence is not surprising that non-regular points won't exist in arbitrary settings. In the following result, we characterize when they exist by obtaining what additional concrete conditions must be satisfied.

Proposition 2.14. *Let* $\mathbf{Y}(\bar{\mathbf{x}})$ *an* $n \times |\Gamma'|$ *matrix whose columns are the vectors* $\mathbf{M}_\gamma \bar{\mathbf{x}}, \forall \gamma \in \Gamma'$ *and let* $W_\alpha(\bar{\mathbf{x}})$ *be the* $|\Gamma'| \times |\Gamma'|$ *sub-matrix of* $\mathbf{Y}(\bar{\mathbf{x}})$ *obtained by deleting* $n - |\Gamma'|$ *rows indexed in* $\alpha \in \mathcal{A} \subset \{1,\dots n\}^{n-|\Gamma'|}$ *with* $\mathcal{A}$ *containing all elements in* $\{1,\dots n\}^{n-|\Gamma'|}$ *with non repeating entries. Therefore, any non-regular point for program* (2.6) *must comply with*

$$\bar{\mathbf{x}}^\top \mathbf{M}_\gamma \bar{\mathbf{x}} = 1, \quad \forall \gamma \in \Gamma' \tag{2.12a}$$

$$\det\, \mathbf{Y}_\alpha(\bar{\mathbf{x}}) = 0, \quad \forall \alpha \in \mathcal{A} \tag{2.12b}$$

Note that the amount of equations of the type (2.12b) are equivalent to the number of sub-matrices of size $|\Gamma'| \times |\Gamma'|$ inside $\mathbf{Y}(\bar{\mathbf{x}})$ which is exactly [123, Page 21]

$$\frac{n!}{|\Gamma'|!(n-|\Gamma'|)!}.$$

Note that there are at least $n - |\Gamma'| + 1$ independent equations of this type since that correspond to the number of $|\Gamma'| \times |\Gamma'|$ sub-matrices of $\mathbf{Y}(\bar{\mathbf{x}})$ with consecutive rows. Hence, the number of equations in (2.12) are $|\Gamma'|$ of the type (2.12a) and at least $n - |\Gamma'| + 1$ of type (2.12b). Equivalently, a non-regular point $\bar{\mathbf{x}} \in \mathbb{R}^n$ would have to comply with at least $n + 1$ polynomial equations. One could solve n equations picked from (2.12) and check if the results comply with the remaining equations. However, due to this over-determined feature of equations in (2.12) non-regular points won't exist in general.

2.4.3 Admissibility checking algorithm

Using the results in the previous sections, we summarize the methodology for admissibility checking in the following. Consider Algorithm 2.3 which takes as an input the dimension of the state space n, the matrix $\mathbf{M}_0$, the set of matrices $M := \{\mathbf{M}_\gamma\}_{\gamma\in\Gamma}$ and the set of schedules Γ. This function builds a set $\mathcal{X}$ of all global minima candidates of (2.5) and checks $\min \mathcal{X} > 1$ as this condition is equivalent to admissibility due to Theorem 2.7. In line 2, all subsets of schedules in Γ are checked, ranging from no schedules at all, one schedule at the time, pairs of schedules, and so on. In line 3 we check if the constraint space is of dimension n, since in such case the constraint space is comprised of just isolated points, and we obtain critical points as in Remark 2.9 by calling `IsolatedSolutions`. Otherwise, we obtain regular values by calling `RegularSolutions`. Cases in which the constraint space is of dimension more than n are ignored. Algorithm 2.4 obtains critical objective values when the constraint space contains only isolated points. This algorithm looks for real solutions of the system of n equations and n variables $\{\bar{\mathbf{x}}^\top \mathbf{M}_\gamma \bar{\mathbf{x}} = 1, \forall \gamma \in \Gamma\}$ in line 1. Then, for all solutions we check one by one if they are not contained in other hyper-ellipses for $\gamma \in \Gamma \setminus \Gamma'$. If any of those conditions are complied we reject such points in line 3 in order to build the set $\mathcal{X}_{\Gamma'}$ required in Corollary 2.8. Algorithm 2.5 looks for regular points of (2.5). We build a set of critical objective values in $\mathcal{X}$ and then return $r \leftarrow \min \mathcal{X}$. First, we solve the system of equations in (2.8). Then, for all real solutions $\boldsymbol{\lambda}$ of such polynomial equations we compute $\mathbf{G}(\boldsymbol{\lambda})$ and the critical point as in (2.11). These points are rejected if they are contained in other hyper-ellipses $\Gamma \setminus \Gamma'$ in line 7. Then, the critical objective is given by $-\sum_{\gamma\in\Gamma'} \lambda_\gamma$ as obtained in Theorem 2.11.

Remark 2.15. *Note that in Algorithm 2.3 we explicitly assumed that* $\mathbf{G}(\boldsymbol{\lambda})$ *has nullity 1 and that there are no non-regular points. This assumption is reasonable since the system of equations made by* (2.8a) *in addition to* (2.10) *and the system of equations in* (2.12) *are over-determined for almost any* $\{\mathbf{M}_\gamma\}_{\gamma\in\Gamma}$.

Remark 2.16. *Note that due to line 2 in Algorithm 2.3, the complexity of the admissibility checking procedure grows as* $|\mathcal{P}(\Gamma)| = 2^{|\Gamma|}$. *Still, this procedure can be stopped as soon as some value contained* $\mathcal{X}$ *is less than 1. Moreover, recall that checking for admissibility is not required online for the* SP^2 *strategy in Algorithm 2.2. Instead, this procedure is used to construct admissible sets of schedules in Algorithm 2.1 offline.*

One important input for Algorithm 2.3 is the matrix $\mathbf{M}_0$ which acts as an hyper-parameter for the overall system. Note that admissibility of the set of schedules Γ only depends on some arbitrary choice of $\mathbf{M}_0$ as long as it is positive definite. Hence, setting $\mathbf{M}_0 = \mathbf{I}$ can be considered a systematic default option. On the other hand, the choice of $\mathbf{M}_0$ may affect the time until admissibility is concluded in Algorithm 2.3. However, optimizing $\mathbf{M}_0$ to reduce such time is considered out of the scope of this chapter but could be explored in future research.

2.5 TOWARDS OPTIMAL SCHEDULING

Up to this point, the analysis has been devoted to the stability aspect of Problem 2.1. In this section, we focus on the optimality aspect of the problem based on the stability results obtained so far. First, we study the inherent complexity of Problem 2.1. Due to the possible conflicting objectives in (2.2), the most general setting of Problem 2.1 is reduced to a combinatorial search between the possible scheduling decisions. This makes it hard to expect that exact and efficient solutions for Problem 2.1 exist except for very particular cases. This issue is formalized in the following result.

Proposition 2.17. *Problem 2.1 is NP-hard.*

Henceforth, our approach from this point will be to obtain approximate solutions to Problem 2.1 and study their performance with respect to the optimal solution. Consider $m > 0$ sets of schedules $\Gamma^1, \dots, \Gamma^m$ constructed using Algorithm 2.1, leading to m scheduling policies. Theorem 2.6 ensures that it is possible to choose between any of these scheduling policies without compromising asymptotic stability. Hence, instead of deciding between different perception configurations, a sub-optimal schedule for Problem 2.1 will be constructed by an appropriate decision sequence between the m scheduling policies induced by $\Gamma^1, \dots, \Gamma^m$.

Using the dynamic programming framework [124], Algorithm 2.6 aims to obtain an optimal schedule for Problem 2.1 when the schedule is constrained to be constructed using the scheduling policies induced by $\Gamma^1, \dots, \Gamma^m$. Note that this algorithm proceeds to evaluate all scheduling policies for the current state $\bar{\mathbf{x}}_0$ as in line 3. Then, proceeds to compute $\bar{\mathbf{x}}(t), \mathbf{P}(t)$ in line 7 for $t \in [\tau, \min(\tau^+, T_f)]$ which is the interval in which the schedule piece γ is active before $t = T_f$, as well as the cost for such interval in line 8. Note that $\bar{\mathbf{x}}(t), \mathbf{P}(t)$ and the cost can be computed explicitly as a result of Proposition A.2 in Appendix A. Then, $\tau^+ < T_f$ it means that γ is not sufficiently long to fill the entire window $[0, T_f]$. Hence, the remaining cost to go J^+ and schedule p^+ optimal for $t \in [\tau^+, T_f]$ is obtained by a recursive call in line 10. Otherwise, the terminal cost is added in line 13. Once all m options have been evaluated, `dynprog` returns the one with the best cost as well as the optimal schedule as in line 16, where $\oplus$ means concatenation of schedules. The optimal properties of this algorithm are detailed in the following.

Proposition 2.18. *Calling* $\texttt{dynprog}(T_f, 0, \bar{\mathbf{x}}_0, \mathbf{P}_0)$ *obtains the optimal schedule and cost* p^*, J^* *for Problem 2.1 for the case when the schedule is constrained to be constructed using the scheduling policies for* $\{\Gamma^1, \dots, \Gamma^m\}$. *Moreover, the worst case complexity of Algorithm 2.6 is* $O\left(m^{\lfloor T_f/c \rfloor}\right)$ *where*

$$c = \min \left\{ \sum_{k=0}^{\mathsf{len}(\gamma)-1} \Delta^{\gamma_k} : \gamma \in \Gamma^1 \cup \cdots \cup \Gamma^m \right\}.$$

Algorithm 2.3 `Admissibility`

Input: $n, \mathbf{M}_0, \{\mathbf{M}_\gamma\}_{\gamma\in\Gamma}, \Gamma$.
Output: $\big(\mathbb{S}_0 \subset \mathsf{int}(\mathbb{S}^\star_\Gamma)\big)$?

Set $\mathcal{X} \leftarrow \varnothing$.
for $\Gamma' \in \mathcal{P}(\Gamma)$ **do**
  if $|\Gamma'| = n$ **then**
    $\mathcal{X} \leftarrow \mathcal{X} \cup \{$ `IsolatedSolutions`$(n, \mathbf{M}_0, \{\mathbf{M}_\gamma\}_{\gamma\in\Gamma}, \Gamma, \Gamma')\}$
  else
    if $|\Gamma'| < n$ **then**
      $\mathcal{X} \leftarrow \mathcal{X} \cup \{$ `RegularSolutions`$(n, \mathbf{M}_0, \{\mathbf{M}_\gamma\}_{\gamma\in\Gamma}, \Gamma, \Gamma')\}$
    end if
  end if
end for
return $\min \mathcal{X} > 1$

Algorithm 2.4 `IsolatedSolutions`

Input: $n, \mathbf{M}_0, \{\mathbf{M}_\gamma\}_{\gamma\in\Gamma}, \Gamma, \Gamma'$.
Output: r.

$\mathcal{X}_{\Gamma'} \leftarrow$ real solutions of $\{\bar{\mathbf{x}}^\top \mathbf{M}_\gamma \bar{\mathbf{x}} = 1, \forall \gamma \in \Gamma'\}$ as in Remark 2.9.
for $\bar{\mathbf{x}} \in \mathcal{X}_{\Gamma'}$ **do**
  if exists $\gamma \in \Gamma \setminus \Gamma'$ such that $\bar{\mathbf{x}}^\top \mathbf{M}_\gamma \bar{\mathbf{x}} < 1$ **then**
    $\mathcal{X}_{\Gamma'} \leftarrow \mathcal{X}_{\Gamma'} \setminus \{\bar{\mathbf{x}}\}$.
  end if
end for
return $r \leftarrow \min\{\bar{\mathbf{x}}^\top \mathbf{M}_0 \bar{\mathbf{x}} : \bar{\mathbf{x}} \in \mathcal{X}_{\Gamma'}\}$

Algorithm 2.5 `RegularSolutions`

Input: $n, \mathbf{M}_0, \{\mathbf{M}_\gamma\}_{\gamma\in\Gamma}, \Gamma, \Gamma'$.
Output: r.

Set $Y \leftarrow$ real solutions of (2.8) as in Remark 2.12
$\mathcal{X} \leftarrow \varnothing$
for $\boldsymbol{\lambda} \in Y$ **do**
  $\mathbf{G}(\boldsymbol{\lambda}) \leftarrow \mathbf{M}_0 + \sum_{\gamma\in\Gamma'} \lambda_\gamma \mathbf{M}_\gamma$.
  $\mathbf{g} \leftarrow$ any element of $\mathsf{ker}(\mathbf{G}(\boldsymbol{\lambda}))$.
  $\bar{\mathbf{x}} \leftarrow \mathbf{g}\sqrt{\dfrac{-\sum_{\gamma\in\Gamma'} \lambda_\gamma}{\mathbf{g}^\top \mathbf{M}_0 \mathbf{g}}}$.
  if $\bar{\mathbf{x}}^\top \mathbf{M}_\gamma \bar{\mathbf{x}} > 1, \forall \gamma \in \Gamma \setminus \Gamma'$ **then**
    $\mathcal{X} \leftarrow \mathcal{X} \cup \left\{ -\sum_{\gamma\in\Gamma'} \lambda_\gamma \right\}$
  end if
end for
return $r \leftarrow \min \mathcal{X}$.

Remark 2.19. *Due to NP-hardness of Problem 2.1, it is not surprising that the dynamic programming approach for this problem leads to an algorithm that does not have a polynomial complexity in T_f. However, these types of algorithms, in which unrolling the recursive calls lead to an exponentially growing tree as in Algorithm 2.6, are widely studied. Hence, some performance improvements in terms of complexity and run-time reduction can be made to Algorithm 2.6 such as applying a branch-and-bound technique [124, Chapter 2.3.3]. In addition, heuristics and approximate solutions such as limited look-ahead policies, roll-out algorithms, among others [124, Chapter 6] can be applied in practice. For instance, a balanced* $\texttt{SP}^2$ *strategy can be applied where the set of schedules* Γ *in line 8 of Algorithm 2.2 is selected by calling*

$$\{-,-,\Gamma\} = \texttt{dynprog}(T, 0, \hat{\mathbf{x}}[k|k-1], \hat{\mathbf{P}}[k])$$

as a moving horizon strategy with a look-ahead window of size T.

Nevertheless, all heuristics and approximate approaches mentioned before will have a relative performance loss with respect to the optimal schedule for Problem 2.1 only when the solution is constrained to be constructed using the scheduling policies induced by $\Gamma^1, \ldots, \Gamma^m$, and not necessarily the real optimum of Problem 2.1. Despite this, in the following result, we show that the solution of Algorithm 2.6 can approximate the optimal performance for sufficiently large number of sets of schedules m, increasing the computational power applied.

Theorem 2.20. *Let p^* be the optimal cost for the general setting of Problem 2.1 and assume that* (2.3) *is stabilizable. Then, for any compact sets $\mathcal{B}_{\bar{\mathbf{x}}} \subset \mathbb{R}^n, \mathcal{B}_{\mathbf{P}} \subset \mathbb{R}^{n\times n}$ and any $\varepsilon > 0$ there exists $\Gamma^1, \ldots, \Gamma^m$ with $m < \infty$ such that*

$$|\mathcal{J}(p^*) - \mathcal{J}(p)| \leq \varepsilon$$

with p from by Algorithm 2.6, for any $\bar{\mathbf{x}}_0 \in \mathcal{B}_{\bar{\mathbf{x}}}$, $\mathbf{P}_0 \in \mathcal{B}_{\mathbf{P}}$.

2.6 SIMULATION EXAMPLES

2.6.1 Double integrator

As a first example consider system (2.1) with

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ \mathbf{C} = [1, 0], \ \mathbf{W} = 1$$

as well as $D = 2$ with perception latencies $\Delta^1 = 0.01, \Delta^2 = 0.1$ with their corresponding covariances $\mathbf{R}^1 = 0.5, \mathbf{R}^2 = 0.01$. In addition, consider

$$\mathbf{L}^1 = \mathbf{L}^2 = [-1.5 - 3], \mathbf{M}_0 = \begin{bmatrix} 3.53 & -1.10 \\ -1.10 & 1.36 \end{bmatrix}$$

where $\mathbf{M}_0$ was chosen as an arbitrarily positive definite matrix. To depict admissibility, Algorithm 2.1 was used with $\ell = 20$ to generate a set of schedules Γ and Algorithm 2.3 was used to check admissibility. Note that in Algorithm 2.3 the only meaningful subsets

Algorithm 2.6 `dynprog`

Input: $T_f, \tau, \bar{\mathbf{x}}_0, \mathbf{P}_0$.
Output: p^*, J^*, Γ^*.

$J^* \leftarrow \infty$.
$\Gamma^* \leftarrow \varnothing$
for Γ in $\{\Gamma^1, \ldots, \Gamma^m\}$ **do**
  $\gamma \leftarrow \gamma^\star(\bar{\mathbf{x}}_0; \Gamma)$ from (2.4).
  $\tau^+ \leftarrow \tau + \sum_{k=0}^{\mathsf{len}(\gamma)-1} \Delta^{\gamma_k}$
  $\alpha \leftarrow \mathsf{att}(\gamma, [\tau, \min(\tau^+, T_f)])$
  Compute $\bar{\mathbf{x}}(t), \mathbf{P}(t)$ for schedule γ using $\bar{\mathbf{x}}(\tau) = \bar{\mathbf{x}}_0, \mathbf{P}(\tau) = \mathbf{P}_0$ on (A.4) for $t \in [\tau, \min(\tau^+, T_f)]$
  $J \leftarrow \frac{\lambda_{\mathbf{x}}}{T_f} \int_\tau^{\min(\tau^+, T_f)} \bar{\mathbf{x}}(t)^\top \mathbf{Q} \bar{\mathbf{x}}(t) \; + \mathsf{tr}(\mathbf{Q}\mathbf{P}(t)) \mathrm{d}t + \frac{\lambda_r}{T_f} \sum_{k=0}^{\alpha-1} r^{\gamma_k}$
  if $(\tau^+ < T_f)$ **then**
    $\{p^+, J^+, \Gamma^+\} \leftarrow \mathtt{dynprog}(T_f, \tau^+, \bar{\mathbf{x}}(\tau^+), \mathbf{P}(\tau^+))$
    $J \leftarrow J + J^+$
  else
    $J \leftarrow J + \lambda_{\mathbf{x}} \left(\bar{\mathbf{x}}(T_f)^\top \mathbf{Q}_f \bar{\mathbf{x}}(T_f) + \mathsf{tr}(\mathbf{Q}\mathbf{P}(T_f)) \right)$
    $p^+ \leftarrow \varnothing$
  end if
  if $J \leq J^*$ **then**
    $J^* \leftarrow J$
    $p^* \leftarrow \gamma \oplus p^+$
    $\Gamma^* \leftarrow \Gamma$
  end if
end for

of schedules Γ' in the power set $\mathcal{P}(\Gamma)$ are Γ' containing either a single schedule, or pairs of schedules as a result of the discussion in Remark 2.9. Moreover, note that in the case of Γ' containing a single schedule γ, (2.9) results in $\mathbf{G}(\boldsymbol{\lambda}) = \mathbf{M}_0 + \lambda \mathbf{M}_\gamma$ and therefore possible solutions of Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}$ are eigenvalues of $\mathbf{M}_0^{-1}\mathbf{M}_\gamma$. The resulting nullity of $\mathbf{G}(\boldsymbol{\lambda})$ for any Γ' in this experiment was 1 and hence regular critical points were computed as in (2.11). In this case, there are no non-regular points, since for $|\Gamma'| = 1$ there is a single vector $\nabla h_\gamma(\bar{\mathbf{x}})$. For the case of $|\Gamma'| = n = 2$ critical points were obtained by solving a polynomial system of equations given by the intersection of the two resulting ellipses as in Remark 2.9. Hence, in this setting the admissibility value was obtained to be $R = 1.142$ which indicates that Γ is admissible. Figure 2.2 shows $\mathbb{S}_\Gamma^\star$ and $\mathbb{S}_0$ for this example where it shown that $\mathbb{S}_0 \subset \mathsf{int}(\mathbb{S}_\Gamma^\star)$. Moreover, note that none of the individual sets $\mathbb{S}_\gamma, \gamma \in \Gamma$ manage to cover $\mathbb{S}_0$ by separate.

Now, we show the performance of an heuristics implementation of the strategy presented in Section 2.5. To do so, we simulated system (2.1) using explicit Euler-Maruyama method with time step $h = 10^{-5}$ over the interval $[0, T_f = 100]$ for each experiment. A balanced SP^2 strategy as described in Remark 2.19 was used with limited look-ahead

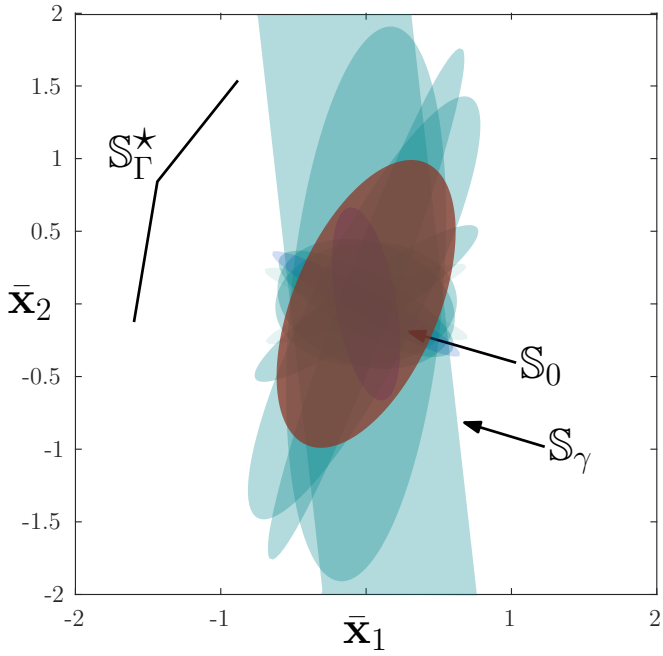

Figure 2.2: Set $\mathbb{S}_\Gamma^\star$ for the set of schedules Γ described in Section 2.6.1 as well as $\mathbb{S}_0$ to show admissibility.

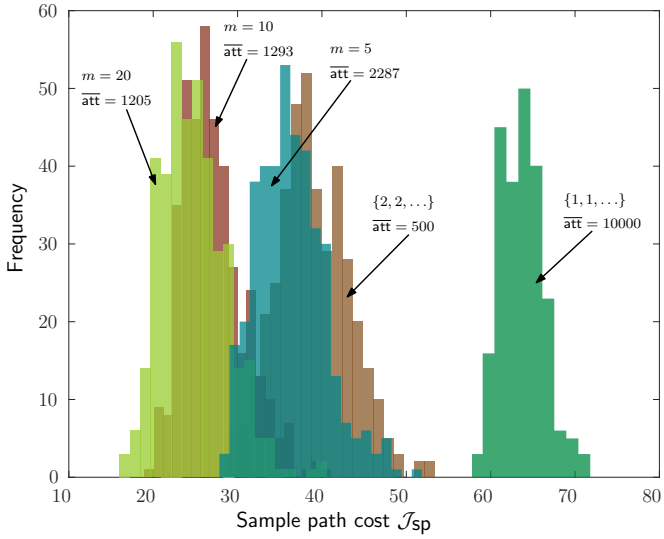

Figure 2.3: Resulting histograms for the sample path cost $\mathcal{J}_{\mathsf{sp}}$ for example of Section 2.6.1 using the balanced SP^2 strategy with $m = 5, m = 10, m = 20$ and static schedules $\{1, 1, \dots\}, \{2, 2, \dots\}$. Average attention $\overline{\mathsf{att}}$ is also shown.

window of size $T = 2$. The performance is evaluated using the cost (2.2) for each sample path with $\mathbf{Q} = \mathbf{Q}_f = \mathsf{diag}([2, 1]), \lambda_{\mathbf{x}} = 1,\ \lambda_r = 0.05$. Moreover, $r^1 = r^2 = 1$ such that the number of sampling events is penalized in the cost. Furthermore, 400 sample paths where generated with $\bar{\mathbf{x}}_0 = [1, 1], \mathbf{P}_0 = \mathbf{I}$ for $m = 5, m = 10, m = 20$ respectively. For comparison, the same amount of sample paths where obtained for schedules $\{1, 1, \dots\}, \{2, 2, \dots\}$ respectively to study the performance of the classical approach of maintaining the same perception configuration during the whole experiment. The resulting histograms for the sample path cost $\mathcal{J}_{\mathsf{sp}}$ for the whole interval $[0, T_f]$ are shown in Figure 2.3 for each case. It can be observed that the average cost of our approach is reduced when compared to the static approaches. Moreover, the results suggest that increasing the number of sets of schedules m improves the cost $\mathcal{J}_{\mathsf{sp}}$ and average attention $\overline{\mathsf{att}}$.

2.6.2 Particle mobile robot

Now consider a system with state $\mathbf{x} = [\mathbf{x}_1, \dots, \mathbf{x}_6]^\top$, and dynamics given

$$\dot{\mathbf{x}}_1 = \mathbf{x}_2, \quad \dot{\mathbf{x}}_3 = \mathbf{x}_4, \quad \dot{\mathbf{x}}_5 = \mathbf{x}_6$$

as well as

$$\dot{\mathbf{x}}_2 = \mathbf{u}_1/\mu + \mathbf{w}_1, \quad \dot{\mathbf{x}}_4 = \mathbf{u}_2/\mu + \mathbf{w}_2, \quad \dot{\mathbf{x}}_6 = \mathbf{u}_3/\mu + \mathbf{w}_3$$

with $\mu = 0.1$. This system corresponds to the model of a particle mobile robot with mass μ. This kind of system has been useful as a model for aerial vehicles such as multi-rotors in [18, 125, 126] were $[\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_5]^\top$ is the position of the multi-rotor and $[\mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_6]^\top$ is the velocity. In fact, a common strategy for the multi-rotor control is adopt a hierarchical control approach, where control signals $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$ are designed exclusively for position as a high-level controller. In a second stage, the resulting position controls are used to construct a low level controller for the attitude of the multi-rotor. Moreover, measurements for position are taken as $\mathbf{C}\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_5]^\top$ using visual perception [18]. In this example we adopt the noise model from [68, Equation (9)] such that given a perception latency Δ the resulting measurement covariance is $\mathbf{R}(\Delta) = (b/\Delta)\mathbf{I}$ with $b = 0.2$. This model is reasonable for demonstrative purposes in this chapter. However, in a more practical setting, $\mathbf{R}(\Delta)$ must be characterized according to the relation between latency and precision of the actual perception algorithms and sensors used, following a similar procedure to the one described in [46]. In this case, even when the measurement covariance is reduced as Δ is increased, the disturbance covariance $\mathbf{W}_{\mathsf{d}}(\Delta)$, given in (A.3) from Appendix A, increases.

Now, we test our scheduling approach assuming only two different latencies $\Delta^1 = 1/30, \Delta^2 = 4/30$ corresponding to the frame-rate of a typical camera or 4 times the frame-rate. Let $\mathbf{R}^1 = \mathbf{R}(\Delta^1), \mathbf{R}^2 = \mathbf{R}(\Delta^2)$ with the covariance model $\mathbf{R}(\Delta)$ described before and $\mathbf{W}_0 = 0.5\mathbf{I}$. Moreover, consider penalties $r^1 = 0.9\Delta^1, r^2 = 0.2\Delta^2$ which correspond to CPU loads of 90% and 20% respectively for each method. In addition, the parameters $\mathbf{Q} = \mathbf{Q}_f = \mathsf{diag}([2, 1, 2, 1, 2, 1]), \lambda_{\mathbf{x}} = 1,\ \lambda_r = 0.1$ were chosen. The SP^2 strategy was tested similarly as in the example of Section 2.6.1 with time step $h = 10^{-5}$, $T_f = 100$ and limited look-ahead window length $T = 10$. Figure 2.4 shows the resulting histograms for sample paths with $\bar{\mathbf{x}}_0 = [1, 1, 1, 1, 1, 1], \mathbf{P}_0 = \mathbf{I}$ for our approach with $m = 20$ and static schedules $\{1, 1, \dots\}, \{2, 2, \dots\}$ with 400 sample paths in each case. It is observed that our approach outperforms the static approaches in the sample cost $\mathcal{J}_{\mathsf{sp}}$. Moreover, the average CPU load is reduced with respect to the worst case value of 90%.

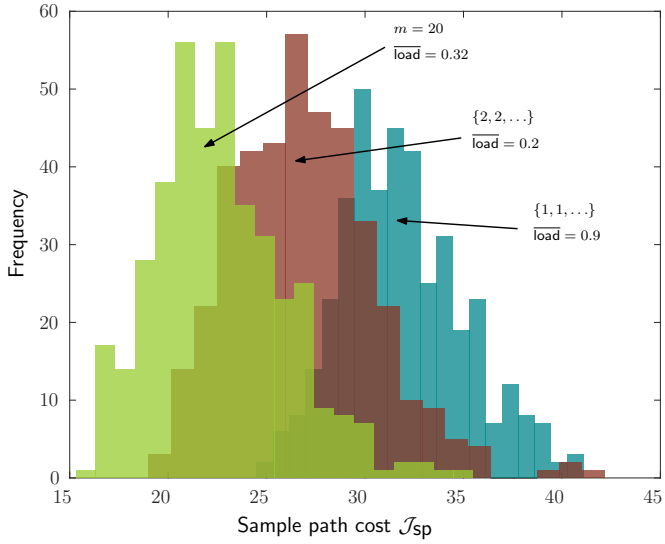

Figure 2.4: Resulting histograms for the sample path cost $\mathcal{J}_{\mathsf{sp}}$ for the example of Section 2.6.1 using the balanced SP^2 strategy with $m = 20$ and static schedules $\{1, 1, \dots\}, \{2, 2, \dots\}$. Average CPU load $\overline{\mathsf{load}}$ is also shown.

2.7 DISCUSSION

This chapter focused on the study of a control problem under a latency-precision trade-off and perception latency scheduling, for a simplified robot model with linear controller. It was shown that the stability of the closed-loop model is tied to the stability of a switching system. Moreover, the latency-precision trade-off was modeled by means of a cost function taking into account control precision and a penalty for each perception mode to incorporate energetic and CPU load points of view. Three main tools were proposed in this context to solve the problem: scheduling policy candidate construction using switching systems theory, an admissibility checking algorithm for policy candidates, and suboptimal strategies based upon the scheduling policy candidates. Different from general switching systems theory, we take advantage of the particular structure of the problem to provide new insights on the construction of multiple scheduling policies for optimization purposes and on its theoretical complexity. Finally, some illustrative examples were provided, including one application for a mobile robot.

In this chapter, the focus was specifically on the control aspect of the overall architecture depicted in Figure 1.4. The cooperation between multiple agents was not considered, and the task for the robot was assumed to be given. However, as discussed earlier, this task can be presented in the form of an exogenous time-varying reference. If this reference is constructed using an additional perception scheduler for target tracking and information fusion across a team of robots, the results obtained in this chapter can still be leveraged to locally optimize resources and tracking performance. In the upcoming chapter, we will delve into the complementary problem of observing an unknown input system, such as the state of a target of interest, using perception processes.

2.8 PROOFS

2.8.1 Proof of Theorem 2.6

In order to show Theorem 2.6 we first provide 2 auxiliary technical lemmas:

Lemma 2.21. *Let Γ be an admissible set of schedules and let $\gamma^\star := \gamma^\star(\bar{\mathbf{x}}_0;\Gamma)$ be a schedule obtained from* (2.4) *for the initial condition $\bar{\mathbf{x}}_0 \in \mathbb{S}_\Gamma^\star$. Thus, if the SP^2 strategy is used in* (2.3) *then $\bar{\mathbf{x}}[\mathsf{len}(\gamma^\star)] \in \mathbb{S}_0 \subseteq \text{int}\left(\mathbb{S}_\Gamma^\star\right)$.*

Proof: Since $\bar{\mathbf{x}}_0 \in \mathbb{S}_\Gamma^\star$ then there exist a non empty set $\Gamma' \subseteq \Gamma$ such that $\bar{\mathbf{x}}_0 \in \mathbb{S}_\gamma$ $\forall \gamma \in \Gamma'$. Moreover, $\bar{\mathbf{x}}_0^\top \mathbf{M}_\gamma \bar{\mathbf{x}}_0 \leq 1$ for any $\gamma \in \Gamma'$ and $\bar{\mathbf{x}}_0^\top \mathbf{M}_\gamma \bar{\mathbf{x}}_0 > 1$ for $\gamma \in \Gamma \setminus \Gamma'$. Hence, $\gamma^\star = \gamma^\star(\bar{\mathbf{x}}_0;\Gamma) \in \Gamma'$ and $\bar{\mathbf{x}}_0 \in \mathbb{S}_{\gamma^\star}$. Therefore, by construction of the set $\mathbb{S}_{\gamma^\star}$, system (2.3) after $\mathsf{len}(\gamma^\star)$ steps will result in $\bar{\mathbf{x}}[\mathsf{len}(\gamma^\star)] \in \mathbb{S}_0 \subset \mathbb{S}_\Gamma^\star$. □

Lemma 2.22. *Let Γ be an admissible set of schedules, $\Psi(\bullet;\mathbb{S}_\Gamma^\star)$ be its corresponding gauge function and let $\gamma^\star := \gamma^\star(\bar{\mathbf{x}}_0;\Gamma)$ be a schedule obtained from* (2.4) *for the initial condition $\bar{\mathbf{x}}_0 \neq 0$. Thus, if the SP^2 strategy is used in* (2.3) *then*

1. $\Psi(\bar{\mathbf{x}}[\mathsf{len}(\gamma^\star)];\mathbb{S}_\Gamma^\star) < \Psi(\bar{\mathbf{x}}_0;\mathbb{S}_\Gamma^\star)$
2. $\Psi(\bar{\mathbf{x}}[\mathsf{len}(\gamma^\star)];\mathbb{S}_0) < \Psi(\bar{\mathbf{x}}_0;\mathbb{S}_0)$

Proof: Note that by Lemma B.2-3) in Appendix B it is obtained that $\bar{\mathbf{x}}_0 \in \partial(\alpha_0 \mathbb{S}_\Gamma^\star)$ with

$$\alpha_0 := \Psi(\bar{\mathbf{x}}_0;\mathbb{S}_\Gamma^\star).$$

Similarly, one can obtain that $\bar{\mathbf{x}}[\mathsf{len}(\gamma^\star)] \in \partial(\alpha \mathbb{S}_\Gamma^\star)$ with $\alpha := \Psi(\bar{\mathbf{x}}[\mathsf{len}(\gamma^\star)];\mathbb{S}_\Gamma^\star)$. Moreover, since $\bar{\mathbf{x}}_0/\alpha_0 \in \partial\mathbb{S}_\Gamma^\star$, then Lemma 2.21 implies that

$$\frac{\bar{\mathbf{x}}[\mathsf{len}(\gamma^\star)]}{\alpha_0} \in \mathbb{S}_0 \subseteq \mathsf{int}(\mathbb{S}_\Gamma^\star).$$

Hence, $\partial(\alpha \mathbb{S}_\Gamma^\star) \subset \mathsf{int}(\alpha_0 \mathbb{S}_\Gamma^\star)$ strictly, unless $\alpha_0 = 0$ which is not possible since $\bar{\mathbf{x}}_0 \neq 0$. However, for that to be true, we require $\alpha < \alpha_0$ which is exactly item 1). For item 2), since Γ is admissible then $\mathbb{S}_0 \subset \mathbb{S}_\Gamma^\star$ and therefore

$$1 = \Psi\left(\frac{\mathbf{x}_0}{\alpha_0};\mathbb{S}_\Gamma^\star\right) < \Psi\left(\frac{\bar{\mathbf{x}}_0}{\alpha_0};\mathbb{S}_0\right)$$

by Lemma B.2-5). Equivalently $\alpha_0 < \Psi(\bar{\mathbf{x}}_0;\mathbb{S}_0)$. Recall that $\bar{\mathbf{x}}[\mathsf{len}(\gamma^\star)]/\alpha_0 \in \mathbb{S}_0$ and therefore

$$\Psi\left(\frac{\bar{\mathbf{x}}[\mathsf{len}(\gamma^\star)]}{\alpha_0};\mathbb{S}_0\right) \leq 1$$

or equivalently

$$\Psi(\bar{\mathbf{x}}[\mathsf{len}(\gamma^\star)];\mathbb{S}_0) \leq \alpha_0 < \Psi(\bar{\mathbf{x}}_0;\mathbb{S}_0).$$

□

Now, we are in position to prove Theorem 2.6. Note that as a consequence of the previous lemma, both $\Psi(\bullet;\mathbb{S}_\Gamma^\star)$ and $\Psi(\bullet;\mathbb{S}_0)$ are Lyapunov function candidates for (2.3). Note that $\Psi(\bullet;\mathbb{S}_\Gamma^\star)$ can be used for a similar setting. However, we use $\Psi(\bullet;\mathbb{S}_0)$ instead to take advantage of the fact that this Lyapunov function is independent of the set of schedules used. Henceforth, switching between admissible sets of schedules as in line 8 of Algorithm 2.2 is possible. Let $\{\ell_\kappa\}_{\kappa=0}^\infty$ be the sequence of switching time instants obtained by using the strategy SP^2 on (2.3). Moreover, define

$$\mathbf{y}[\kappa] := \bar{\mathbf{x}}[\ell_\kappa]$$

as the state at instants where the schedule changes. Consider the Lyapunov function candidate

$$V(\mathbf{y}[\kappa]) := \Psi(\mathbf{y}[\kappa];\mathbb{S}_0).$$

Lemma 2.22 implies $V(\mathbf{y}[\kappa+1]) - V(\mathbf{y}[\kappa]) < 0$ which in turn implies

$$\lim_{\kappa\to\infty} \mathbf{y}[\kappa] = \lim_{\kappa\to\infty} \bar{\mathbf{x}}[\ell_\kappa] = 0.$$

For any $\bar{\mathbf{x}}[k]$ with $k \notin \{\ell_\kappa\}_{\kappa=0}^\infty$, i.e. values of $\bar{\mathbf{x}}$ not at schedule change instants, it can be obtained from (2.3) that

$$\bar{\mathbf{x}}[k] = \mathbf{\Lambda}(\Delta^{p_{k-1}})\mathbf{\Lambda}(\Delta^{p_{k-2}})\cdots\mathbf{\Lambda}(\Delta^{p_{\ell'}})\bar{\mathbf{x}}[\ell']$$

with

$$\ell' = \max\{\ell \in \{\ell_\kappa\}_{\kappa=0}^\infty : \ell < k\}.$$

Hence $\|\bar{\mathbf{x}}[k]\| \leq \rho_k \|\bar{\mathbf{x}}[\ell']\|$ with ρ_k the spectral radius of $\mathbf{\Lambda}(\Delta^{p_{k-1}})\mathbf{\Lambda}(\Delta^{p_{k-2}})\cdots\mathbf{\Lambda}(\Delta^{p_{\ell'}})$, which since

$$\lim_{\ell'\to\infty} \bar{\mathbf{x}}[\ell'] = \lim_{\kappa\to\infty} \mathbf{y}[\kappa] = 0$$

implies that $\lim_{k\to\infty} \bar{\mathbf{x}}[k] = 0$ too.

2.8.2 Proof of Theorem 2.7

First, note that due to Lemma B.2-6) in Appendix B, $R > 1$ if and only if

$$\Psi(\bar{\mathbf{x}};\mathbb{S}_0) > 1, \forall \bar{\mathbf{x}} \in \partial\mathbb{S}_\Gamma^\star,$$

where $\Psi(\bullet;*)$ is the gauge function given in Definition B.1 from Appendix B. Now, we show that if $R > 1$ then $\mathbb{S}_0 \subset \mathbb{S}_\Gamma^\star$. We proceed by contradiction: let's assume that there exists $\bar{\mathbf{x}} \in \mathbb{S}_0$ such that $\Psi(\bar{\mathbf{x}};\mathbb{S}_\Gamma^\star) > 1$. Let

$$\beta := \Psi(\bar{\mathbf{x}};\mathbb{S}_\Gamma^\star)$$

and thus $\bar{\mathbf{x}}/\beta \in \partial\mathbb{S}_\Gamma^\star$ due to Lemma B.2-3). Henceforth, since $R > 1$, then $\Psi(\bar{\mathbf{x}}/\beta;\mathbb{S}_0) > 1$ or equivalently $\Psi(\bar{\mathbf{x}};\mathbb{S}_0) > \beta > 1$ due to Lemma B.2-4) and due to the assumption $\beta = \Psi(\bar{\mathbf{x}};\mathbb{S}_\Gamma^\star) > 1$. However, for $\bar{\mathbf{x}}$ to belong in $\mathbb{S}_0$ we require $\Psi(\bar{\mathbf{x}};\mathbb{S}_0) < 1$ which leads to a contradiction. Therefore,

$$\beta = \Psi(\bar{\mathbf{x}};\mathbb{S}_\Gamma^\star) \leq 1$$

for all $\bar{\mathbf{x}} \in \mathbb{S}_0$. Hence,

$$\bar{\mathbf{x}} \in \partial(\beta\mathbb{S}_\Gamma^\star) \subseteq \beta\mathbb{S}_\Gamma^\star \subseteq \mathbb{S}_\Gamma^\star$$

due to Lemma B.2-3) and Lemma B.2-1) for any $\bar{\mathbf{x}} \in \mathbb{S}_0$ and thus $\mathbb{S}_0 \subseteq \mathbb{S}_\Gamma^\star$. Finally, note that since all points $\bar{\mathbf{x}} \in \partial\mathbb{S}_\Gamma^\star$ have $\bar{\mathbf{x}}^\top \mathbf{M}_0 \bar{\mathbf{x}} > 1$ (by the assumption $R > 1$), then they do not belong to $\mathbb{S}_0$. Thus,

$$\mathbb{S}_0 \cap \partial\mathbb{S}_\Gamma^\star = \varnothing$$

and then $\mathbb{S}_0 \subset \mathsf{int}(\mathbb{S}_\Gamma^\star)$. Now, we show that if $\mathbb{S}_0 \subset \mathsf{int}(\mathbb{S}_\Gamma^\star)$, then $R > 1$. From here, note that $\mathbb{S}_0 \subseteq \mathsf{int}(\mathbb{S}_\Gamma^\star)$. We proceed by contradiction: assume that there exists $\bar{\mathbf{x}} \in \partial\mathbb{S}_\Gamma^\star$ such that $\bar{\mathbf{x}}^\top \mathbf{M}_0 \bar{\mathbf{x}} \leq 1$. Hence,

$$\bar{\mathbf{x}} \in \mathbb{S}_0 \subseteq \mathsf{int}(\mathbb{S}_\Gamma^\star)$$

which is a contradiction. Therefore, $R > 1$ for all $\bar{\mathbf{x}} \in \partial\mathbb{S}_\Gamma^\star$ which concludes the proof.

2.8.3 Proof of Corollary 2.8

Let $\bar{\mathbf{x}}^*$ be the critical point of (2.5) which leads to the global minimum R and note that $\bar{\mathbf{x}}^* \in \partial\mathbb{S}_\Gamma^\star$. Hence, $\bar{\mathbf{x}}^* \in \partial\mathbb{S}_\gamma$ for at least one $\gamma \in \Gamma$ and at most all $\gamma \in \Gamma$. Let Γ' be the set of all those $\gamma \in \Gamma$ and note that $\Gamma' \in \mathcal{P}(\Gamma)$. Therefore, $(\bar{\mathbf{x}}^*)^\top \mathbf{M}_\gamma (\bar{\mathbf{x}}^*) = 1$ for all $\gamma \in \Gamma'$ and is a local minimum of (2.6) for such Γ'. Since $\bar{\mathbf{x}}^* \in \partial\mathbb{S}_\Gamma^\star$ and $\bar{\mathbf{x}}^* \in \partial\mathbb{S}_\gamma$ only for $\gamma \in \Gamma'$, thus $\bar{\mathbf{x}}^* \notin \mathbb{S}_\gamma$ for $\gamma \in \Gamma \setminus \Gamma'$, and thus $\bar{\mathbf{x}}^* \in \mathcal{X}_{\Gamma'}$. Now, for $\bar{\mathbf{x}}^*$ to be the global minimum, critical points $\bar{\mathbf{x}}$ comply $(\bar{\mathbf{x}}^*)^\top \mathbf{M}_0 (\bar{\mathbf{x}}^*) \leq \bar{\mathbf{x}}^\top \mathbf{M}_0 \bar{\mathbf{x}}$ in (2.7).

2.8.4 Proof of Theorem 2.11

In order to show Theorem 2.11 we proceed by duality analysis, which is summarized in the following two technical lemmas.

Lemma 2.23. *Any $\bar{\mathbf{x}}^*$ which is a regular critical point of program* (2.6) *comply with*

$$\mathbf{G}(\boldsymbol{\lambda})\bar{\mathbf{x}}^* = 0$$
$$(\bar{\mathbf{x}}^*)^\top \mathbf{M}_\gamma (\bar{\mathbf{x}}^*) = 1 \quad \forall \gamma \in \Gamma'$$

for some unique $\boldsymbol{\lambda} = \{\lambda_\gamma\}_{\gamma \in \Gamma'}$.

Proof: Let

$$\mathcal{L}(\bar{\mathbf{x}}, \boldsymbol{\lambda}) = \bar{\mathbf{x}}^\top \mathbf{M}_0 \bar{\mathbf{x}} + \sum_{\gamma \in \Gamma'} \lambda_\gamma (\bar{\mathbf{x}}^\top \mathbf{M}_\gamma \bar{\mathbf{x}} - 1)$$

be the Lagrangian for (2.6) with Lagrange multipliers $\boldsymbol{\lambda}$. Moreover, $\mathcal{L}(\bar{\mathbf{x}}, \boldsymbol{\lambda})$ can be written as

$$\mathcal{L}(\bar{\mathbf{x}}, \boldsymbol{\lambda}) = \bar{\mathbf{x}}^\top \mathbf{G}(\boldsymbol{\lambda}) \bar{\mathbf{x}} - \sum_{\gamma \in \Gamma'} \lambda_\gamma \tag{2.13}$$

According to [122, Propositon 3.1.1], all regular critical points $\bar{\mathbf{x}}^*$ must satisfy $(\bar{\mathbf{x}}^*)^\top \mathbf{M}_\gamma \bar{\mathbf{x}}^* = 1$, $\forall \gamma \in \Gamma'$ and $\nabla_{\bar{\mathbf{x}}} \mathcal{L}(\bar{\mathbf{x}}^*, \boldsymbol{\lambda}) = 0$ or equivalently $\mathbf{G}(\boldsymbol{\lambda})\bar{\mathbf{x}}^* = 0$ for some unique Lagrange multipliers $\boldsymbol{\lambda}$. □

Lemma 2.24. *Let $\boldsymbol{\lambda}^*$ be any critical point of the dual problem of* (2.6) *and $\mathbf{G}(\boldsymbol{\lambda})$ be its dual objective (in the sense of [127, Chapter 5]). Moreover, let $(\bar{\mathbf{x}}^*)^\top \mathbf{M}_0 (\bar{\mathbf{x}}^*) - \mathbf{G}(\boldsymbol{\lambda}^*)$ be the duality gap between the primal in* (2.6) *and its dual. Then, any regular critical point*

$\bar{\mathbf{x}}^*$ *of the primal has zero duality gap. Consequently, the duality gap is zero if and only if* $\boldsymbol{\lambda}^*$ *is also a critical point of*

$$\begin{aligned} \max_{\lambda} \quad & \left(-\sum_{\gamma \in \Gamma'} \lambda_\gamma \right) \\ \text{s.t. } & \det \mathbf{G}(\boldsymbol{\lambda}) = 0 \end{aligned} \tag{2.14}$$

Proof: First, we compute the dual function (see [127, Page 216]) of the objective in (2.6) as

$$\mathbf{G}(\boldsymbol{\lambda}) = \inf_{\bar{\mathbf{x}} \in \mathbb{R}^n} \mathcal{L}(\bar{\mathbf{x}}, \boldsymbol{\lambda}) = \left(\inf_{\bar{\mathbf{x}} \in \mathbb{R}^n} \bar{\mathbf{x}}^\top \mathbf{G}(\boldsymbol{\lambda}) \bar{\mathbf{x}} \right) - \sum_{\gamma \in \Gamma'} \lambda_\gamma$$

where $\mathcal{L}(\bar{\mathbf{x}}, \boldsymbol{\lambda})$ is the Lagrangian given in (2.13). Therefore, we conclude that

$$\mathbf{G}(\boldsymbol{\lambda}) = \begin{cases} -\sum_{\gamma \in \Gamma'} \lambda_\gamma & \text{if } \mathbf{G}(\boldsymbol{\lambda}) \succeq 0 \text{ or } \det \mathbf{G}(\boldsymbol{\lambda}) = 0 \\ -\infty & \text{otherwise} \end{cases}$$

since the infimum of the form $\bar{\mathbf{x}}^\top \mathbf{G}(\boldsymbol{\lambda}) \bar{\mathbf{x}}$ is either zero (for $\mathbf{G}(\boldsymbol{\lambda}) \succeq 0$ or $\det \mathbf{G}(\boldsymbol{\lambda}) = 0$ with $\bar{\mathbf{x}}$ in the kernel of $\mathbf{G}(\boldsymbol{\lambda})$) or $-\infty$ (see [127, Page 220]). Hence, the dual program results in

$$\begin{aligned} \max_{\lambda} \quad & \left(-\sum_{\gamma \in \Gamma'} \lambda_\gamma \right) \\ \text{s.t. } & \mathbf{G}(\boldsymbol{\lambda}) \succeq 0 \end{aligned}$$

making the constraints explicit (see [127, Page 224]), ignoring the case when $\mathbf{G}(\boldsymbol{\lambda}) = -\infty$. Note that due to Lemma 2.23 it is true that $\mathbf{G}(\boldsymbol{\lambda}^*)\bar{\mathbf{x}}^* = 0$ for some $\boldsymbol{\lambda}^*$. Hence, we can multiply

$$\begin{aligned} (\bar{\mathbf{x}}^*)^\top \mathbf{G}(\boldsymbol{\lambda}^*) \bar{\mathbf{x}}^* &= (\bar{\mathbf{x}}^*)^\top \mathbf{M}_0(\bar{\mathbf{x}}^*) + \sum_{\gamma \in \Gamma'} \lambda_\gamma^* (\bar{\mathbf{x}}^*)^\top \mathbf{M}_\gamma(\bar{\mathbf{x}}^*) \\ &= (\bar{\mathbf{x}}^*)^\top \mathbf{M}_0(\bar{\mathbf{x}}^*) + \sum_{\gamma \in \Gamma'} \lambda_\gamma^* = (\bar{\mathbf{x}}^*)^\top \mathbf{M}_0(\bar{\mathbf{x}}^*) - \mathbf{G}(\boldsymbol{\lambda}^*) \\ &= 0 \end{aligned}$$

which results in zero duality gap, only possible if $\boldsymbol{\lambda}^*$ is a critical point of the dual problem by the definition of $\mathbf{G}(\boldsymbol{\lambda})$. Note that zero duality gap plus the condition that $(\bar{\mathbf{x}}^*)^\top \mathbf{M}_\gamma(\bar{\mathbf{x}}^*) = 1$ implies $\bar{\mathbf{x}}^* \neq 0$. Then, $\mathbf{G}(\boldsymbol{\lambda}^*)\bar{\mathbf{x}}^* = 0$ if and only if $\det \mathbf{G}(\boldsymbol{\lambda}^*) = 0$ and $\bar{\mathbf{x}}^*$ in the kernel of $\mathbf{G}(\boldsymbol{\lambda}^*)$. □

Using these results, we proceed to show Theorem 2.11. Build a Lagrangian for (2.14) as

$$\mathcal{L}'(\boldsymbol{\lambda}, \mu) = -\sum_{\gamma \in \Gamma'} \lambda_\gamma + \mu \det \mathbf{G}(\boldsymbol{\lambda})$$

with Lagrange multiplier $\mu \in \mathbb{R}$. Thus, $\boldsymbol{\lambda}^*$ comply with

$$\nabla_\lambda \mathcal{L}'(\boldsymbol{\lambda}^*, \mu) = 0$$

from [122, Proposition 3.1.1] and

$$\frac{\partial \mathcal{L}'}{\partial \lambda_\gamma} = -1 + \mu \ \mathrm{tr}(\mathbf{G}(\boldsymbol{\lambda}^*)^\dagger \mathbf{M}_\gamma) = 0$$

using Jacobi's formula for the derivative of the determinant [123, Page 29]. Thus,

$$\mathrm{tr}(\mathbf{G}(\boldsymbol{\lambda}^*)^\dagger \mathbf{M}_\gamma) = \frac{1}{\mu}$$

or equivalently

$$\mathrm{tr}(\mathbf{G}(\boldsymbol{\lambda}^*)^\dagger \mathbf{M}_\gamma) = \ \mathrm{tr}(\mathbf{G}(\boldsymbol{\lambda}^*)^\dagger \mathbf{M}_\nu), \quad \forall \gamma, \nu \in \Gamma'$$

This result, in addition to the condition $\det \mathbf{G}(\boldsymbol{\lambda}^*) = 0$ result in (2.8). From Lemma 2.24 we can also conclude that any critical point complies with $(\bar{\mathbf{x}}^*)\mathbf{M}_0(\bar{\mathbf{x}}^*) = -\sum_{\gamma \in \Gamma'} \lambda_\gamma^*$ since this condition is equivalent to the zero duality gap property of (2.14). Finally, due to Lemma 2.23, we know that $\mathbf{G}(\boldsymbol{\lambda}^*)\bar{\mathbf{x}}^* = 0$ and this is true only if $\bar{\mathbf{x}}^*$ is in the kernel of $\mathbf{G}(\boldsymbol{\lambda}^*)$.

2.8.5 Proof of Corollary 2.13

Recall that the rank of $\mathbf{G}(\boldsymbol{\lambda})$ is equivalent to the size of the largest invertible sub-matrix $\mathbf{G}(\boldsymbol{\lambda})$ [123, Page 12]. We proceed by contradiction: assume that $\boldsymbol{\lambda}$ doesn't comply with at least one equation (2.10) for some pair (i, j). This would mean that $\mathbf{G}_{ij}(\boldsymbol{\lambda})$ is invertible resulting in the rank of $\mathbf{G}(\boldsymbol{\lambda})$ to be $n-1$. However, the nullity of $\mathbf{G}(\boldsymbol{\lambda})$ was greater than 1. Then, by the rank-nullity theorem [123, Page 6] the rank should have been less than $n-1$ leading to a contradiction.

2.8.6 Proof of Proposition 2.14

Let $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_r\}$ be arbitrarily $r = n - |\Gamma'|$ vectors $\mathbf{v}_1, \ldots, \mathbf{v}_r \in \mathbb{R}^n$. Build a matrix

$$\mathbf{Y}_V(\bar{\mathbf{x}}) = [\mathbf{Y}(\bar{\mathbf{x}}), \mathbf{v}_1, \ldots, \mathbf{v}_r].$$

The vectors $\{\mathbf{M}_\gamma \bar{\mathbf{x}}\}_{\gamma \in \Gamma'}$ are linearly dependent if and only if the matrix $\mathbf{Y}_V(\bar{\mathbf{x}})$ is singular for any choice of $\mathbf{v}_1, \ldots, \mathbf{v}_r$. Using the Laplace expansion of the determinant of $\mathbf{Y}_V(\bar{\mathbf{x}})$ [123, Page 8] we obtain

$$\det \mathbf{Y}_V(\bar{\mathbf{x}}) = \sum_{\alpha \in \mathcal{A}} q_\alpha(V) \det W_\alpha(\bar{\mathbf{x}})$$

where $q_\alpha(V)$ are polynomials in the components of the vectors in V. In order to comply $\det \mathbf{Y}_V(\bar{\mathbf{x}}) = 0$ for any set of vectors V, then $\det \mathbf{Y}_\alpha(\bar{\mathbf{x}}) = 0$ for all $\alpha \in \mathcal{A}$.

2.8.7 Proof of Proposition 2.17

The proof follows by performing a Karp reduction [128, Definition 15.15] of the 3-SAT problem, known to be NP-complete [128, Theorem 15.22], to an instance of Problem 2.1. First, consider an instance of Problem 2.1 with $\mathbf{A} = \mathbf{0}, \mathbf{B} = \mathbf{I}$ and $\Delta^1 = \cdots = \Delta^D = \Delta, T_f = \alpha\Delta, \alpha \in \mathbb{N}$ with

$$(\mathbf{I} + \mathbf{L}^1\Delta), \ldots, (\mathbf{I} + \mathbf{L}^D\Delta)$$

being $0-1$ left stochastic matrices. Moreover, let $\mathbf{P}_0 = 0, W_0 = 0, \Sigma^1 = \cdots = \Sigma^D = 0$ as well as $\mathbf{Q} = 0, \lambda_r = 0, \lambda_{\mathbf{x}} = 1$ and $\mathbf{Q}_f = \mathsf{diag}(c)$ for some $0-1$ vector c. Finally, consider the initial condition $\bar{\mathbf{x}}_0$ to be a $0-1$ vector as well. This results in $\mathbf{\Lambda}(\Delta^{p_k}) = \mathbf{I} + \mathbf{L}^{p_k}\Delta$ being $0-1$ matrices and as a consequence $\bar{\mathbf{x}}[\alpha]$ is a $0-1$ vector. Thus,

$$\mathcal{J}(p) = \bar{\mathbf{x}}[\alpha]^\top \mathsf{diag}(c)\bar{\mathbf{x}}[\alpha] = c^\top \bar{\mathbf{x}}[\alpha].$$

Hence, solving this instance of Problem 2.1 requires to find a schedule $\mathbf{P}_0, \ldots, p_{\alpha-1}$ such that the linear function $c^\top \mathbf{x}[\alpha]$ is minimized, or equivalently $-c^\top \mathbf{x}[\alpha]$ is maximized. Thus, the Problem [118, Problem (P)] for stochastic matrices, initial condition and vector c all with $0-1$ components can be reduced to Problem 2.1. Finally, the proof of [118, Theorem 3.1] implies that the 3-SAT problem can be reduced to this instance of [118, Problem (P)].

2.8.8 Proof of Proposition 2.18

The proof of correctness follows by a direct application of the dynamic programming algorithm [124, Page 23]. For the complexity, note that the largest amount of recursive calls in Algorithm 2.6 is obtained when the schedule covering the smallest amount of time c over $[0, T_f]$ is chosen repeatedly. In this case, the depth of recursive calls to `dynprog` is $\lfloor T_f/c \rfloor$. Hence, due to line 2 in Algorithm 2.6, the total number of recursive calls is at most $m^{\lfloor T_f/c \rfloor}$.

2.8.9 Proof of Theorem 2.20

For the proof, for any matrix P let $\|P\|$ denote its matrix norm [123, Section 5.6] induced by the Euclidean norm. To show Theorem 2.20 we provide the following.

Lemma 2.25. *Let p be a fixed perception schedule over $[0, T_f]$. Moreover, let any compact sets $\mathcal{B}_{\bar{\mathbf{x}}} \in \mathbb{R}^n, \mathcal{B}_{\mathbf{P}} \in \mathbb{R}^{n\times n}$ and any $\varepsilon > 0$. Then, there exists $\delta_{\bar{\mathbf{x}}}, \delta_{\mathbf{P}} > 0$ such that for any $\bar{\mathbf{x}}_0, \bar{\mathbf{x}}_0' \in \mathcal{B}_{\bar{\mathbf{x}}}$ and any $\mathbf{P}, \mathbf{P}' \in \mathcal{B}_{\mathbf{P}}$ with $\|\bar{\mathbf{x}}_0 - \bar{\mathbf{x}}_0'\| < \delta_{\bar{\mathbf{x}}}$ and $\|\mathbf{P} - \mathbf{P}'\| < \delta_{\mathbf{P}}$ implies*

$$|\mathcal{J}(p; \bar{\mathbf{x}}_0, \mathbf{P}_0) - \mathcal{J}(p; \bar{\mathbf{x}}_0', \mathbf{P}_0')| < \varepsilon.$$

Proof: First, let

$$r_{\bar{\mathbf{x}}} = \sup\{\|\bar{\mathbf{x}}\| : \bar{\mathbf{x}} \in \mathcal{B}_{\bar{\mathbf{x}}}\}.$$

Then, we leverage continuity of solutions of (A.4) with respect to initial conditions. Concretely, using the bound in [129, Page 356 - Equation (9.28)] applied to (A.4) for the compact interval $[0, T_f]$ it follows that

$$\|\bar{\mathbf{x}}(t) - \bar{\mathbf{x}}'(t)\| \leq \|\bar{\mathbf{x}}_0 - \bar{\mathbf{x}}_0'\| \exp(c_{\bar{\mathbf{x}}} T_f) \leq \delta_{\bar{\mathbf{x}}} \exp(c_{\bar{\mathbf{x}}} T_f), \forall t \in [0, T_f]$$

for some constant $c_{\bar{\mathbf{x}}} \in \mathbb{R}$. A similar reasoning leads to obtain

$$\|\mathbf{P}(t) - \mathbf{P}'(t)\| \leq \delta_P \exp(c_{\mathbf{P}} T_f), \forall t \in [0, T_f]$$

for some constant $c_{\mathbf{P}} \in \mathbb{R}$. Moreover,

$$|\bar{\mathbf{x}}(t)^\top \mathbf{Q} \bar{\mathbf{x}}(t) - \bar{\mathbf{x}}(t)'^T \mathbf{Q} \bar{\mathbf{x}}'(t)| = |(\bar{\mathbf{x}}(t) + \bar{\mathbf{x}}'(t))^\top \mathbf{Q} (\bar{\mathbf{x}}(t) - \bar{\mathbf{x}}'(t))| \leq 2 r_{\bar{\mathbf{x}}} \|\mathbf{Q}\| \delta_{\bar{\mathbf{x}}} \exp(c_{\bar{\mathbf{x}}} T_f)$$

using the matrix norm properties in [123, Section 5.6]. Similarly,

$$|\mathsf{tr}(\mathbf{Q}\mathbf{P}(t)) - \mathsf{tr}(\mathbf{Q}\mathbf{P}'(t))| = |\mathsf{tr}(\mathbf{Q}(\mathbf{P}(t) - P'(t)))| \leq n \|\mathbf{Q}(\mathbf{P}(t) - P'(t))\| \leq n \|\mathbf{Q}\| \delta_{\mathbf{P}} \exp(c_{\mathbf{P}} T_f).$$

Hence, using these results on

$$e := |\mathcal{J}(p; \bar{\mathbf{x}}_0, \mathbf{P}_0) - \mathcal{J}(p; \bar{\mathbf{x}}'_0, \mathbf{P}'_0)|$$

which using (A.5) leads to conclude that for appropriate (sufficiently small) $\delta_{\bar{\mathbf{x}}}, \delta_P > 0$:

$$e \leq 2\lambda_{\mathbf{x}} r_{\bar{\mathbf{x}}} \delta_{\bar{\mathbf{x}}} (\|\mathbf{Q}\| + \|\mathbf{Q}_f\|) \exp(c_{\bar{\mathbf{x}}} T_f) + n \lambda_{\mathbf{x}} \delta_{\mathbf{P}} (\|\mathbf{Q}\| + \|\mathbf{Q}_f\|) \exp(c_{\mathbf{P}} T_f)) \leq \varepsilon$$

□

We proceed to show Theorem 2.20. First, given $\varepsilon > 0$ and $\mathcal{B}_{\bar{\mathbf{x}}}, \mathcal{B}_{\mathbf{P}}$ choose $\delta_{\bar{\mathbf{x}}}, \delta_{\mathbf{P}}$ as in Lemma 2.25 and note that this values are independent on initial conditions as long as they lie in $\mathcal{B}_{\bar{\mathbf{x}}}, \mathcal{B}_P$. Hence, there is $N > 0$ such that we can partition the compact set $\mathcal{B}_{\bar{\mathbf{x}}}$ into regions $\mathcal{B}^1_{\bar{\mathbf{x}}}, \ldots, \mathcal{B}^N_{\bar{\mathbf{x}}}$ which cover all $\mathcal{B}_{\bar{\mathbf{x}}}$ and

$$\|\bar{\mathbf{x}} - \bar{\mathbf{x}}'\| \leq \delta_{\bar{\mathbf{x}}}, \forall \bar{\mathbf{x}}, \bar{\mathbf{x}}' \in \mathcal{B}^i_{\bar{\mathbf{x}}}, \forall i \in \{1, \ldots, N\}.$$

A similar partition of N sections can be performed for $\mathcal{B}_{\mathbf{P}}$ as well. Now, choose any $\bar{\mathbf{x}}^i_0 \in \mathcal{B}^i_{\bar{\mathbf{x}}}, \mathbf{P}^i_0 \in \mathcal{B}^i_{\mathbf{P}}$ for partition i and denote with p^{*i} the optimal schedule in this case for the general setting of Problem 2.1. As a consequence of Lemma 2.25, the cost of p^{*i} for any $\bar{\mathbf{x}}'_0 \in \mathcal{B}^i_{\bar{\mathbf{x}}}, \mathbf{P}'_0 \in \mathcal{B}^i_{\mathbf{P}}$ differs from the optimal one in that partition by at most ε. Now, let Γ be any admissible set of schedules. Moreover, since p^{*i} is a solution of Problem 2.1, then it must be a stabilizing perception schedule for $\bar{\mathbf{x}}[0] = \bar{\mathbf{x}}^i_0$. Thus, there exists $\ell^i < \infty$ (sufficiently large) with

$$T_f \leq \sum_{k=0}^{\ell^i - 1} \Delta^{p^{*i}_k}$$

such that

$$\|\bar{\mathbf{x}}[\ell^i]\|^2 < \|\mathbf{M}_0\|^{-1} \min_{\gamma \in \Gamma} (\bar{\mathbf{x}}^i_0)^\top \mathbf{M}_\gamma \bar{\mathbf{x}}^i_0$$

for the truncated schedule $\gamma^i = \{p^{*i}_k\}_{k=0}^{\ell^i - 1}$. This last relation implies

$$(\bar{\mathbf{x}}^i_0)^\top M_{\gamma^i} \bar{\mathbf{x}}^i_0 = (\mathbf{\Lambda}^{\gamma^i} \bar{\mathbf{x}}^i_0)^\top \mathbf{M}_0 (\mathbf{\Lambda}^{\gamma^i} \bar{\mathbf{x}}^i_0) = \bar{\mathbf{x}}[\ell^i]^\top \mathbf{M}_0 \bar{\mathbf{x}}[\ell^i] \leq \|\mathbf{M}_0\| \|\bar{\mathbf{x}}[\ell^i]\|^2 < \min_{\gamma \in \Gamma} (\bar{\mathbf{x}}^i_0)^\top \mathbf{M}_\gamma \bar{\mathbf{x}}^i_0.$$

Hence, $\Gamma^i = \Gamma \cup \{\gamma^i\}$ is admissible and complies $\gamma^\star(\bar{\mathbf{x}}^i_0; \Gamma^i) = \gamma^i$ by (2.4) and similarly for any $\bar{\mathbf{x}}_0 \in B^i_{\bar{\mathbf{x}}}$ and $\delta_{\bar{\mathbf{x}}}$ sufficiently small. Therefore, the result follows by choosing $m = N$ and all $\Gamma^1, \ldots, \Gamma^m$ constructed in this way, one for each partition. The reason is that for any $\bar{\mathbf{x}}_0 \in \mathcal{B}_{\bar{\mathbf{x}}}, \mathbf{P}_0 \in \mathcal{B}_{\mathbf{P}}$, a schedule with cost differing from the optimal one by at most ε is always evaluated in Algorithm 2.6.

Chapter Three

Perception Latency Scheduling In Estimation

In this chapter, we explore the perception latency scheduling problem for estimation, which is particularly relevant in target tracking tasks where a robot aims to estimate the state of a target using onboard sensors such as cameras, lidar, or radar. The main focus of this chapter centers around the estimation aspect, abstracting away the control components within the robot architecture. As a result, our architecture takes the form illustrated in Figure 3.1, with the *Target Estimation Block* set to evolve into a fusion block once multiple agents are integrated in subsequent chapters.

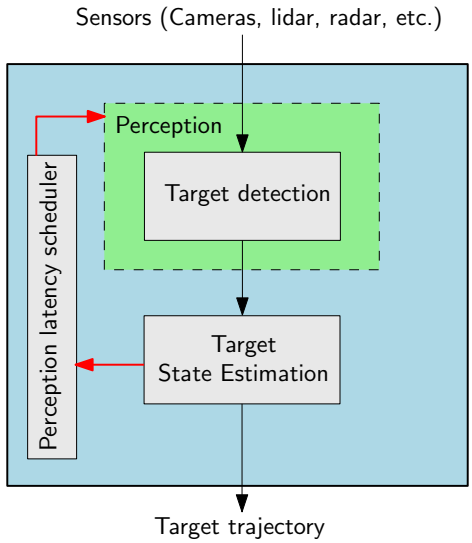

Figure 3.1: Individual robot architecture with perception latency scheduler for target state estimation.

Within this context, the robot's objective is to use measurements from onboard sensors, apply a perception process to detect the target's position of interest, and reconstruct its trajectory. This trajectory can then be used to construct formation references or enable

tracking control.

The architecture depicted in Figure 3.1 forms the basis of the framework introduced in this chapter, referred to as PLATE (Perception LATency aware Estimator). PLATE combines a target state estimator with a scheduling policy. Its primary objective is to obtain an estimate for the target state while optimizing a performance metric akin to that in Chapter 2, encompassing estimation performance and resource usage in terms of attention and CPU load.

As discussed in Chapter 2, the perception scheduling problem is susceptible to a combinatorial explosion. Thus, the main contribution of PLATE lies in an efficient scheduling methodology for approximating optimal scheduling. While PLATE is supported by rigorous formal analysis, it is also validated through several experiments, including one involving standard benchmark datasets with real-world data. The contributions of this chapter were also published in [5].

3.1 RELATED WORK

In the context of target tracking, anytime neural networks were used in [58] in order to schedule different deep learning jobs with latency chosen such that real-time deadlines are maintained with the best possible accuracy overall. However, being a general deep learning scheduler, [58] does not take into account that for a tracking problem, while a longer latency might lead to a better accuracy for the detector, the model uncertainty might increase as well due to the delay introduced by the perception-latency [8].

Instead of using an adaptive perception-latency, some works have used frame-skipping techniques in order to reduce computational load. In [60], the authors propose an event-triggered rule in which the detection is only updated under certain events, based on the expected motion of the target. This approach reduces the computational burden of the perception task by effectively skipping frames when the expected motion of the target does not require to. Other approaches follow a similar idea such as the one in [73] in which the perception task goes to an idle state depending on the expected state of the environment, obtaining an energy efficient use of battery powered smart cameras. Similarly, in [74] a neural network is trained based on examples in order to make a decision on whether or not the detection for some target should be updated. Although some of these methods shows good experimental performance in certain scenarios, they are mostly based in several heuristic rules from which it is hard to provide formal guarantees on the impact over the resource usage and quality for target tracking in the general case.

3.2 PROBLEM STATEMENT

Consider the following target model, with state $\mathbf{x}(t) \in \mathbb{R}^n$ e.g. containing the target's position, and possibly higher order dynamics such as velocity. Assume a simplified motion model given by the following Stochastic Differential Equation (SDE):

$$\mathrm{d}\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t)\mathrm{d}t + \mathbf{B}\mathrm{d}\mathbf{w}(t), \quad t \geq 0 \tag{3.1}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}, \mathbf{B} \in \mathbb{R}^{n\times n_{\mathbf{w}}}$ and $\mathbf{w}(t)$ is a $n_{\mathbf{w}}$-dimensional Wiener processes with covariance $\mathsf{cov}\{\mathbf{w}(s), \mathbf{w}(r)\} = \mathbf{W}\min(s, r)$ [119, Page 63]. As usual, the process $\mathbf{w}(t)$ models disturbances, unknown inputs for the target and non-modeled dynamics. Moreover, $\mathbf{x}(0)$ is normally distributed with mean $\mathbf{x}_0$ and covariance $\mathbf{P}_0$.

The goal is to construct an estimation framework for the state $\mathbf{x}(t)$ using available sensors, e.g. vision or range. Consider a similar perception mechanism model as in Chapter 2. However, we use additional simplifications which will aid the development of efficient scheduling algorithms. First, assume that the sensors produce the raw measurements with a minimum sampling period of Δ_s. Similarly as before, the system can choose from D different perception methods to process each raw measurement, producing the position of the target through a detection process. Processed measurements are represented by $\mathbf{Cx}[k]$ with some constant matrix $\mathbf{C} \in \mathbb{R}^{n_\mathbf{z} \times n}$ with $(\mathbf{A}, \mathbf{C})$ observable. Each method has a different perception-latency in $\{\Delta^1, \ldots, \Delta^D\}$. These latencies are multiples of the sampling period Δ_s of the sensors leading to a frame-skipping technique. As in Chapter 2, if method $p_k \in \{1, \ldots, D\}$ is chosen at $t = \tau_k$, a new measurement $\mathbf{z}[k] = \mathbf{Cx}[k] + \mathbf{v}[k]$ is available at $t = \tau_k + \Delta^{p_k}$ where $\mathbf{v}[k]$ is a noise modeling the accuracy of the perception method and the perception-latency and accuracy relation for the perception method is modeled by the covariance matrix $\mathbf{R}^{p_k} = \mathsf{cov}\{\mathbf{v}[k]\}$.

The main issue we address in this chapter is to obtain an estimate of the target state $\mathbf{x}(t)$ at any time $t \geq 0$ and choose which perception method to use at each time slot $[\tau_k, \tau_{k+1})$ provided some performance measure is optimized:

Problem 3.1 (Perception scheduling in estimation)**.** *Design a causal estimator which picks a perception method p_k at each $t = \tau_k$ and produces an estimate $\hat{\mathbf{x}}(t)$ of $\mathbf{x}(t)$ with $\hat{\mathbf{P}}(t) := \mathsf{cov}\{\mathbf{x}(t) - \hat{\mathbf{x}}(t)\}$ using only information prior to the instant t. Moreover, the estimator output must minimize*

$$\mathcal{J}(\hat{\mathbf{x}}, p) = \frac{1}{T_f}\left(\int_0^{T_f} \mathit{tr}\left(\hat{\mathbf{P}}(t)\right) dt\right) + \frac{\lambda_\alpha}{T_f}\left(\sum_{k=0}^{\alpha} r^{p_k}\right) \tag{3.2}$$

for some window of interest $[0, T_f]$, where $\alpha := \mathsf{att}(p; [0, T_f])$, $\lambda_\alpha > 0$ and $r^{p_k} \geq 0$ are additional penalties assigned to each perception method. Recall, that $\mathsf{att}(p; [0, T_f])$ is the attention of p in the interval $[0, T_f]$ as defined in Chapter 2.

Remark 3.2. *The cost function in* (3.2) *is used to capture the perception latency-accuracy trade-off. The integral term measures the mean expected squared error of $\hat{\mathbf{x}}(t)$ by evaluating*

$$\mathbb{E}(\hat{\mathbf{x}}(t) - \mathbf{x}(t))^\top (\hat{\mathbf{x}}(t) - \mathbf{x}(t)) \equiv \mathit{tr}(\hat{\mathbf{P}}(t)).$$

Thus, the first term in (3.2) *represents the quality of the estimation $\hat{\mathbf{x}}(t)$. The summation term in* (3.2) *introduces the rewards $r^1, \ldots, r^D$ just as in Chapter 2, in order to enable the cost to incorporate the use of resources in the form of attention or CPU load.*

We propose PLATE as a strategy to compute the optimal schedule p and the optimal estimations $\hat{\mathbf{x}}(t)$ in the sense of Problem 3.1. Causality in Problem 3.1 is required in an online target tracking context. Thus, PLATE works as a causal estimator as is summarized in Algorithm 3.1 and is comprised by the combination of an estimation stage and a perception-latency scheduling policy.

For the sake of readability, all proofs for the formal results are placed at the end of this chapter in Section 3.9.

Algorithm 3.1 PLATE loop

$\tau_s \leftarrow 0, k \leftarrow 0$
for sampling event $t = \tau_s$ **do**
  # Estimation stage:
  Use $\mathbf{x}_0$ and $\{\mathbf{z}[0], \dots, \mathbf{z}[k-1]\}$ to compute $\hat{\mathbf{x}}[k]$.
  Read raw data from sensors.
  # Perception latency scheduling:
  Decide which perception method $p_k \in \{1, \dots, D\}$ to use.
  Wait Δ^{p_k} units of time until the perception-latency has elapsed to produce $\mathbf{z}[k]$, i.e. $\tau_s \leftarrow \tau_s + \Delta^{p_k}$.
  $k \leftarrow k + 1$
end for

3.3 PERCEPTION-LATENCY AWARE ESTIMATION

In this section, we establish the structure of the estimation stage of PLATE. To do so, and in order to study the latency-precision trade-off under a cost of the form (3.2), it is useful to study an equivalent model of (3.1) as a sampled-data system. The following result follows from [130, Section 4.5.2]:

Proposition 3.3. *Consider a perception schedule p. Hence, the solution $\mathbf{x}(t)$ of (3.1) satisfy:*

$$\mathbf{x}(t) = \mathbf{A}_{\mathsf{d}}(t - \tau_k)\mathbf{x}[k] + \mathbf{w}_d(t) \tag{3.3a}$$
$$\mathbf{x}[k+1] = \mathbf{A}_{\mathsf{d}}(\Delta^{p_k})\mathbf{x}[k] + \mathbf{w}_d[k] \tag{3.3b}$$

for $t \in [\tau_k, \tau_{k+1}), k \geq 0$ where $\mathbf{A}_{\mathsf{d}}(t-\tau_k) := \exp(\mathbf{A}(t-\tau_k))$ and $\mathbf{w}_d(t)$ normally distributed with $\mathsf{cov}\{\mathbf{w}_d(t)\}$ given as

$$\mathbf{W}_d(t - \tau_k) = \int_0^{t-\tau_k} \mathbf{A}_{\mathsf{d}}(\tau)\mathbf{B}\mathbf{W}\mathbf{B}^\top \mathbf{A}_{\mathsf{d}}(\tau)^\top \mathrm{d}\tau$$

The discrete-time nature of the perception process is made explicit by (3.3), which is equivalent to (3.1). As a result, we adopt (3.3) as the motion model throughout the manuscript.

The following result shows the computation of an optimal estimation $\hat{\mathbf{x}}(t)$ or $\mathbf{x}(t)$ provided an optimal schedule p.

Theorem 3.4. *Let p be the optimal schedule for Problem 3.1. Thus, the optimal estimation for (3.1) at $t \in [\tau_k, \tau_{k+1})$ is given by the conditional mean $\mathbb{E}\left\{\mathbf{x}(t) \mid \mathbf{z}[0], \dots, \mathbf{z}[k-1]\right\}$ and can be computed as:*

$$\begin{aligned} \hat{\mathbf{x}}(t) &= \mathbf{A}_{\mathsf{d}}(t - \tau_k)\hat{\mathbf{x}}[k] \\ \hat{\mathbf{P}}(t) &= \mathsf{cov}\{\mathbf{x}(t) - \hat{\mathbf{x}}(t)\} = \mathbf{A}_{\mathsf{d}}(t - \tau_k)\hat{\mathbf{P}}[k]\mathbf{A}_{\mathsf{d}}(t - \tau_k)^\top + \mathbf{W}_d(t - \tau_k) \end{aligned} \tag{3.4}$$

for $t \in [\tau_k, \tau_{k+1})$ *with* $\hat{\mathbf{x}}[k], \hat{\mathbf{P}}[k]$ *generated according to*

$$\begin{aligned}
\hat{\mathbf{x}}[0] &= \mathbf{x}_0,\ \hat{\mathbf{P}}[0] = \mathbf{P}_0 \\
\mathbf{L}[k] &= \mathbf{A}_{\mathtt{d}}(\Delta^{p_k})\hat{\mathbf{P}}[k]\mathbf{C}^\top \left(\mathbf{C}\hat{\mathbf{P}}[k]\mathbf{C}^\top + \mathbf{R}^{p_k}\right)^{-1} \\
\hat{\mathbf{x}}[k+1] &= \mathbf{A}_{\mathtt{d}}(\Delta^{p_k})\hat{\mathbf{x}}[k] + \mathbf{L}[k]\left(\mathbf{z}[k] - \mathbf{C}\hat{\mathbf{x}}[k]\right) \\
\hat{\mathbf{P}}[k+1] &= \left(\mathbf{A}_{\mathtt{d}}(\Delta^{p_k}) - \mathbf{L}[k]\mathbf{C}\right)\hat{\mathbf{P}}[k]\left(\mathbf{A}_{\mathtt{d}}(\Delta^{p_k}) - \mathbf{L}[k]\mathbf{C}\right)^\top + \mathbf{L}[k]\mathbf{R}^{p_k}\mathbf{L}[k]^\top + \mathbf{W}_d(\Delta^{p_k})
\end{aligned} \tag{3.5}$$

3.4 SCHEDULING POLICY

Now that the estimation stage of PLATE has been established, we turn our attention to obtaining the optimal scheduling policy, minimizing (3.2). In contrast to the estimations $\hat{\mathbf{x}}(t)$ which may take arbitrary values in $\mathbb{R}^{n\times}$, the perception schedule p is of discrete nature.

One possible way to obtain the optimal schedule that solves Problem 3.1 is to use Algorithm 3.2 by calling $\mathtt{dynProg}(0, \hat{\mathbf{P}}[0], T_f)$. This algorithm is essentially an exhaustive search for the optimal schedule, organized as a dynamic programming recursive algorithm.

Figure 3.2 illustrates how our dynamic programming algorithm works. The algorithm starts at time $\tau^+ = 0$ when calling $\mathtt{dynProg}(0, \hat{\mathbf{P}}[0], T_f)$ and tries out each of the D possible perception methods. For each method, it computes a cost-to-arrive J (line 6 of Algorithm 3.2) and checks whether the current schedule length τ^+ is less than the desired length T_f. If $\tau^+ < T_f$, the algorithm must continue exploring other scheduling options that may extend the current schedule. This process is represented by the black nodes in Figure 3.2. On the other hand, if $\tau^+ \geq T_f$, the algorithm has found a valid schedule that covers the entire time window $[0, T_f]$. This is represented by the white nodes in the figure.

At each step of the algorithm, a recursive call to the function is made using the current schedule length τ^+ and the updated covariance state $\hat{\mathbf{P}}[k+1]$. The recursion continues until the algorithm reaches the end of the time window T_f. At that point, the algorithm returns the optimal schedule decision p_k and the final cost for that schedule.

The correctness and complexity of the algorithm are established in the following result:

Proposition 3.5. *Calling*

$$\{p, J\} \leftarrow \mathtt{dynProg}(0, \hat{\mathbf{P}}[0], T_f)$$

and computing $\hat{\mathbf{x}}(t)$ *using the PLATE structure in* (3.4) *for such* p*, results in the optimal estimations, schedule and cost* $\hat{\mathbf{x}}, p, \mathcal{J}(\hat{\mathbf{x}}, p) \equiv J$ *for Problem 3.1 with worst case complexity given by* $O\left(D^{\alpha_{\max}}\right)$ *where*

$$\alpha_{\max} := \lfloor T_f / \min\{\Delta^1, \ldots, \Delta^D\} \rfloor.$$

The previous result evidences some important complications of Problem 3.1. First, as shown in Proposition 3.2, the complexity of the exact solution in Algorithm 3.2 increases exponentially as T_f increases. This is not surprising, since the discrete nature of the perception schedule p suggests that the problem is subject to a combinatorial explosion. Moreover, due to the transient behaviour of $\hat{\mathbf{P}}(t)$, choosing p_k just to minimize a one step

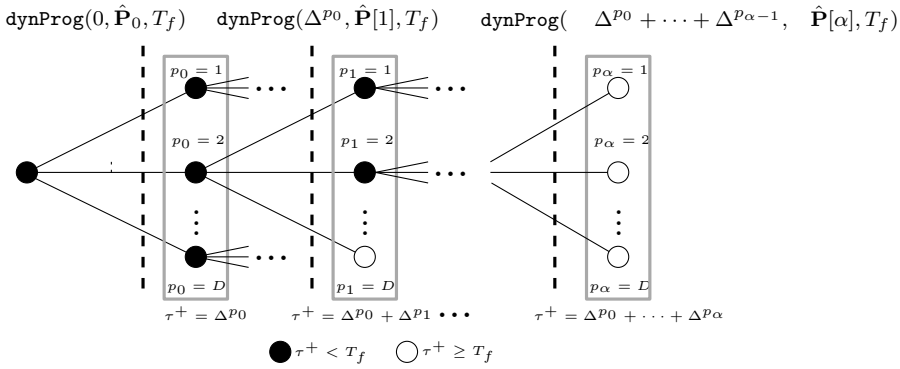

Figure 3.2: Graphical depiction of Algorithm 3.2. At each step, all scheduling method options are explored in a recursive fashion with a maximum depth until the condition $\sum_{k=0}^{\alpha} \Delta p_k \geq T_f$ where $\alpha = \mathsf{att}(p; [0, T_f])$ is reached.

Algorithm 3.2 `dynProg`

Input: $\tau, \hat{\mathbf{P}}[k], T_f$
Output: p, J

1: $J \leftarrow \infty$
2: $p \leftarrow \varnothing$
3: **for** $\rho \in \{1, \ldots, D\}$ **do**
4: $\quad \tau^+ \leftarrow \tau + \Delta^\rho$
5: $\quad p^+ \leftarrow \varnothing$
6: $\quad J_\rho \leftarrow \frac{1}{T_f}\left(\lambda_\alpha r^\rho + \int_\tau^{\min(\tau^+, T_f)} \mathsf{tr}(\mathbf{W}_d(t-\tau))\mathrm{d}t + \int_\tau^{\min(\tau^+, T_f)} \mathsf{tr}(\mathbf{A}_\mathsf{d}(t-\tau)\hat{\mathbf{P}}[k]\mathbf{A}_\mathsf{d}(t-\tau)^\top)\mathrm{d}t\right)$
7: $\quad$ **if** $\tau^+ < T_f$ **then**
8: $\quad\quad$ Compute $\hat{\mathbf{P}}[k+1]$ with (3.5) for ρ and $\hat{\mathbf{P}}[k]$
9: $\quad\quad$ $\{p^+, J^+\} \leftarrow$dynProg$(\tau^+, \hat{\mathbf{P}}[k+1], T_f)$
10: $\quad\quad$ $J_\rho \leftarrow J_\rho + J^+$
11: $\quad$ **end if**
12: $\quad$ **if** $J_\rho < J$ **then**
13: $\quad\quad$ $J \leftarrow J_\rho$
14: $\quad\quad$ Append ρ to the start of p^+ and assign the result to p
15: $\quad$ **end if**
16: **end for**

ahead of the cost in line 6 of `dynProg` may not be sufficient to find an exact solution for the problem without having to explore future scheduling method options over the window $[0, T_f]$. Some performance improvements can be made to Algorithm 3.2 such as applying a branch-and-bound technique [124, Chapter 2.3.3] or general approximations or heuristics [124, Chapter 6]. However, seeking for tailor-made approximate solution

and studying its performance gap with respect to the optimal is more appropriate for a practical implementation.

3.4.1 Quantized covariance approach

We provide an approximate alternative to Algorithm 3.2 in order to make it computationally feasible for use in PLATE. This approximation is sub-optimal, but can be made as close to optimal as desired, jeopardizing the computational complexity. The key idea is to observe that as the window $[0, T_f]$ is traversed during Algorithm 3.2, the set of possible states of the covariance $\hat{\mathbf{P}}[k]$ increase exponentially in size as well. This is evident from Figure 3.2 which shows how the nodes at each level of the exploration tree increase exponentially as all possible combinations of perception methods are tested. However, if the amount of nodes at each level of the tree in Figure 3.2 is managed to keep a bound regardless of T_f, the complexity of the algorithm can be reduced. This can be performed by grouping similar values of $\hat{\mathbf{P}}[k]$ into a single element of $\mathbb{R}^{n_\star \times n_\star}$ through quantization. The result is a compressed exploration graph with a non-growing number of nodes per level.

The quantization procedure for the covariance values is described as follows. Consider a compact set $\mathcal{B}_0 \subset \mathbb{R}^{n_\star \times n_\star}$ containing only positive semi-definite matrices, characterized by a bound $B_0 > 0$ such that

$$\|\hat{\mathbf{P}}[k]\|_F \leq B_0, \forall \hat{\mathbf{P}}[k] \in \mathcal{B}_0.$$

Now, given some $\delta > 0$, partition $\mathcal{B}_0$ into $Q_0(\delta) \in \mathbb{N}$ non-overlapping regions $\mathcal{B}_1, \ldots, \mathcal{B}_{Q_0(\delta)}$ such that

$$\sup_{\hat{\mathbf{P}}'[k], \hat{\mathbf{P}}''[k] \in \mathcal{B}_q} \|\hat{\mathbf{P}}'[k] - \hat{\mathbf{P}}''[k]\|_F \leq \delta, \forall q \in \{1, \ldots, Q_0(\delta)\}$$

and

$$\bigcup_{q=1}^{Q_0(\delta)} \mathcal{B}_q = \mathcal{B}_0$$

. Moreover, pick a representative $\hat{\mathbf{P}}_q \in \mathcal{B}_q$ as an identifier for all other $\hat{\mathbf{P}}[k] \in \mathcal{B}_q$. Finally, let the quantization function

$$\mathcal{Q} : \mathcal{B}_0 \to \{\hat{\mathbf{P}}_1, \ldots, \hat{\mathbf{P}}_{Q_0(\delta)}\}$$

which takes any $\hat{\mathbf{P}}[k] \in \mathcal{B}_0$ and maps it to the $\hat{\mathbf{P}}_q$ such that $\hat{\mathbf{P}}[k] \in \mathcal{B}_q$. Note that we do not require to use a uniform quantization procedure. In fact, for practical purposes it is convenient to use a non-uniform quantization scheme in which $Q_0(\delta)$ points $\hat{\mathbf{P}}_1, \ldots, \hat{\mathbf{P}}_{Q_0(\delta)} \in \mathcal{B}_0$ are provided instead, e.g. sampled from $\mathcal{B}_0$, from which the maximum distance $\delta > 0$ is obtained.

The idea is to track how the identifiers $\hat{\mathbf{P}}_q$ with $q \in \{1, \ldots, Q_0(\delta)\}$ are related between them using the PLATE estimation stage evolution in (3.5). To do so, a weighted directed graph $(\mathcal{V}, \mathcal{E}, \rho)$ is constructed with vertex set $\mathcal{V} = \{1, \ldots, Q_0(\delta)\}$. Moreover, each edge $e = (i, j) \in \mathcal{E}$ corresponds to a connection between $i, j \in \{1, \ldots, Q_0(\delta)\}$ when $\hat{\mathbf{P}}_j = \mathcal{Q}(\hat{\mathbf{P}}[k+1])$ with $\hat{\mathbf{P}}[k+1]$ computed using (3.5) for $\hat{\mathbf{P}}[k] = \hat{\mathbf{P}}_i$ and a weight function $\rho : \mathcal{E} \to \{1, \ldots, D\}$ with $\rho(e) = p_k$.

However, it might be the case that $\hat{\mathbf{P}}[k+1]$ lies outside of $\mathcal{B}_0$ for some p_k and $\hat{\mathbf{P}}[k] = \hat{\mathbf{P}}_q$, $q \in \{1, \dots, Q_0(\delta)\}$. To account for these cases, we apply the steps described in Algorithm 3.3 adding new states outside $\mathcal{B}_0$ as required. The result is a graph $\mathcal{G}$ with $Q(\delta) \geq Q_0(\delta)$ states where $\hat{\mathbf{P}}_q \in \mathcal{B}, \forall q \in \{1, \dots, Q(\delta)\}$ and $\mathcal{B}$ is a region of $\mathbb{R}^{n_{\mathbf{x}} \times n_{\mathbf{x}}}$ with $\mathcal{B}_0 \subset \mathcal{B}$, and its corresponding bound $\|\hat{\mathbf{P}}[k]\|_F \leq B, \forall \hat{\mathbf{P}}[k] \in \mathcal{B}$. When Algorithm 3.3 finishes, it is ensured that if $\hat{\mathbf{P}}[0] \in \mathcal{B}_0$, then the quantized covariance trajectories will be contained in $\mathcal{B}$ for any perception schedule.

Algorithm 3.3 `expandB`

Input: $\delta, \mathcal{B}_0$ with $\|\hat{\mathbf{P}}[0]\| \leq B_0$
Output: $\mathcal{B}, \mathcal{G}$

```
Quantize B_0 into Q_0(δ) patches {B_1, ..., B_{Q_0(δ)}} with identifiers {P̂_1, ..., P̂_{Q_0(δ)}}
q ← 0
V ← {1, ..., Q_0(δ)}
E ← ∅
B ← B_0
Q(δ) ← Q_0(δ)
while q ≤ Q(δ) do
  for p ∈ {1, ..., D} do
    Compute P̂[k+1] from (3.5) using P̂[k] = P̂_q for p_k = p
    if P̂_{q'} ≠ Q(P̂[k+1]) for any q' ∈ {1, ..., Q(δ)} then
      Q(δ) ← Q(δ) + 1
      V ← V ∪ {Q(δ)}
      Create a new patch B_{Q(δ)} and identifier P̂_{Q(δ)} such that P̂_{Q(δ)}, Q(P̂[k+1]) ∈
      B_{Q(δ)} and sup_{P̂,P̂'∈B_{Q(δ)}} ||P̂ − P̂'|| = δ
      B ← B ∪ B_{Q(δ)}
    end if
    Add edge e = (q, q') to E with weight ρ(e) = p
  end for
  q ← q + 1
end while
Construct G using V, E and its weights from line 16.
```

It is beneficial for practical purposes, to estimate how $Q(\delta)$ grows with respect to original size of $\mathcal{B}_0$ as a result of Algorithm 3.3. The following result ensures that Algorithm 3.3 finishes with finite $Q(\delta)$. In addition, we provide an explicit upper bound for B, from which a worst case of $Q(\delta)$ can be computed depending on the actual quantization scheme.

Proposition 3.6. *Algorithm 3.3 finishes for any compact set $\mathcal{B}_0 \subset \mathbb{R}^{n \times n}$ and $\delta > 0$. In addition, the resulting bound B for $\mathcal{B}$ complies*

$$B \leq B_s := \sqrt{n_{\mathbf{x}}} \left(\frac{\lambda_{\max}(\mathbf{\Omega})}{\lambda_{\min}(\mathbf{\Omega})} \right) \left(B_0 + \frac{\overline{G}}{1-\gamma} \right) \tag{3.6}$$

where $\mathbf{\Omega} \in \mathbb{R}^{n_{\mathbf{x}} \times n_{\mathbf{x}}}$ and $\mathbf{Y}^i \in \mathbb{R}^{n_{\mathbf{x}} \times n_{\mathbf{z}}}, i \in \{1, \dots, D\}$ satisfy the following Linear Matrix

Inequality (LMI):

$$\begin{bmatrix} \gamma\mathbf{\Omega} & (\mathbf{\Omega}\mathbf{A}_{\mathsf{d}}(\Delta^i) - \mathbf{Y}^i\mathbf{C})^\top \\ (\mathbf{\Omega}\mathbf{A}_{\mathsf{d}}(\Delta^i) - \mathbf{Y}^i\mathbf{C}) & \mathbf{\Omega} \end{bmatrix} \succeq 0, \quad i = 1, \ldots, D, \quad \mathbf{\Omega} \succ 0 \tag{3.7}$$

for some $0 < \gamma < 1$. *Moreover,* $\lambda_{\min}(\mathbf{\Omega}), \lambda_{\max}(\mathbf{\Omega})$ *are the minimum and maximum eigenvalues of* $\mathbf{\Omega}$ *respectively and*

$$\overline{G} := \max_{i \in \{1,\ldots,D\}} \|(\mathbf{L}^{p_k})\mathbf{R}^{p_k}(\mathbf{L}^{p_k})^\top + \mathbf{W}_d(\Delta^{p_k})\|_F$$

with $\mathbf{L}^{p_k} = \mathbf{\Omega}^{-1}\mathbf{Y}^{p_k}$.

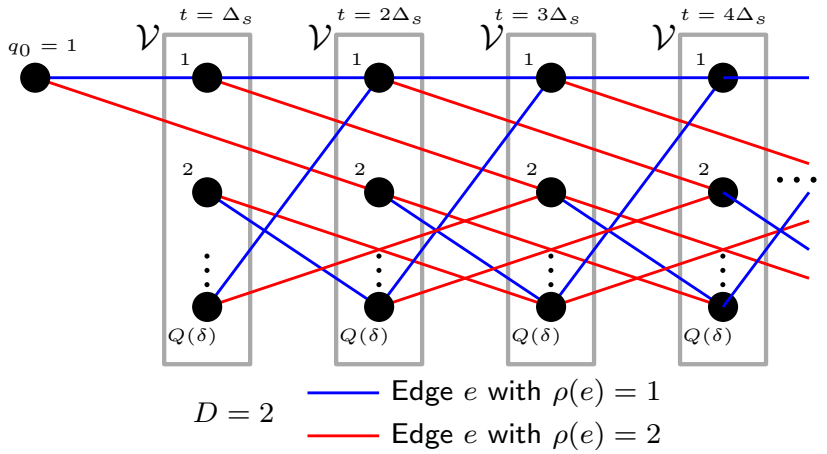

Figure 3.3: Transition graph example for covariance evolution using $\mathcal{G}$. In this example $D = 2$. Each node labeled in $1, \ldots, Q(\delta)$ is connected to other nodes through one of the two perception decisions, either one or two steps ahead since $\Delta^1 = \Delta_s, \Delta^2 = 2\Delta_s$ for this example.

Using the graph $\mathcal{G}$, the evolution of covariance states in $\mathcal{V}$ can be tracked from an initial condition $q_0 \in \mathcal{V}$ as time advances in $[0, T_f]$. This can be visualized using a transition diagram, as shown in Figure 3.3 which depicts an example with $D = 2$ and $\Delta^1 = \Delta_s, \Delta^2 = 2\Delta_s$. Here, at $t = 0$ the initial state is q_0 corresponding to $\hat{\mathbf{P}}_{q_0} = \mathcal{Q}(P[0])$. We consider a discrete time $t = \ell\Delta_s$ as integer multiples of the minimum sampling interval. As ℓ increases, edges $e \in \mathcal{E}$ connect states between time steps, with different perception decisions $\rho(e)$. Note that, under this perspective, number of states at each time step is maintained constant. Hence, a dynamic programming algorithm for finite-state deterministic systems can be applied in order to obtain the optimal schedule p [124, Page 64]. In this context, the optimal p corresponds to the shortest route starting at q_0, which traverses the transition diagram of $\mathcal{G}$ until it reaches any state at stage $\ell = \lfloor T_f/\Delta_s \rfloor$ with distance measured by $\mathcal{J}(\hat{\mathbf{x}}, p)$.

The concrete steps required to run this dynamic programming solution, called `qDP`, are described in Algorithm 3.4 and are discussed as follows. First, auxiliary matrices

$\mathbf{M}_Q, \mathbf{M}_P, \mathbf{M}_J$ are computed from Algorithm 3.5 given an initial state q_0. In line 6 of Algorithm 3.5 time steps up to $\alpha_{\max}$ are traversed. Moreover, in line 7 all possible states q at time ℓ are evaluated as well. Furthermore, for each of these states q, in line 8 all perception options ρ are checked. In this way, Algorithm 3.5 tracks optimal routes where $[\mathbf{M}_J]_{q,\ell}$ stores the best cost-to-arrive from q_0 to q after ℓ time steps. Moreover, $[\mathbf{M}_Q]_{q',\ell}$ stores the best state q connected to q' at time step ℓ, where the perception decision is stored in $[\mathbf{M}_P]_{q',\ell}$ defining how many stages separate both states as well. These matrices are used in Algorithm 3.4 to trace-back the optimal scheduling.

Algorithm 3.4 `qDP`

Input: $q_0, T_f, \mathcal{G}$ computed from Algorithm 3.3
Output: p, J

```
{M_Q, M_P, M_J} ← qDPMatrices (q_0, T_f, G)
Set J and q to the best cost and state at the last column of M_J
p ← ∅
# Use M_P to trace back the optimal schedule ending at state q
ℓ ← ⌊T_f/Δ_s⌋
while ℓ > 0 do
  ρ ← [M_P]_{q,ℓ}
  Append ρ at the start of p
  q ← [M_Q]_{q,ℓ}
  ℓ ← ℓ − Δ^ρ/Δ_s
end while
```

The sense in which we verify correctness of the dynamic programming as from Algorithm 3.4 and its complexity is established in the following:

Proposition 3.7. *Consider the* `qDP` *algorithm and assume that the evolution of* $\hat{\mathbf{P}}[k]$ *is constrained to evolve according to the structure in* $\mathcal{G}$ *and that* $\hat{\mathbf{P}}[0] \in \{\hat{\mathbf{P}}_1, \ldots, \hat{\mathbf{P}}_{Q(\delta)}\}$. *Thus,* `qDP` *obtains the optimal value of* (3.2) *for such constrained trajectories. Moreover, the worst-case of complexity of* `qDP` *is* $O(\alpha_{\max} Q(\delta) D)$ *where* $\alpha_{\max} = \lfloor T_f/\Delta_s \rfloor$.

Proof: The proof can be found in Section 3.9.4. □

As evidenced by the previous result, the complexity of the quantized covariance approach is reduced to something linear in T_f. Nonetheless, since this is an approximate solution, there will be a trade-off between complexity and the performance gap of the quantized solution with respect to the true optimal. Note that by decreasing δ, the number of patches $\mathcal{B}_q$ needed to cover $\mathcal{B}$ will increase. However, it is expected that as $\delta \to 0$, the resulting sub-optimal solution perfomance improves. These ideas are formalized in the following result.

Theorem 3.8. *Let* $\mathcal{J}$ *be the optimal cost for Problem 3.1 and* $\mathbf{P}[0] \in \mathcal{B}_0$. *Then, for any* $\varepsilon > 0$ *there exists sufficiently small* $\delta > 0$ *such that*

$$|\mathcal{J} - \mathcal{J}_{\mathcal{Q}}(\delta)| \leq \varepsilon$$

where $\mathcal{J}_{\mathcal{Q}}(\delta)$ *is the cost obtained from* `qDP` *in Algorithm 3.4 for such* δ *and initial condition* $\mathcal{Q}(\hat{\mathbf{P}}[0]) \in \mathcal{B}_0$.

Algorithm 3.5 `qDPMatrices`

Input: $q_0, T_f, \mathcal{G}$
Output: $\mathbf{M}_Q, \mathbf{M}_P, \mathbf{M}_J$

$\alpha_{\max} \leftarrow \lfloor T_f/\Delta_s \rfloor$
$\mathbf{M}_Q \leftarrow [0] \in \{0, 1, \ldots, Q(\delta)\}^{Q(\delta)\times(\alpha_{\max}+1)}$
\# $[\mathbf{M}_Q]_{q',\ell}$: best state q connected to q' at step ℓ
$\mathbf{M}_P \leftarrow [1] \in \{0, 1, \ldots, D\}^{D\times(\alpha_{\max}+1)}$ \# $[\mathbf{M}_P]_{q',\ell}$: Perception connecting best q connected to q' at step ℓ
$\mathbf{M}_J \leftarrow [\infty] \in \bar{\mathbb{R}}^{Q(\delta)\times(\alpha_{\max}+1)}$ \# $[\mathbf{M}_J]_{q,\ell}$: best cost from q_0 to q in ℓ steps
$[\mathbf{M}_J]_{q_0,0} \leftarrow 0$
for $\ell \in \{1, \ldots, \alpha_{\max}\}$ **do**
  for $q \in \{1, \ldots, Q(\delta)\}$ **do**
    for $\rho \in \{1, \ldots, D\}$ **do**
      $q' \leftarrow$ state connected to q through edge with weight ρ
      $\tau \leftarrow (\ell - 1)\Delta_s$
      $\tau^+ \leftarrow \tau + \Delta^\rho$
      $J \leftarrow \frac{1}{T_f}\Bigg(\lambda_\alpha r^\rho + \int_\tau^{\min(\tau^+, T_f)} \mathbf{W}_d(t-\tau)\mathrm{d}t +$
      $\int_\tau^{\min(\tau^+, T_f)} \mathsf{tr}(\mathbf{A}_\mathsf{d}(t-\tau)\hat{\mathbf{P}}_q\mathbf{A}_\mathsf{d}(t-\tau)^\top)\mathrm{d}t\Bigg)$
      \# $[\mathbf{M}_J]_{q,\ell} + J$ is the cost-to-arrive from q_0 to q' in $\ell + \rho$ steps
      if $[\mathbf{M}_J]_{q,\ell} + J < [\mathbf{M}_J]_{q',\ell+\rho}$ **then**
        $[\mathbf{M}_J]_{q',\ell+\rho} \leftarrow [\mathbf{M}_J]_{q,\ell}$
        $[\mathbf{M}_Q]_{q',\ell+\rho} \leftarrow q$
        $[\mathbf{M}_P]_{q',\ell+\rho} \leftarrow \rho$
      end if
    end for
  end for
end for

Henceforth, we have established that the PLATE strategy with estimation stage in (3.5) and perception-scheduling policy obtained using `qDP` is a sub-optimal for Problem 3.1, with mild computational complexity in terms of the window size T_f and can approximate the optimal solution with arbitrary precision if more computational power is available.

Remark 3.9. *It is important to note that only the perception schedule computation is impacted by quantization, while the actual estimator in* (3.4) *and* (3.5) *is not quantized. Therefore, the quantization step δ only affects the resulting quality of the perception schedules in terms of the cost* (3.2)*. We provide an analysis of the asymptotic effect of quantization on the cost in Theorem 3.8. While decreasing δ is expected to increase the quality of the perception schedules, it is not obvious whether the performance can be made arbitrarily close to the optimal one. However, this accuracy feature is ensured by Theorem 3.8. Obtaining closed-form guarantees for a given performance for a value of $\delta > 0$ is challenging and will be explored in future work.*

Remark 3.10. *Our approach can be extended to filter structures different than* (3.5) *such as a particle filter, as long as a similar transition graph for the covariance matrix is obtained for its use in Algorithm 3.3. However, obtaining similar theoretical guarantees as in this chapter for other filter structures is not a trivial task and require more in depth analysis.*

3.5 MOVING HORIZON PLATE

One property of the qDP method used in PLATE is that given an initial condition $\hat{\mathbf{P}}[0]$, a schedule p for the whole time window $[0, T_f]$ is obtained. Thus, it only suffices to compute the schedule at the beginning and traverse it element by element in line 5 of Algorithm 3.1 after each perception method is used. However, in a more practical target tracking scenario, there will be missing measurements as a result of occlusions, or distinguishability problems between targets in a multi-target setting. An appropriate re-detection of the targets may be performed by a maintenance mechanism as widely discussed in the literature. This may alter the quality of the pre-computed perception schedule.

In addition, the processed measurement quality, represented by the covariances $\mathbf{R}^1$, $\ldots$, $\mathbf{R}^D$ may not remain fixed in a practical setting. Instead, it is usual to use the current estimation for the state of the target to improve the quality of subsequent measurements as in [53, 131, 132]. Moreover, the detection method may include some measure of uncertainty of their current output, e.g. through Bayesian techniques [133], or data-driven methods [134]. Thus, an online covariance estimate $\mathbf{R}[k]$ for the processed measurements may be available and may not be any of the nominal covariances. This discussion motivates to change the perception schedule according to the current state of the system by using qDP as a predictive policy.

To do so, we propose to use moving-horizon scheme for PLATE in the following way. First, we construct the transition graph $\mathcal{G}$ as described in the previous section for nominal covariances $\{\mathbf{R}^1, \ldots, \mathbf{R}^D\}$. Now, the dynamic programming algorithm qDP is used at τ_k with initial condition given by the current $q_0 = \mathcal{Q}(\hat{\mathbf{P}}[k])$ to obtain a perception schedule $p' = \texttt{qDP}(q_0, T_f, \mathcal{G})$ for the next window $[\tau_k, \tau_k + T_f]$. As a result, we apply the first perception method of p' during $[\tau_k, \tau_k + \Delta^{p_k})$ and repeat the procedure for the next interval.

Furthermore, every time a new perception output is obtained, the PLATE correction in (3.5) is computed with the actual processed measurement uncertainty $\mathbf{R}[k]$ if available, otherwise with its nominal covariance. When there are no measurements, we simply predict the state of the target and its covariance using (3.4). This procedure is summarized in Algorithm 3.6 which describes the steps that must be performed in line 5 of Algorithm 3.1 in order to implement this strategy.

Moreover, the latency of executing qDP at each τ_k can be considered negligible if two of the following approaches is used. First, since $\hat{\mathbf{P}}[k]$ is quantized through $\mathcal{Q}(\bullet)$ there are only $Q(\delta)$ possible outcomes for which their resulting perception method obtained through qDP can be pre-computed offline. On the other hand, similarly to what is suggested in [8], taking $\hat{\mathbf{P}}[k]$ and converting it to a perception method in $\{1, \ldots, D\}$ is a classification problem with a low dimensional input. Thus, the perception decision can be learned by a classifier.

Algorithm 3.6 Moving-horizon PLATE

 # Offline:
Pre-compute $\mathcal{G}$ using Algorithm 3.3, alternatively pre-compute $\texttt{qDP}(\mathcal{Q}(\hat{\mathbf{P}}_q, \mathcal{G}, T_f)$ for all $\hat{\mathbf{P}}_q, q \in \{1, \ldots, Q(\delta)\}$.
 # Online:
if There is new processed measurement **then**
  Update $\hat{\mathbf{P}}[k]$ using (3.5) for the previous scheduling decision p_{k-1} and its resulting perception uncertainty $\mathbf{R}[k-1]$.
else
  Predict $\hat{\mathbf{P}}[k]$ using (3.4) for the previous scheduling decision p_{k-1}.
end if
$p', \_ \leftarrow \texttt{qDP}(\mathcal{Q}(\hat{\mathbf{P}}[k]), \mathcal{G}, T_f)$
$p_k \leftarrow$ first element of p'

Remark 3.11. *Comparing our approach with previous work, note that the perception scheduling policy as in Algorithm 3.6 acts as a frame-skipping technique similar to [60, 73, 135]. However, unlike prior frame-skipping approaches which are based on heuristic rules, we provide an actual performance guarantees for our sub-optimal perception schedules as given in Theorem 3.8. On the other hand, note that unlike related multi-sensor scheduling approaches in the literature as in [77–83], Algorithm 3.4 can be extended for different cost function expressions by modifying line 12 of Algorithm 3.5, which makes our proposal more versatile.*

Remark 3.12. *The moving horizon PLATE proposal allows us to change the system parameters online as in line 3 of Algorithm 3.6, where we use an uncertainty* $\mathbf{R}[k-1]$ *which need not to be any of the nominal ones for the perception methods. This poses an important advantage with respect to the philosophy of some multi-sensor scheduling works such as [77, 78] which rely on periodic schedules or other strategies that are highly dependant on the problem structure.*

3.6 NUMERICAL EXPERIMENTS

In order to evaluate PLATE we consider the following scenario. First, let a target with sate $\mathbf{x} = [x, v_x, y, v_y]^\top$ describing its position and velocity on the plane. Let camera images available every $\Delta_s = 1/30$ seconds, coinciding with the usual frame rate of a camera. In addition, the position $[x, y]^\top$ is measured through a detection process from which there are $D = 2$ available perception configuration. First, a lightweight detector with latency $\Delta^1 = 3\Delta_s$ and nominal covariance $\mathbf{R}^1 = \mathsf{diag}(0.5, 0.5)$. Second, a more accurate detector with latency $\Delta^2 = 9\Delta_s$ and nominal covariance $\mathbf{R}^2 = \mathsf{diag}(0.05, 0.05)$. We consider CPU loads of $f^1 = 0.5$ and $f^2 = 0.8$ resulting in penalties $r^1 = 0.5\Delta^1$ and $r^2 = 0.8\Delta^2$. The system matrices in this scenario are:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

In addition, let the process covariance $\mathbf{W} = \mathsf{diag}(0.5, 0.5)$. This double integrator target model is often used for generic targets in the literature. In the following, we evaluate the different aspects of our proposals for this setting.

3.6.1 Numerical covariance bound estimation

We start by building a transition graph $\mathcal{G}$. To do so, we set $\mathcal{B}_0 \in \mathbb{R}^{n\times n}$ as the set of all positive definite matrices $\mathbf{P}$ with $\|\mathbf{P}\|_F \leq \mathbf{B}_0 := 1$ and use the mechanism in Algorithm 3.3. In order to depict the result in Proposition 3.6, we explicitly obtain a bound for $\mathcal{B}$. First, note that the LMI (3.7) has a solution for $\gamma = 0.98$ as

$$\mathbf{Y}^1 = \begin{bmatrix} 0.031 & 0 \\ 0.037 & 0 \\ 0 & 0.031 \\ 0 & 0.037 \end{bmatrix}, \mathbf{Y}^2 = \begin{bmatrix} 0.122 & 0 \\ 0.137 & 0 \\ 0 & 0.122 \\ 0 & 0.137 \end{bmatrix}$$

with a resulting bound (3.6) of $B_s = 4.922$. Figure 3.4 depicts how given initial conditions $P[0] \in \mathcal{B}_0$ and a randomly generated perception schedule, the magnitude of the PLATE covariance complies $\|\hat{\mathbf{P}}[k]\|_F \leq B_s, \forall k \geq 0$.

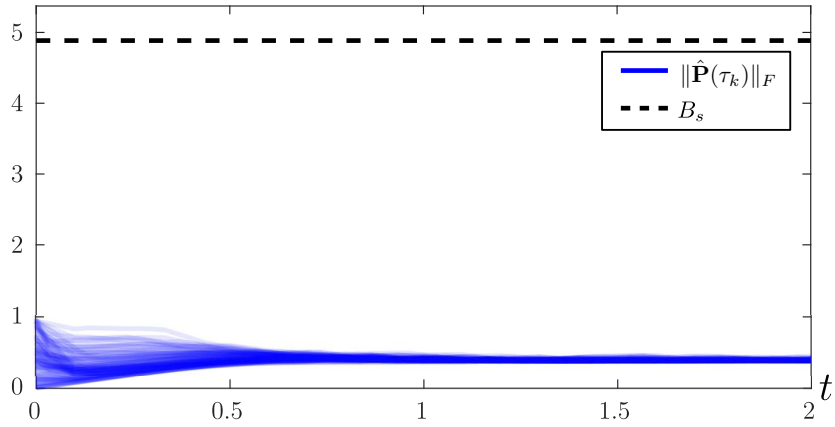

Figure 3.4: Covariance norm evolution, showing the bound B_s as well as $\|\hat{\mathbf{P}}(\tau_k)\|$ for PLATE with 100 random initial conditions $\hat{\mathbf{P}}[0]$ and schedules p with $\|\hat{\mathbf{P}}[0]\|_F \leq B_0 = 1$.

3.6.2 Cost comparison using `qDP`

Now, in order to depict Theorem 3.8 through this example, we use different levels of quantization for evaluation. Note that the results presented until now don't require uniform quantization. In fact, in the following examples for simplicity we use non-uniform quantization by sampling a fixed number of points $\hat{\mathbf{P}}_1, \ldots, \hat{\mathbf{P}}_{Q(\delta)} \in \mathcal{B}_0$ and quantizing any other $\hat{\mathbf{P}} \in \mathcal{B}_0$ through a nearest neighbor rule under the $\|\bullet\|_F$ norm. Next, we expand $\mathcal{G}$ through Algorithm 3.3 to obtain $\mathcal{B}$. We use $Q(\delta) = 50, 500, 5000$ which after this process results in $\delta = 10.45, 4.23, 2.14$ respectively for these arbitrarily selected points, in order to analyze the impact of the quantization step in the tracking quality.

We evaluate qDP for 100 random samples for $P[0]$ over $\mathcal{B}$ and compute their resulting cost (3.2). First, consider $T_f = 1, \lambda_\alpha = 5$. In this case, it is possible to obtain the minimum cost $J_{\min}$ by evaluating all schedules covering the window $[0, T_f]$. We show the results in the first row of Figure 3.5 where the histograms depict the frequency of the distance $|J_{\min} - J_p|$ where J_p is the cost obtained for a schedule p. The schedules tested where obtained through qDP for graphs $\mathcal{G}$ using the previously described values of $Q(\delta)$, as well as the static schedules $p = \{1, 1, \dots\}$ and $p = \{2, 2, \dots\}$. The results show that as the quanitization gets finer, the cost concentrate more and more towards the minimum, i.e. $|J_{\min} - J_p|$ becomes smaller. In addition, it is observed that even with a coarse discretization $Q(\delta) = 50$, the cost of qDP is almost always less than the cost for the static schedules. Moreover, the average CPU load seems to decrease as well as the discretization gets finer. A similar behaviour is obtained when using $T_f = 10, \lambda_\alpha = 100$, as depicted in the second row of Figure 3.5. In this case, $J_{\min}$ is computed for 10000 different randomly selected perception schedules aiming to approximate the otherwise intractable exhaustive search for the true optimal cost.

Another interesting experiment is obtained by increasing $\lambda_\alpha = 15$ such that the cost of the covariance is negligible to the one of the penalties r^{p_k} in (3.2). In this case, $J_{\min}$ is directly the static schedule $p = \{1, 1, \dots\}$ which is the schedule with less CPU load. In addition, since the particular values of covariance $\hat{\mathbf{P}}[k]$ have little impact on the final cost, it only suffices to use a single point $Q(\delta) = 1$ in the discretization of the covariance space. Hence, the result of qDP is the static schedule $p = \{1, 1, \dots\}$ regardless of the initial condition $\hat{\mathbf{P}}[0]$ as shown in the third row of Figure 3.5.

3.6.3 Moving-horizon PLATE scheduling

We simulate (3.1) for $\mathbf{x}_0 = 0, P[0] = 4I$ using the Euler-Maruyama discretization with time step $\Delta t = 10^{-3}$ and run the loop in Algorithm 3.1 under the moving-horizon scheduling of Algorithm 3.6 where qDP is configured similarly as in the previous examples, with $T_f = 10, \lambda_\alpha = 5, Q(\delta) = 5000, B_0 = 5$ (Algorithm 3.3). Figure 3.6 depicts the time evolution of $\mathbf{x}(t)$ as well as the PLATE estimate $\hat{\mathbf{x}}(t)$, where we show only x and v_x for convenience, since y and v_y behave similarly. In addition, we show confidence intervals of 3 times the standard deviation for each coordinate obtained from $\hat{\mathbf{P}}(t)$. In addition, we show the scheduling decision p_k at each time. In order to compare the cost in this experiment we show $\mathsf{tr}(\hat{\mathbf{P}}(t))$ for the moving-horizon scheduling, as well as for the static schedules. It is worth noting that the best quality measured with $\mathsf{tr}(\hat{\mathbf{P}}(t))$ is obtained by the static schedule with $p_k = 2$. However, it has the highest CPU load of 0.8. The moving-horizon PLATE strategy manages to have an intermediate quality between the two static schedules, with a CPU load of 0.65 for this experiment. Thus, PLATE manages to obtain a better trade-off between quality and resource usage. In addition, an occlusion is simulated in the interval $t \in [4, 6]$ which shows how uncertainty increases during this period, but recovers once new measurements arrive.

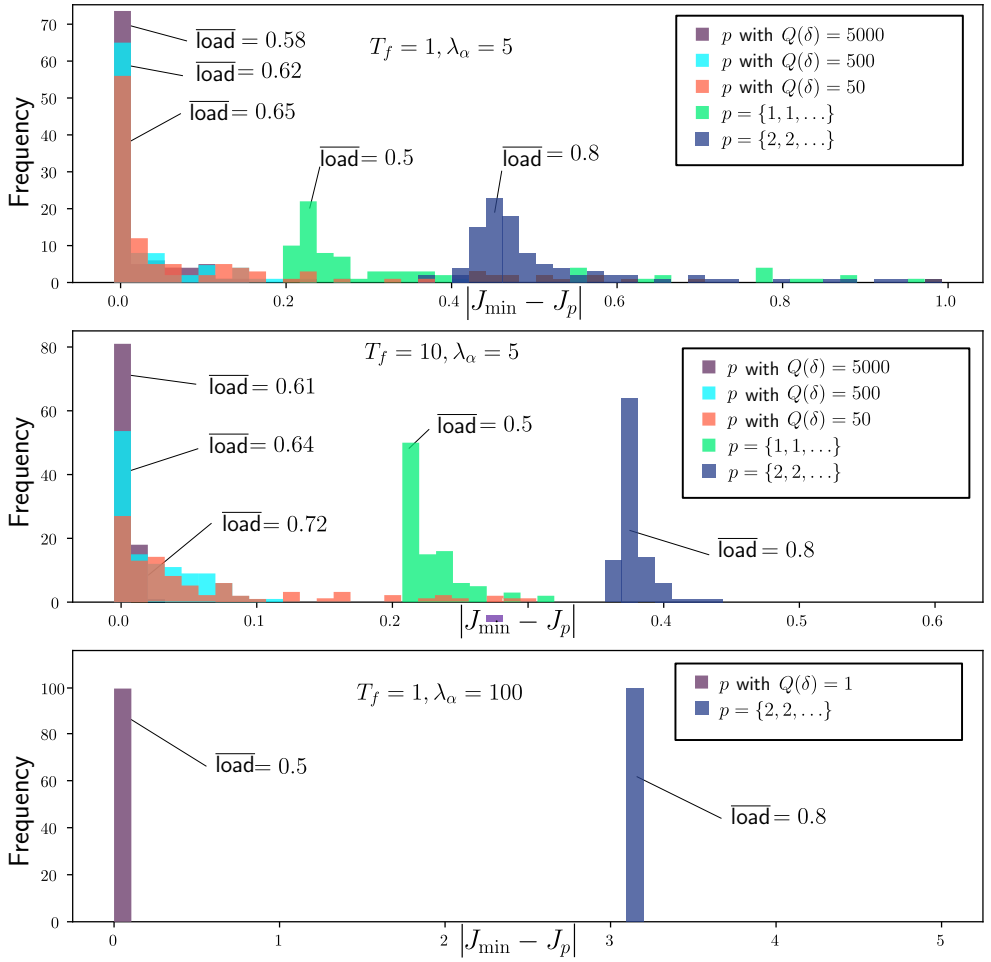

Figure 3.5: Resulting histograms for the cost difference $|J_{\min} - J_p|$ with the schedule obtained from the approximate dynamic programming approach for 100 random initial conditions for $P[0]$ as described in Section 3.6.2. The parameters T, λ_α of the cost function in (3.2) were changed for three different scenarios as well as the number of quantization points $Q(\delta)$. In addition, only the average CPU load for all experiments is shown for convenience.

3.7 EVALUATION ON REAL DATA

In this section, we evaluate PLATE on a real target tracking task. We use the MOT16 benchmark from [136] since it is part of a public standard data-set, widely used to evaluate target tracking algorithms. In the simplest setting, the task is to take images from a monocular camera and track a target of interest across multiple frames. As usual, and for the sake of simplicity, we are interested in tracking the pixel position of the geometric center of a bounding box surrounding the target. For this example, we model the pixel position with a two-dimensional single integrator with fixed $n_{\mathbf{x}} = n_{\mathbf{z}} = n_{\mathbf{w}} = 2$,

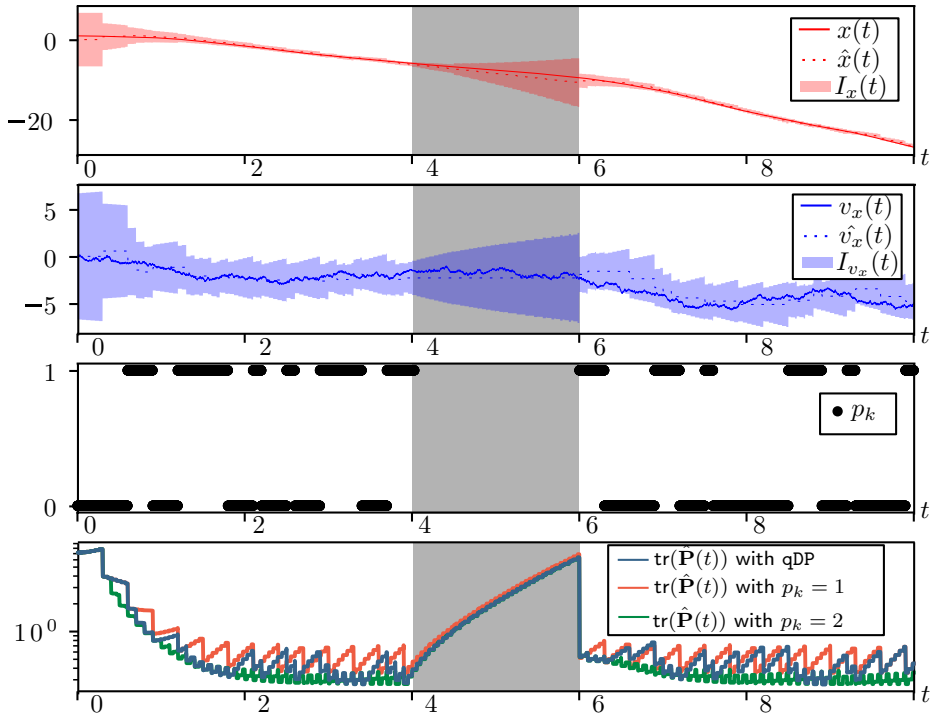

Figure 3.6: Behaviour of the target model (3.1) and the output of PLATE using the moving-horizon scheduling with $T_f = 10$ and $Q(\delta) = 5000$ when an occlusion occurs for $t \in [4, 6]$. Moreover, $I_x(t)$ and $I_{v_x}(t)$ represent confidence intervals around $\hat{x}(t)$ and $\hat{v}_x(t)$ of 3 times their standard deviation.

$\mathbf{A} = \mathbf{0} \in \mathbb{R}^{2\times 2}, \mathbf{B} = \mathbf{C} = \mathbf{I} \in \mathbb{R}^{2\times 2}$.

In the following, we describe a target tracking pipeline that consists of a perception stage composed of three different target detection methods based on neural networks, each exhibiting varying latency and accuracy performance, and a scheduling stage responsible for deciding which of these methods to use at each time. The objective of this section is to demonstrate that PLATE is the best choice for the scheduling stage. For this purpose, we compare the tracking performance of PLATE with two other approaches: 1) a standard neural network-based method, which involves using the perception methods separately with no scheduling, 2) event-triggered approaches for frame-skipping.

Name	Target IDs	fps	Usage
MOT16-05	2,37,41,124,128	14	T
MOT16-13	13,14,29,30,76,77,79,81,95,124	25	T
MOT16-02	3,19,20,32,33,34,35	30	E
MOT16-04	2,67,70,80,83,98,99,100	30	E
MOT16-09	12,13,19,20,23	30	E
MOT16-10	1,2,4,7,18,24	30	E
MOT16-11	1,3,4,5,6,11,25,27,29	30	E

Table 3.1: Ground-truth target IDs, as provided by the MOT16 data-set, picked for each video sequence in our evaluation. Frames per second is abbreviated as fps. The last column indicates T or E if the target tracks in the corresponding video sequence are used for training or for evaluation respectively.

3.7.1 Evaluation framework

The MOT16 data-set includes 7 video sequences, each with labeled bounding-boxes for different moving targets. We manually selected several targets for each sequence. We recorded their ground-truth positions and the corresponding portion of the sequence, which we refer to as target tracks from now on. We chose targets that provided longer experiments, in order to obtain more comprehensive results. Our evaluation data consists of 50 target tracks across all 7 video sequences, whose ground-truth IDs are shown in Table 3.1 as a reference. Note that for all video sequences the frame rate is set to 30 frames per second except for MOT16-05 and MOT16-13 with frame rates of 14 and 25 frames per second respectively. As a result, target tracks for MOT16-13 and MOT16-05 (15 tracks in total) were used exclusively to estimate the nominal detection covariance for each perception method as described later, and the covariance $\mathbf{W}$ in (3.1). This process is called training in Table 3.1. The rest of the 35 target tracks contained in the remaining 5 video sequences were used to evaluate the online performance of PLATE. From this point onward, all experiments (both for our proposal and other proposals used for comparison) were conducted on a PC equipped with an Intel Core i7-8700, along with the aid of an NVIDIA GeForce GTX 1080 Ti graphics card.

To evaluate the performance of PLATE, it is necessary to use a setup that includes a bank of perception methods. However, the purpose of this chapter is not to evaluate the individual performance of these methods, but rather to measure the advantages of PLATE in terms of target state estimation quality and resource usage, such as CPU load. Therefore, we selected well-established perception methods from the public model zoo of Detectron2 [137]. This repository provides many neural network-based target detection algorithms with varying levels of latency and precision, making it suitable for our work. Specifically, we chose the `faster_rcnn_R_50_FPN_3x` and `faster_rcnn_R_101_C4_3x` models (hereinafter referred as the fast and slow networks respectively), which represent the clearest examples in the zoo of low quality detection with short latency and good

quality detection with long latency, respectively. These models have a reported latency of 0.038s and 0.104s on a standard computing platform, which were verified in our own setup. To further simplify the experiments, for the fast network we sub-sample the input image in a factor of two, reducing the latency to roughly $1/(30\text{fps}) \approx 0.033s$ which is the inverse of the frame rate. With these two networks we propose 3 perception methods as follows:

- $p_k = 1$: This method uses the fast network with a latency of $1/(30\text{fps})$. This means that this method does not skip any frame and has a CPU load of 100%.
- $p_k = 2$: It uses the slow network with a processing latency of $0.104s$. This means that the network is computed over four frames, causing it to skip three frames, processing only the first one. Consequently, the CPU load over these four frames is $\frac{0.104s}{4\text{frames}/(30\text{fps})} = 78\%$.
- $p_k = 3$: In order to have an option which favours a low CPU load, we use the fast network, but deliberately skip the next 4 frames. This means that the perception quality of this method is the same as $p_k = 1$ but with a CPU load of $\frac{1/30\ s}{5\text{frames}/(30\text{fps})} = 20\%$ and longer overall latency.

These perception method options are depicted graphically in Figure 3.7. The previous framework has a disadvantage in terms of memory consumption, as the adaptive strategy requires storing both neural networks, leading to higher memory usage compared to using the perception methods individually. However, it is worth noting that depending on the application, it may be possible to use a single ANN, as demonstrated in [57], which can offer different latency and quality levels with a fixed amount of memory space. As

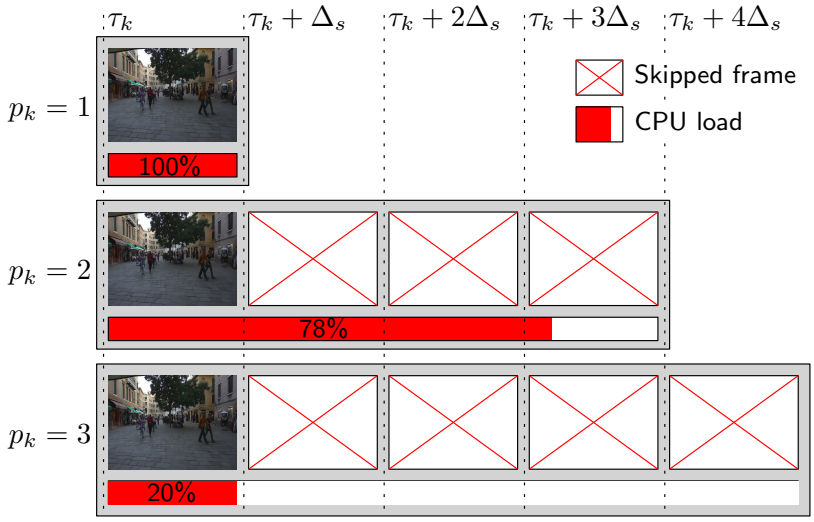

Figure 3.7: Different features of the three perception methods described in Section 3.7.1. The number of skipped frames as well as the CPU load in each method is depicted.

depicted in Figure 3.8, the output of each of the neural networks is a list of bounding

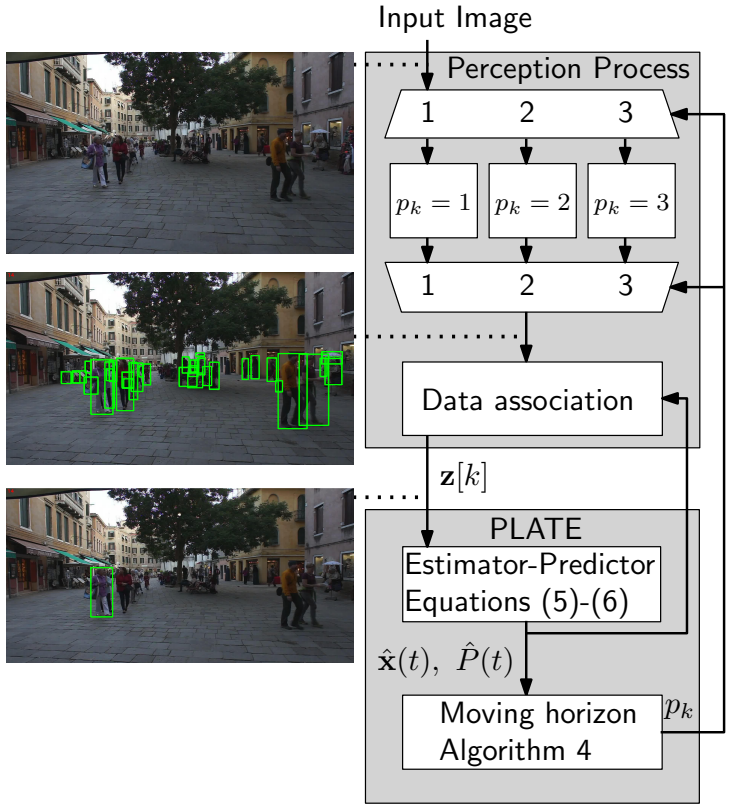

Figure 3.8: Implemented Pipeline for the PLATE evaluation framework described in Section 3.7.1

boxes for all the detected objects in the image. Therefore, a data association stage is necessary to select the appropriate bounding box for the target object in each experiment. There exist many data association techniques in the literature, such as those based on similarity measures using neural networks [138] or end-to-end detection and association siamese networks [139]. These methods are highly dependent on the training procedure, which might have an undesired effect when evaluating the scheduling feature of PLATE by itself. Hence, since the goal of this chapter is to evaluate PLATE for a fixed perception framework, we use a more standard approach. We compare the predicted position of the target, $\lim_{t\to\tau_k^-} \hat{\mathbf{x}}(t)$, from (3.4) at the time τ_k when the image was captured, and use the Hungarian algorithm to select the most suitable bounding box candidate [135]. The computational latency of this procedure is negligible and can be ignored for simplicity in subsequent discussions. Figure 3.8 shows that the output of the data association stage is the best bounding box candidate for the target of interest from which a processed position measurement $\mathbf{z}[k]$ is obtained as its geometric center.

In this setting, $\Delta_s = 1/30, \Delta^1 = \Delta_s, \Delta^2 = 3\Delta_s, \Delta^3 = 5\Delta_s$. In addition, the CPU loads are $f^1 = 1, f^2 = 0.78, f^3 = 0.2$. We estimate the nominal error covariance for each perception method $\mathbf{R}^1, \mathbf{R}^2, \mathbf{R}^3$ by comparing the outputs of each perception method with

the ground-truth data across all images in the 15 training target tracks. The resulting covariance matrices are

$$\mathbf{R}^1 = \mathsf{diag}(13.12^2, 25.87^2), \mathbf{R}^2 = \mathsf{diag}(9.94^2, 17.06^2), \mathbf{R}^3 = \mathbf{R}^1$$

The matrix $\mathbf{W}$ was estimated using the training target tracks as well, following a standard parameter estimation procedure described in [134]. As described in Section 3.5, we also aim to evaluate PLATE when an online estimation of the current covariance $\mathbf{R}[k]$ is available besides the nominal covariances. As a result, we adapt the standard parameter estimation in [134] for this setting, using the resulting performance of previous detections compared to the estimation $\hat{\mathbf{x}}(t)$. Let $\mathcal{I}^{p_k}[k] = \{\ell \leq k : p_\ell = p_k, k - \ell \leq N_w\}$ be the set of all discrete instants ℓ corresponding to the moments τ_ℓ in which the perception method p_k was used prior to the current τ_k, in a moving window picked here of $N_w = 10$ samples. Thus, if p_k is to be picked at $t = \tau_k$ one can estimate the current covariance for such method as:

$$\mathbf{R}[k] = \frac{1}{N_w} \sum_{\ell \in \mathcal{I}^{p_k}[k]} \mathbf{e}[\ell]\mathbf{e}[\ell]^\top - \mathbf{C}\hat{\mathbf{P}}^-(\tau_k)\mathbf{C}^\top \tag{3.8}$$

where $\mathbf{e}[\ell] = \mathbf{C}\hat{\mathbf{x}}[\ell] - \mathbf{z}[\ell]$ and $\hat{\mathbf{P}}^-(\tau_k) = \lim_{t \to \tau^-} \hat{\mathbf{P}}(t)$ from (3.4).

The PLATE module, depicted in Figure 3.8, executes the estimator-predictor equations specified in (3.4) and (3.5), and implements the scheduler algorithm presented in Algorithm 3.6. This module employs a moving-horizon PLATE with $T_f = 10s$ and utilizes pre-computed schedules for each quantized covariance value $\hat{\mathbf{P}}_q$, which are based on nominal $\mathbf{R}^1, \mathbf{R}^2$, and $\mathbf{R}^3$. When a new processed measurement $\mathbf{z}[k]$ is obtained, the pre-computed values for line 7 in Algorithm 3.6 for all possible $\hat{\mathbf{P}}[k]$ in a compact set enable the computation of the new scheduling decision p_k with negligible computing latency. Additionally, the computation time required for (3.4) and (3.5) in this example's state space dimensions is negligible compared to the latencies Δ^1, Δ^2, and Δ^3. The values of $Q(\delta)$ will be varied between $50, 500, 5000$ as in previous examples with $\mathcal{B}_0$ as the set of all positive definite matrices $\mathbf{P}$ with $\|\mathbf{P}\|_F \leq \mathbf{B}_0 = 20$.

Remark 3.13. *The cost function* (3.2) *was set using* $\lambda_\alpha = 1$ *and*

$$r^{p_k} = \lambda_{\mathsf{load}} f^{p_k} \Delta^{p_k} + \lambda_{\mathsf{att}}. \tag{3.9}$$

The term $\lambda_{\mathsf{load}} f^{p_k} \Delta^{p_k}$ *penalizes the CPU load, while* λ_{att} *penalizes the attention, weighted by* $\lambda_{\mathsf{load}}, \lambda_{\mathsf{att}}$. *Minimizing these objectives while providing the best possible accuracy are conflicting goals, but can be balanced by adjusting* $\lambda_{\mathsf{load}}, \lambda_{\mathsf{att}}$. *If* $\lambda_{\mathsf{load}} = \lambda_{\mathsf{att}} = 0$, *the offline construction of* p_k *results in* $p_k = 1$ *for every* $\hat{\mathbf{P}}_q$, *maximizing the attention regardless of the quantization step. Conversely, we verified numerically that for* $\lambda_{\mathsf{load}} = 1$ *and* $\lambda_{\mathsf{att}} > 15$, $p_k = 3$ *for all* $\hat{\mathbf{P}}_q$, *minimizing the attention, regardless of the quantization step as well. For other values of* $\lambda_{\mathsf{load}}, \lambda_{\mathsf{att}}$, *the schedule* p_k *changes according to the current* $\hat{\mathbf{P}}_q$, *achieving a balance between these objectives.*

3.7.2 Performance of the implemented pipeline

In this section, we evaluate the proposed pipeline on the 35 evaluation target tracks. The performance evaluation metric used is the Mean Squared Error (MSE) of $\hat{\mathbf{x}}(k\Delta_s), k = 0, 1, \ldots$ compared to the ground-truth data available in the MOT16 data-set. Additionally, the CPU load for all experiments is recorded, as well as the attention measured as the percentage of non-skipped frames compared to the total number of frames. To illustrate the trade-off between the previously described metrics, a combined cost is computed as $(\text{MSE}) + \lambda_{\mathsf{load}}(\text{CPU load}) + \lambda_{\mathsf{att}}(\text{Attention})$ as an sampled version of the cost (3.2) under penalties (3.9).

In the following, we used $\lambda_{\mathsf{load}} = \lambda_{\mathsf{att}} = 1/2$ in (3.9) which results in a trade-off between accuracy, CPU load and attention, producing different p_k through time for PLATE. The results are shown in Table 3.2 where we compare the performance of the proposal under different values of $Q(\delta)$ as well as using nominal $\mathbf{R}[k] \in \{\mathbf{R}^1, \mathbf{R}^2, \mathbf{R}^3\}$ or adaptive $\mathbf{R}[k]$ in (3.8).

There are three main conclusions that can be obtained from these results. First, note that using adaptive $\mathbf{R}[k]$ produces an improvement with respect to the nominal case, particularly with low $Q(\delta)$ where an improvement of $(35.76 - 32.78)/35.76 \approx 8\%$ in MSE is obtained when $Q(\delta) = 50$. Second, consistent with Theorem 3.8, increasing $Q(\delta)$ improves the MSE as well, with diminishing returns as $Q(\delta)$ increases. Third, the CPU load and attention remain roughly the same meaning that high values of $Q(\delta)$ might not be needed in practice to maintain reasonable values for these objectives.

Now, we compare PLATE with other ideas in the literature. As a baseline for the comparison we used a framework which does not use scheduling. This is, the neural network method to be used for perception is chosen at the beginning, and remains fixed at all times. The results are shown in the first three rows of Table 3.3. Note that the configuration in row 1) of Table 3.3 (fixed $p_k = 1$) obtains the best MSE among all experiments at the expense of high CPU load and attention. In addition, row 3) of Table 3.3 (fixed $p_k = 3$) produces the best results in terms of CPU load and attention at the expense of high MSE. However, recall from Remark 3.13 that with appropriate configuration of the penalties (3.9), PLATE can produce either fixed $p_k = 1$ or $p_k = 3$ and obtain the best performance for MSE, CPU load or attention by separate.

Moreover, we evaluated frame-skipping techniques [60, 73, 135] which rely on an event-triggered condition to determine whether a new frame needs processing. To evaluate a similar approach, we decided at each new frame whether to use $p_k = 1$ or skip the frame entirely. Send-on-delta strategies are commonly employed, where a frame is processed if $\mathsf{tr}(\hat{\mathbf{P}}(k\Delta_s)) \geq \delta_{\mathsf{ET}}$ with a threshold $\delta_{\mathsf{ET}} > 0$. The results for different δ_{ET} are shown in rows 4) and 5) of Table 3.3.

Finally, row 6) of Table 3.3 shows the best configuration for PLATE as obtained in Table 3.2. For comparison, note that while fixed $p_k = 2$ yields a small MSE, CPU load and attention values remain high. In contrast, PLATE in row 6) of Table 3.3 also has a small MSE with the advantage of reducing the CPU load from 78% to 43.5% and attention from 75% to 43.4%, when compared to fixing $p_k = 2$.

The event-triggered approach in row 4) of Table 3.3 shows a small improvement in the MSE can be obtained when compared to PLATE in row 6) of Table 3.3. Despite this, the CPU load and attention values are considerably bigger for the event triggered alternative. More concretely, PLATE in row 6) of Table 3.3 improves the CPU load from

71.2% to 43.5% (a relative improvement of 38.9%) when compared to the event triggered approach in row 4) of Table 3.3, even with a similar MSE performance. This trade-off is clear since PLATE obtains the best performance among all options as ilustrated with the combined cost in the last column of Table 3.3.

Hence, using PLATE with appropriate configuration in (3.9) can obtain a good trade-off between accuracy, CPU load and attention, with improved performance with respect to static perception configurations. In addition, PLATE provides a clear connection via the cost function (3.2) and the penalty (3.9) to achieve in each situation different behaviors, prioritizing CPU load and attention or MSE.

Remark 3.14. *The performance of the proposal is highly dependent on the parameters $Q(\delta)$, λ_{load}, and λ_{att}, as evident from the results presented in Tables 3.2 and 3.3. It is important to note that the optimal parameter selection generally varies based on the specific application and target dynamics under consideration. However, as discussed in this section, conducting several experimental tests can help determine the most suitable parameters for the user. Our experiments demonstrate a range of potential performance values obtained by varying the parameter $Q(\delta)$, allowing for selection based on specific application requirements and resource availability. In addition, regarding the penalties λ_{load}, and λ_{att}, these values should reflect the application specifications. Still, it is beneficial to identify parameter sets for these penalties that exhibit similar behavior to the ones described in Remark 3.13. This knowledge becomes particularly valuable when the user needs to make a trade-off between accuracy and resource usage. More principled design rules are not trivial to obtain and will be considered in our future work.*

3.8 DISCUSSION

This chapter focused on a specific aspect of the architecture presented in Figure 1.4 from Chapter 1. The primary objective was to address the complementary problem of observing an unknown input system, specifically for target tracking applications. This contrasts the previous Chapter 2, where we tackled a control problem. For this purpose, we introduced PLATE as a perception-latency aware estimator. PLATE leverages a bank of perception methods with different latency-precision trade-offs to adaptively select the best method for the current estimation task. The proposed algorithm allows for skipping input frames, reducing CPU load and resource usage while maintaining high tracking accuracy. Unlike other frame-skipping techniques, PLATE's algorithm is based on a formal dynamic programming argument rather than heuristics. We found that while the exact solution of the problem is subject to a combinatorial explosion, an approximate solution can be obtained with efficient computational complexity. We evaluated PLATE using both simulations and real-world data sets. We found that it outperforms other state-of-the-art approaches in terms of both tracking accuracy and computational efficiency.

The ideas presented here for perception-based target estimation can be used to generate references for control tasks in a multi-agent context, which will be explored in Chapter 7. So far, we haven't considered cooperation between robots, but upcoming chapters will present various cooperation strategies, including information fusion and averaging using consensus algorithms.

	$\mathbf{R}[k]$	$Q(\delta)$	**MSE** [px]	**CPU load** [%]	**Attention** [%]	**Combined cost**
1)	Nominal	50	35.76	44.5	44.1	80.06
2)	Adaptive	50	32.78	44.0	43.9	76.73
3)	Nominal	500	32.14	44.1	43.9	76.14
4)	Adaptive	500	32.02	43.8	43.6	75.72
5)	Nominal	5000	31.54	43.9	43.6	75.29
6)	Adaptive	5000	**30.04**	**43.5**	**43.4**	**73.49**

Table 3.2: Evaluation of the implemented pipeline for PLATE on MOT16 data as described in Section 3.7. MSE stands for Mean-Squared-Error with respect to groundthruth data. The attention column corresponds to the percentage of non-skipped frames with respect to the total number of frames. The penalty in (3.9) is configured with $\lambda_{\mathsf{load}} = \lambda_{\mathsf{att}} = 1/2$. Moreover, to illustrate the trade-off between the previously described metrics, a combined cost is computed as (MSE) + λ_{load}(CPU load) + λ_{att}(Attention) as an sampled version of the cost (3.2) under penalties (3.9).

	Method	**Configuration**	**MSE** [px]	**CPU load** [%]	**Attention** [%]	**Combined cost**
1)	No scheduling	fixed $p_k = 1$	**19.61**	100	100	119.61
2)	No scheduling	fixed $p_k = 2$	29.05	78	75	105.55
3)	No scheduling	fixed $p_k = 3$	73.73	**20**	**20**	93.73
4)	Event-triggered	$\delta_{\mathsf{ET}} = 5$	25.04	71.2	69.2	95.24
5)	Event-triggered	$\delta_{\mathsf{ET}} = 50$	64.74	35.9	35.2	100.29
6)	**PLATE**	$Q(\delta) = 5000$	**30.04**	**43.5**	**43.4**	**73.49**

Table 3.3: Evaluation of the implemented pipeline. The evaluation metrics are the MSE, CPU load and attention as well as the combined cost taking into account all previous metrics weighted by $\lambda_{\mathsf{load}} = \lambda_{\mathsf{att}} = 1/2$. The methods under evaluation are: 1) No scheduling with fixed $p_k = 1$, corresponding to using only the fast neural network `faster_rcnn_R_50_FPN_3` at all times without additional frame skipping. 2) No scheduling with fixed $p_k = 2$ corresponding to using only the slow neural network `faster_rcnn_R_101_C4_3x` at all times without additional frame skipping. 3) No scheduling with fixed $p_k = 3$ corresponding to using only the fast neural network `faster_rcnn_R_50_FPN_3` followed by 4 skipped frames repeatedly. 4) and 5) Event triggered approach with different δ_{ET}. 6) The best configuration for PLATE as obtained from Table 3.2.

3.9 PROOFS

3.9.1 Proof of Theorem 3.4

First, we show an following auxiliary result.

Lemma 3.15. *Let the state equation* (3.3) *and the measurement model* $\mathbf{z}[k] = \mathbf{C}\mathbf{x}[k] + \mathbf{v}[k]$ *available at* $t = \tau_k + \Delta^{p_k}$ *with* $\mathsf{cov}\{\mathbf{v}[k]\} = \mathbf{R}^{p_k}$ *and a fixed latency schedule* p*. Thus, the estimate at time* $t \in [\tau_k, \tau_{k+1})$ *of* $\mathbf{x}(t)$ *based on measurements* $\{\mathbf{z}[0], \ldots, \mathbf{z}[k]\}$ *which minimize* $\mathsf{tr}(\hat{\mathbf{P}}(t))$*, is given by* $\mathbb{E}\{\mathbf{x}(t)|\mathbf{z}[0], \ldots, \mathbf{z}[k-1]\}$ *and satisfies* (3.4)*.*

PROOF: First, consider estimates for $\mathbf{x}[k]$, given that $\mathbf{x}[k]$ evolves according to the discrete-time system (3.3b). Moreover, note that the measurement $\mathbf{z}[k]$ is available at $t = \tau_{k+1}$. Then, [119, Page 228 - Theorem 4.1] implies that the structure in (3.5), inherited from a Kalman filter with predictor, satisfies $\hat{\mathbf{x}}[k+1] \equiv \mathbb{E}\{\mathbf{x}[k+1]|\mathbf{z}[0], \ldots, \mathbf{z}[k]\}$. For any other $t \in (\tau_k, \tau_{k+1})$ the measurement $\mathbf{z}[k]$ is not available. Thus, $\mathbb{E}\{\mathbf{x}(t)|\mathbf{z}[0], \ldots, \mathbf{z}[k-1]\}$ can be computed using $\mathbb{E}\{\mathbf{x}[k]|\mathbf{z}[0], \ldots, \mathbf{z}[k-1]\}$ through the same structure in (3.5) but applied to (3.3a) with $\mathbf{C} = 0$ resulting in (3.4). Now that $\hat{\mathbf{x}}(t) = \mathbb{E}\{\mathbf{x}(t)|\mathbf{z}[0], \ldots, \mathbf{z}[k-1]\}$ has been established for any $t \in [\tau_k, \tau_{k+1})$, [119, Page 228 - Theorem 4.1] implies that $\mathbf{a}^\top \hat{\mathbf{P}}(t)\mathbf{a}$ is minimized for this estimate with arbitrary vector $\mathbf{a} \in \mathbb{R}^{n\times}$. This means that for any other estimation $\hat{\mathbf{x}}'(t)$ of $\mathbf{x}(t)$ with covariance $\hat{\mathbf{P}}'(t) := \mathsf{cov}\{\mathbf{x}(t) - \hat{\mathbf{x}}'(t)\}$ we have

$$\mathbf{a}^\top \hat{\mathbf{P}}(t)\mathbf{a} \leq \mathbf{a}^\top \hat{\mathbf{P}}'(t)\mathbf{a}, \forall \mathbf{a} \in \mathbb{R}^n.$$

Hence, $\hat{\mathbf{P}}(t) \preceq \hat{\mathbf{P}}'(t)$ from which $\mathsf{tr}(\hat{\mathbf{P}}(t)) \leq \mathsf{tr}(\hat{\mathbf{P}}'(t))$ follows using Lemma C.2 from C for any other estimation $\mathbf{x}'(t)$. □

We are now ready to show Theorem 3.4. First, note that for fixed perception schedule p, the term $\sum_{k=0}^{\alpha} r^{p_k}$ in (3.2) is constant. Moreover, Lemma 3.15 implies that $\hat{\mathbf{x}}(t) = \mathbb{E}\{\mathbf{x}(t)|\mathbf{z}[0], \ldots, \mathbf{z}[k-1]\}$ computed recursively using (3.4) and (3.5) leads to an optimal trajectory of the signal $\mathsf{tr}(\hat{\mathbf{P}}(t))$. This means that for any other estimation $\hat{\mathbf{x}}'(t)$ of $\mathbf{x}(t)$ with $\hat{\mathbf{P}}'(t) := \mathsf{cov}\{\mathbf{x}(t) - \hat{\mathbf{x}}'(t)\}$ one has $\mathsf{tr}(\hat{\mathbf{P}}(t)) \leq \mathsf{tr}(\hat{\mathbf{P}}'(t))$. Integrating both sides of the previous inequality leads to conclude that

$$\int_0^{T_f} \mathsf{tr}(\hat{\mathbf{P}}(t))\mathrm{d}t \leq \int_0^{T_f} \mathsf{tr}(\hat{\mathbf{P}}'(t))\mathrm{d}t$$

which implies $\mathcal{J}(\hat{\mathbf{x}}, p) \leq \mathcal{J}(\hat{\mathbf{x}}', p)$. Hence, the estimation $\hat{\mathbf{x}}(t), \forall t \in [0, T_f]$ from (3.4) solves Problem 3.1 provided that p is the optimal schedule as well.

3.9.2 Proof of Proposition 3.5

To show correctness, first note that a direct application of the dynamic programming principle [124, Page 23] leads to conclude that p as obtained from $\texttt{dynProg}(0, P[0], T_f)$ is optimal for the cost (3.2) whenever the estimations $\{\hat{\mathbf{x}}(t) : t \in [0, T_f]\}$ are of the form (3.4) as required in lines 4 and 6 of `dynProg`. In addition, Theorem 3.4 implies that the optimal estimations are computed from (3.4), from which optimality of both $\hat{\mathbf{x}}(t)$ and p using `dynProg` follows, solving Problem 3.1. For the complexity, note that the largest amount of recursive calls in Algorithm 3.2 is obtained when $p_k = i$ at line 3 of `dynProg` with $i = \text{argmin}\{\Delta^j\}_{j=1}^D$. In this case, $\lfloor T_f / \min\{\Delta^1, \ldots, \Delta^D\} \rfloor$ recursive calls

to `dynprog` are required to cover the whole window $[0, T_f]$, i.e. for $\tau^+ \geq T_f$ in line 7 of `dynProg`. Hence, due to line 3 in `dynProg`, the total number of recursive calls is at most $D^{\lfloor T_f / \min\{\Delta^1,\ldots,\Delta^D\}\rfloor}$.

3.9.3 Proof of Proposition 3.6

First we show that Algorithm 3.3 finishes. Note that for any initial condition in a compact set $\mathcal{B}_0$, the result in [140, Lemma 6.1] ensures that the covariance $\hat{\mathbf{P}}[k]$ a the filter of the form (3.5) is uniformly bounded from above. This means that there exists a compact set $\mathcal{B} \subset \mathbb{R}^{n\times n}$ such that $\hat{\mathbf{P}}[k] \in \mathcal{B}, \forall k \geq 0$ and $\hat{\mathbf{P}}_0 \in \mathcal{B}_0$. Existence of such compact set in the quantized setting is ensured as well. The reason is that quantization only adds a disturbance term $\|\mathbf{\Pi}_{\mathcal{Q}}[k]\| \leq \delta$ to (3.5) which can be absorbed into the covariance $W(\Delta^{p_k})$. Now, note that since quantization patches obtained in Algorithm 3.3 are non overlapping and comply

$$\sup_{\hat{\mathbf{P}}[k],\hat{\mathbf{P}}[k]'\in\mathcal{B}_{Q(\delta)}} \|\hat{\mathbf{P}}[k] - \hat{\mathbf{P}}[k]'\| = \delta$$

thus, there exists a maximum finite number $\bar{Q}(\delta)$ of regions of this kind covering $\mathcal{B}$ due to compactness. Hence, Algorithm 3.3 finishes in at most $\bar{Q}(\delta)$ steps.

For the rest of the proposition, the proof outline is described in the following. Note that obtaining an explicit bound for $\hat{\mathbf{P}}[k]$ in (3.5) is not trivial due to the nonlinearity of the update equation for the covariance. In addition, the bound in [140, Lemma 6.1] is very overestimated. Hence, we follow a similar idea as in [140] which is to study an artifact filter whose bound can be obtained explicitly and use it as a proof tool, appealing to the optimality of PLATE to conclude that a covariance bound of the artifact filter is a bound for PLATE as well. In this case, instead of the structure in (3.5) where the gain $\mathbf{L}[k]$ depends on $\hat{\mathbf{P}}[k]$, we use a gain $\mathbf{L}^{p_k} \in \{\mathbf{L}^1, \ldots, \mathbf{L}^D\}$ so that the artifact filter takes the form:

$$\hat{\mathbf{x}}_s[k+1] = \mathbf{A}_{\mathsf{d}}(\Delta^{p_k})\hat{\mathbf{x}}_s[k] + \mathbf{L}^{p_k}(\mathbf{z}[k] - \mathbf{C}\hat{\mathbf{x}}_s[k]) \tag{3.10}$$

with $\hat{\mathbf{x}}_s[0] = \mathbf{x}_0$. Hence, we require to design the gains $\mathbf{L}^{p_k}$ to make the filter asymptotically stable, making it feasible for the bound to exist. This cannot be done by designing each $\mathbf{L}^{p_k}$ by separate. The reason is that, the error $\tilde{\mathbf{x}}_s[k] = \mathbf{x}[k] - \hat{\mathbf{x}}_s[k]$ for (3.10) satisfies

$$\tilde{\mathbf{x}}_s[k+1] = \mathbf{\Lambda}^{p_k}\tilde{\mathbf{x}}_s[k] + \mathbf{L}^{p_k}\mathbf{v}[k] + \mathbf{w}_d[k] \tag{3.11}$$

with $\mathbf{\Lambda}^{p_k} = \mathbf{A}_{\mathsf{d}}(\Delta^{p_k}) + \mathbf{L}^{p_k}\mathbf{C}$. Hence, (3.11) is a switched system which switches between system matrices $\mathbf{\Lambda}^{p_k}$ with the schedule p_k as switching signal. In the following auxiliary technical lemmas, we aim to ensure asymptotic stability of the filter, by designing $\mathbf{L}^{p_k}$ through the method of the common Lyapunov function [108, Page 22].

Lemma 3.16. *Consider the filter* (3.10) *for some perception schedule* p*. Then,* $\boldsymbol{\varrho}_s[k] := \mathsf{vec}(\hat{\mathbf{P}}_s[k])$ *with* $\hat{\mathbf{P}}_s[k] := \mathit{cov}\{\mathbf{x}[k] - \mathbf{x}_s[k]\}$ *satisfies:*

$$\boldsymbol{\varrho}_s[k+1] = (\mathbf{\Lambda}^{p_k} \otimes \mathbf{\Lambda}^{p_k})\boldsymbol{\varrho}_s[k] + \boldsymbol{\omega}[k], \boldsymbol{\varrho}_s[0] \quad = \mathsf{vec}(\mathbf{P}_0) \tag{3.12}$$

where

$$\boldsymbol{\omega}[k] = \mathsf{vec}((\mathbf{L}^{p_k})\mathbf{R}^{p_k}(\mathbf{L}^{p_k})^\top + \mathbf{W}_d(\Delta^{p_k}))$$

PROOF: First, compute $\hat{\mathbf{P}}_s[k+1]$ by applying $\mathsf{cov}(\bullet)$ to both sides of (3.11) as:

$$\hat{\mathbf{P}}_s[k+1] = (\mathbf{\Lambda}^{p_k})\hat{\mathbf{P}}_s[k](\mathbf{\Lambda}^{p_k})^\top + (\mathbf{L}^{p_k})\mathbf{R}^{p_k}(\mathbf{L}^{p_k})^\top + \mathbf{W}_d(\Delta^{p_k})$$

Then, apply $\mathsf{vec}(\bullet)$ to both sides of the previous equation as well as the identity

$$\mathsf{vec}((\mathbf{\Lambda}^{p_k})\hat{\mathbf{P}}_s[k](\mathbf{\Lambda}^{p_k})^\top) \equiv (\mathbf{\Lambda}^{p_k} \otimes \mathbf{\Lambda}^{p_k})\mathsf{vec}(\hat{\mathbf{P}}_s[k])$$

to obtain (3.12). Finally, $\hat{\mathbf{P}}[0] = \mathsf{cov}\{\mathbf{x}[0] - \mathbf{x}_0\} = \mathbf{P}_0$. □

Lemma 3.17. *Consider $\mathbf{\Omega}, \gamma, \mathbf{Y}^i, i \in \{1, \ldots, D\}$ satisfy* (3.7)*. Henceforth, the following nonlinear matrix inequality is satisfied:*

$$\left((\mathbf{\Lambda}^{p_k})^\top \mathbf{\Omega}(\mathbf{\Lambda}^{p_k}) \otimes (\mathbf{\Lambda}^{p_k})^\top \mathbf{\Omega}(\mathbf{\Lambda}^{p_k})\right) \preceq \gamma^2(\mathbf{\Omega} \otimes \mathbf{\Omega}) \tag{3.13}$$

with $\mathbf{\Lambda}^i = \mathbf{A}_\mathsf{d}(\Delta^i) - \mathbf{L}^i\mathbf{C}$ and $\mathbf{L}^i = \mathbf{\Omega}^{-1}\mathbf{Y}^i, i \in \{1, \ldots, D\}$.

PROOF: Equivalence between (3.7) and $(\mathbf{\Lambda}^{p_k})^\top \mathbf{\Omega}(\mathbf{\Lambda}^{p_k}) \preceq \gamma\mathbf{\Omega}$ follows from the well known relationship of the Schur complement similarly as in [141]. Now, apply Lemma C.3 with $\mathbf{M}_1 = (\mathbf{\Lambda}^{p_k})^\top \mathbf{\Omega}(\mathbf{\Lambda}^{p_k})$ and $\mathbf{M}_2 = \gamma\mathbf{\Omega}$ to obtain (3.13). □

Lemma 3.18. *Consider the assumptions in Lemmas 3.16 and 3.17. Moreover, let*

$$V(\boldsymbol{\varrho}_s[k]) := \sqrt{\boldsymbol{\varrho}_s[k]^\top (\mathbf{\Omega} \otimes \mathbf{\Omega})\boldsymbol{\varrho}_s[k]}.$$

Then, the following inequality is satisfied:

$$V(\boldsymbol{\varrho}_s[k+1]) \leq \gamma V(\boldsymbol{\varrho}_s[k]) + \lambda_{\max}(\mathbf{\Omega})\overline{G} \tag{3.14}$$

where $\lambda_{\max}(\mathbf{\Omega})$ is the maximum eigenvalue of $\mathbf{\Omega}$ and

$$\overline{G} := \max_{i \in \{1,\ldots,D\}} \|(\mathbf{L}^{p_k})\mathbf{R}^{p_k}(\mathbf{L}^{p_k})^\top \ + \ \mathbf{W}_d(\Delta^{p_k})\|_F$$

.

PROOF: First, note that since $\mathbf{\Omega}$ is positive definite, henceforth

$$\| \bullet \|_{\mathbf{\Omega}\otimes\mathbf{\Omega}} := \sqrt{(\bullet)^\top (\mathbf{\Omega} \otimes \mathbf{\Omega})(\bullet)}$$

is a norm [123, Page 321] and $V(\boldsymbol{\varrho}_s[k]) \equiv \|\boldsymbol{\varrho}_s[k]\|_{\mathbf{\Omega}\otimes\mathbf{\Omega}}$. Now compute $V(\boldsymbol{\varrho}_s[k+1])$ from (3.12) as:

$$V(\boldsymbol{\varrho}_s[k+1]) = \|(\mathbf{\Lambda}^{p_k} \otimes \mathbf{\Lambda}^{p_k})\boldsymbol{\varrho}_s[k] + \boldsymbol{\omega}[k]\|_{\mathbf{\Omega}\otimes\mathbf{\Omega}} \leq \|(\mathbf{\Lambda}^{p_k} \otimes \mathbf{\Lambda}^{p_k})\boldsymbol{\varrho}_s[k]\|_{\mathbf{\Omega}\otimes\mathbf{\Omega}} + \|\boldsymbol{\omega}[k]\|_{\mathbf{\Omega}\otimes\mathbf{\Omega}}$$

using the triangle inequality [123, Definition 5.1.1-(3)]. Furthermore, use Lemma 3.17 to obtain:

$$\|(\mathbf{\Lambda}^{p_k} \otimes \mathbf{\Lambda}^{p_k})\boldsymbol{\varrho}_s[k]\|_{\mathbf{\Omega}\otimes\mathbf{\Omega}} = \sqrt{\boldsymbol{\varrho}_s[k]^\top \left((\mathbf{\Lambda}^{p_k})^\top \mathbf{\Omega}(\mathbf{\Lambda}^{p_k}) \otimes (\mathbf{\Lambda}^{p_k})^\top \mathbf{\Omega}(\mathbf{\Lambda}^{p_k})\right)\boldsymbol{\varrho}_s[k]}$$
$$\leq \sqrt{\gamma^2 \boldsymbol{\varrho}_s[k]^\top (\mathbf{\Omega} \otimes \mathbf{\Omega})\boldsymbol{\varrho}_s[k]} = \gamma\|\boldsymbol{\varrho}_s[k]\|_{(\mathbf{\Omega}\otimes\mathbf{\Omega})} \equiv \gamma V(\boldsymbol{\varrho}_s[k])$$

Moreover, note that

$$\boldsymbol{\omega}[k]^\top(\boldsymbol{\Omega}\otimes\boldsymbol{\Omega})\boldsymbol{\omega}[k] \leq \lambda_{\max}(\boldsymbol{\Omega}\otimes\boldsymbol{\Omega})\|\boldsymbol{\omega}[k]\|^2$$

by the Rayleigh inequality [123, Theorem 4.2.2]. Furthermore, note that

$$\|\boldsymbol{\omega}[k]\| = \|(\mathbf{L}^{p_k})\mathbf{R}^{p_k}(\mathbf{L}^{p_k})^\top + \mathbf{W}_d(\Delta^{p_k})\|_F.$$

Thus, $\|\boldsymbol{\omega}[k]\|_{\boldsymbol{\Omega}\otimes\boldsymbol{\Omega}} \leq \lambda_{\max}(\boldsymbol{\Omega})\overline{G}$. Then, (3.14) is the combination of the previous results. □

We are now ready to show the rest of Proposition 3.6. First, consider a scalar signal $v[k] \in \mathbb{R}$ satisfying:

$$v[k+1] = \gamma v[k] + \lambda_{\max}(\boldsymbol{\Omega})\overline{G},$$

with $\lambda_{\max}(\boldsymbol{\Omega})\overline{G}$ as in Lemma 3.18, $\gamma \in (0,1)$ and

$$v[0] = \sqrt{\mathsf{vec}(\mathbf{P}_0)^\top(\boldsymbol{\Omega}\otimes\boldsymbol{\Omega})\mathsf{vec}(\mathbf{P}_0)}.$$

It can be verified that the solution to the previous linear difference equation satisfies:

$$v[k] = \gamma^k v[0] + \lambda_{\max}(\boldsymbol{\Omega})\overline{G}\left(\frac{1-\gamma^k}{1-\gamma}\right) \leq v[0] + \frac{\lambda_{\max}(\boldsymbol{\Omega})\overline{G}}{1-\gamma}, \forall k \geq 0$$

In addition, note that

$$v[0] \leq \sqrt{\lambda_{\max}(\boldsymbol{\Omega}\otimes\boldsymbol{\Omega})\mathsf{vec}(\mathbf{P}_0)^\top\mathsf{vec}(\mathbf{P}_0)} = \lambda_{\max}(\boldsymbol{\Omega})\|\mathbf{P}_0\|_F$$

by means of the Rayleigh inequality [123, Theorem 4.2.2] and the identity $\|\mathsf{vec}(\mathbf{P}_0)\| \equiv \|\mathbf{P}_0\|_F$. Combine the previous results to conclude that $v[k] \leq B'_s, \forall k \geq 0$ with

$$B'_s = \lambda_{\max}(\boldsymbol{\Omega})\left(\|\mathbf{P}_0\|_F + \frac{\overline{G}}{1-\gamma}\right).$$

Recall that $V(\boldsymbol{\varrho}_s[k])$ complies (3.14) by Lemma 3.18. Use the comparison lemma in [23, Lemma 13] together with $V(\boldsymbol{\varrho}_s[0]) = v[0]$ to conclude that $V(\boldsymbol{\varrho}_s[k]) \leq v[k], \forall k \geq 0$ which leads directly $V(\boldsymbol{\varrho}_s[k]) \leq B'_s$. Now, Rayleigh inequality is used again to conclude that

$$\lambda_{\min}(\boldsymbol{\Omega})\|\boldsymbol{\varrho}_s[k]\| \leq V(\boldsymbol{\varrho}_s[k]) \leq B'_s$$

equivalently

$$\|\boldsymbol{\varrho}_s[k]\| = \|\hat{\mathbf{P}}_s[k]\|_F \leq \frac{B'_s}{\lambda_{\min}(\boldsymbol{\Omega})} \equiv \frac{B_s}{\sqrt{n_{\mathbf{x}}}}$$

with B_s defined in (3.6). Use the same arguments as in the the proof of Lemma 3.15 to conclude that $\hat{\mathbf{P}}[k] \preceq \hat{\mathbf{P}}_s[k]$ where $\hat{\mathbf{P}}[k]$ is obtained from PLATE in (3.5). Hence, use the same arguments as in the proof of Lemma C.2-b) in C to obtain

$$\|\hat{\mathbf{P}}[k]\|_F \leq \sqrt{n_{\mathbf{x}}}\|\hat{\mathbf{P}}_s[k]\|_F \leq B_s.$$

3.9.4 Proof of Proposition 3.7

Similar to the proof of Proposition 3.5, optimality of the solution of Algorithm 3.4 comes from the dynamic programming principle [124, Page 23]. As for the complexity, note that combining lines 6, 7 and 9 of Algorithm 3.5 results in $\alpha_{\max}Q(\delta)D$ iterations needed to compute $\mathbf{M}_P, \mathbf{M}_J, M_\alpha$. In addition, the number of steps in lines 2 and 5 in Algorithm 3.4 are $Q(\delta)$ and $\alpha_{\max}$ respectively. Hence, the asymptotic complexity of Algorithm 3.4 is only given by the term $\alpha_{\max}Q(\delta)D$.

3.9.5 Proof of Theorem 3.8

In this section, we denote with $\mathbf{F}^{p_k}$ the function that computes $\hat{\mathbf{P}}[k+1] = F^{p_k}(\hat{\mathbf{P}}[k])$ according to (3.5). Moreover, in order to show Theorem 3.8, we provide an auxiliary lemma.

Lemma 3.19. *Let a given $K \in \mathbb{N}$ and consider the systems:*

$$\hat{\mathbf{P}}[k+1] = F^{p_k}(\hat{\mathbf{P}}[k]), \quad \hat{\mathbf{P}}[0] = \mathbf{P}_0$$

$$\hat{\mathbf{P}}'[k+1] = F^{p_k}(\hat{\mathbf{P}}'[k]) + \mathbf{\Pi}_{\mathcal{Q}}[k], \quad \hat{\mathbf{P}}'[0] = \mathbf{P}'_0$$

with $\|\hat{\mathbf{P}}[0] - \hat{\mathbf{P}}'[0]\|_F \leq \delta$ and $\|\mathbf{\Pi}_{\mathcal{Q}}[k]\|_F \leq \delta, \forall k \in \{1, \dots, K\}$. Therefore, for any $\varepsilon' > 0$ there exists $\delta > 0$ such that $\|\hat{\mathbf{P}}[k] - \hat{\mathbf{P}}'[k]\|_F \leq \varepsilon', \forall k \in \{1, \dots, K\}$.

Proof: First, note that for any $\varepsilon_0 > 0$, there exists $\delta > 0$ such that

$$\|\hat{\mathbf{P}}[1] - \hat{\mathbf{P}}'[1]\|_F = \|\mathbf{F}^{p_0}(\hat{\mathbf{P}}[0]) - \mathbf{F}^{p_0}(\hat{\mathbf{P}}'[0]) - \mathbf{\Pi}_{\mathcal{Q}}[0]\|_F \leq \|\mathbf{F}^{p_0}(\hat{\mathbf{P}}[0]) - \mathbf{F}^{p_0}(\hat{\mathbf{P}}'[0])\|_F + \delta \leq \varepsilon_0$$

due to continuity of $\mathbf{F}^{p_0}(\bullet)$. The same reasoning applies for the remaining $K-1$ steps, making $\hat{\mathbf{P}}[k]$ arbitrarily close to $\hat{\mathbf{P}}'[k]$ by choosing $\delta > 0$ sufficiently small. □

Using the previous result, the proof of Theorem 3.8 follows. First note that the initial condition $\mathbf{P}_0$ for the original problem and the quantized version $\hat{\mathbf{P}}_{q_0} = \mathcal{Q}(\mathbf{P}_0)$ comply $\|\mathbf{P}_0 - \hat{\mathbf{P}}_{q_0}\|_F \leq \delta$. Note that if $\hat{\mathbf{P}}[k] \in \{\hat{\mathbf{P}}_1, \dots, \hat{\mathbf{P}}_{Q(\delta)}\}$, then

$$\left\| \mathbf{F}^{p_k}(\hat{\mathbf{P}}[k]) - \mathcal{Q}\left(\mathbf{F}^{p_k}(\hat{\mathbf{P}}[k])\right) \right\|_F \leq \delta$$

Thus, in the quantized setting, $\hat{\mathbf{P}}'[k]$ evolves according to $\hat{\mathbf{P}}'[k+1] = F^{p_k}(\hat{\mathbf{P}}'[k]) + \mathbf{\Pi}_{\mathcal{Q}}[k]$ where $\mathbf{\Pi}_{\mathcal{Q}}[k]$ is the quantization noise complying $\|\mathbf{\Pi}_{\mathcal{Q}}[k]\|_F \leq \delta$ and initial condition $\hat{\mathbf{P}}'[0] = \hat{\mathbf{P}}_{q_0}$. Thus, Lemma 3.19 implies that for any $\varepsilon' > 0$ there is sufficiently small $\delta > 0$ such that true optimal trajectory for the covariance $\hat{\mathbf{P}}[k]$ and the quantized one $\hat{\mathbf{P}}'[k]$ comply $\|\hat{\mathbf{P}}[k] - \hat{\mathbf{P}}'[k]\|_F \leq \varepsilon'$ equivalently $\hat{\mathbf{P}}[k] - \hat{\mathbf{P}}'[k] = \tilde{\mathbf{P}}[k]$ for some $\tilde{\mathbf{P}}[k]$ with $\|\tilde{\mathbf{P}}[k]\|_F \leq \varepsilon'$. Now, with $\tau^+_{k+1} := \min(\tau_{k+1}, T_f)$:

$$\begin{aligned}
|\mathcal{J} - \mathcal{J}_{\mathcal{Q}}(\delta)| &\leq \frac{1}{T_f} \sum_{k=0}^{\mathsf{len}(p)} \int_{\tau_k}^{\tau^+_{k+1}} \left| \mathsf{tr}(\mathbf{A}_{\mathsf{d}}(t-\tau_k)\tilde{\mathbf{P}}[k]\mathbf{A}_{\mathsf{d}}(t-\tau_k)^\top) \right| \mathrm{d}t \\
&\leq \frac{1}{T_f} \sum_{k=0}^{\mathsf{len}(p)} \left(\tau^+_{k+1} - \tau_k\right) \sup_{\tau_k \leq \tau \leq \tau_{k+1}} \left| \mathsf{tr}(\mathbf{A}_{\mathsf{d}}(\tau)\tilde{\mathbf{P}}[k]\mathbf{A}_{\mathsf{d}}(\tau)^\top) \right| \\
&\leq \max_{0 \leq k \leq \mathsf{len}(p)} \sup_{\tau_k \leq \tau \leq \tau_{k+1}} \left| \mathsf{tr}(\mathbf{A}_{\mathsf{d}}(\tau)\tilde{\mathbf{P}}[k]\mathbf{A}_{\mathsf{d}}(\tau)^\top) \right|
\end{aligned}$$

However, since $\|\tilde{\mathbf{P}}[k]\|_F \leq \varepsilon'$, one can choose $\delta > 0$ sufficiently small to make $\varepsilon' > 0$ and as a consequence $\left|\mathsf{tr}(\mathbf{A}_\mathsf{d}(\tau)\tilde{\mathbf{P}}[k]\mathbf{A}_\mathsf{d}(\tau)^\top)\right|$ to be arbitrarily small for any $\tau \in [\tau_k, \tau_{k+1}], k \in \{1, \ldots, \mathsf{len}(p)\}$. Thus, $|\mathcal{J} - \mathcal{J}_\mathcal{Q}(\delta)| \leq \varepsilon$ for some $\delta > 0$.

Chapter Four

Exact Dynamic Consensus (EDC)

In this chapter, we explore scenarios involving the cooperation of multiple robots. Our focus lies in studying the fusion of information from time-varying signals among multiple robots in a distributed manner. We start with the simplest scenario, aiming to compute the average of time-varying exogenous signals, with each robot having its own signal. Here, we assume rapid communication between agents, modeled by continuous-time interactions. However, in subsequent chapters, we introduce additional communication challenges, such as Open Multi-Agent Systems (OMAS) and asynchronous discrete-time communication. Following a modular methodology, we abstract away the control aspects of the robot and the origin of the exogenous signals, which could represent, for instance, the measured target position at each robot. When solely addressing these distributed problems without incorporating the robot model and control, we adopt the term *agent* instead of *robot* for the sake of simplicity and to maintain consistency with the existing literature on this topic. In this context, the architecture considered in this chapter is shown in Figure 4.1.

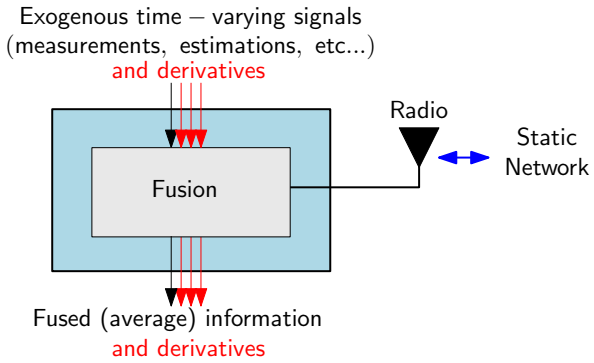

Figure 4.1: Individual agent architecture for distributed information averaging. Red lines denote derivatives, which are important for full-state estimation if the exogenous signals are the output of a state space system, or if the fused information needs to be used to construct time-varying references for a trajectory tracking controller.

In this particular setting, the agent is assumed to have knowledge of the exogenous time-varying signal and its derivatives at all times. Moreover, it can communicate with its neighbors in continuous-time under a static communication network. As an output, the agent obtains the time-varying average of all exogenous signals as well as its derivatives. We consider the following multi-agent system model, which will be used through the rest of the thesis.

Multi-agent system model: the system is composed by N agents. For convenience in the presentation, each of the agents is labeled by an index $i \in \mathcal{I} := \{1, \dots, \mathsf{N}\}$. The communication network between agents is modeled by a graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$, where $\mathcal{I}$ is the node set and $\mathcal{E} \subset \mathcal{I} \times \mathcal{I}$ is the edge set. An edge $(i, j) \in \mathcal{E}$ represents a single communication link between agents $i, j \in \mathcal{I}$.

Each agent i has access to a local signal $z_i(t)$ which is $(m+1)$-times differentiable. The goal is to compute the time varying average signal

$$\bar{z}(t) = \frac{z_1(t) + \cdots + z_{\mathsf{N}}(t)}{\mathsf{N}}$$

in a distributed fashion. Of particular interest are algorithms capable of not only computing $\bar{z}(t)$ but also its derivatives. This becomes particularly useful when $\bar{z}(t)$ is intended for constructing time-varying references for the robots.

If the problem is solved successfully, each agent ends up computing the same signal, leading to consensus among all agents. As a consequence, this problem has been referred to in the literature as Dynamic Consensus (DC), Dynamic Average Consensus (DAC), or Dynamic Average Tracking (DAT).

However, the time-varying nature of the average signal makes this problem particularly challenging. Furthermore, the aspect of differentiation in this context has not been addressed in the existing literature. Therefore, the primary objective of this chapter is to present novel advancements we have introduced in this field. Specifically, we propose a comprehensive framework capable of computing the time-varying average and its derivatives exactly, introducing the concept of *Exact Dynamic Consensus*.

Throughout this chapter, we introduce our main algorithm and provide a formal stability analysis for it. Additionally, we include simulations to validate and demonstrate the advantages of the proposed algorithm. The contributions of this chapter were also published in [1, 7].

4.1 RELATED WORK

Static consensus, where all subsystems (herein referred as agents) manage to agree on a static value such as the average of certain quantities of interest, is a widely studied topic, see for example [142, 143]. On the other hand, consensus towards a time-varying quantity has recently attracted attention due to its potential applications such as distributed formation control [144], distributed unconstrained convex optimization [145], distributed state estimation [146] and distributed resource allocation [147] just to give some examples.

The typical approach, which is widely exposed in [67], relies in a linear protocol. However, in this case, only practical stability towards consensus can be guaranteed, where the accuracy of the steady state depends on the bounds of the derivative of the reference signals, and it is improved as the connectivity is increased.

In DAT applications, dynamic consensus algorithms are used as virtual observers for the average of some time-varying reference signals, which can be tracked by a controller for a local physical system at each agent [84, 148]. Depending on the order of the system it may be desirable for the dynamic consensus observer to obtain derivatives of the average signal [84]. In this context, the works [67, 142, 149] propose linear dynamic consensus algorithms for scalar systems. However, these algorithms have the disadvantage of having a non-zero terminal error bound for some classes of reference signals. This issue has been tackled for scalar systems in [85, 150] and second order systems in [148, 151–153] by means of First Order Sliding Modes (FOSM), allowing exact convergence for more general classes of reference signals. However, these approaches suffer from the so-called chattering effect due to the discontinuous character of the FOSM [86, Chapter 3]. This makes the system sensitive to delays and noise. For the high-order case, some algorithms are able to obtain the average signal and its derivatives with exact convergence for vanishing reference differences in [84] and reference differences with a bounded high-order derivative in [154]. Nonetheless, these approaches impose a higher communication burden since agents share all high-order errors instead of a single scalar.

4.2 PROBLEM STATEMENT

In the following, we describe more formally what we mean by EDC. Assuming that each agent $i \in \mathcal{I}$ has access to its local signal $z_i(t)$, we assume that it is also capable of performing computations, storing memory and communicate with its neighbors according to $\mathcal{G}$. Moreover, we define the *outputs* for each agent as $\hat{z}_{i,0}(t), \ldots, \hat{z}_{i,m}(t)$ which will serve as estimations for $\bar{z}(t)$ and its first m derivatives.

Definition 4.1 (Exact Dynamic Consensus)**.** *The multi-agent system is said to achieve EDC, if there exists $T > 0$ such that the individual output signals for each agent reach*

$$\hat{z}_{1,\mu}(t) = \hat{z}_{2,\mu}(t) = \cdots = \hat{z}_{n,\mu}(t) = \bar{z}^{(\mu)}(t)$$

$\forall t \geq t_0 + T$, $\forall \mu \in \{0, \ldots, m\}$ *and*

$$\bar{z}(t) = \frac{1}{\mathsf{N}} \sum_{i=1}^{\mathsf{N}} z_i(t)$$

Problem 4.2 (High order exact average consensus)**.** *Given the set of local signals $z_1(t)$, $\ldots$, $z_{\mathsf{N}}(t)$, the problem consists in designing a distributed one-hop algorithm such that the multi-agent system achieve EDC.*

4.3 THE EDCHO ALGORITHM

The EDCHO (EDC of High Order) algorithm proposed in this chapter to obtain EDC has the following structure:

Protocol:

$$\begin{aligned}\dot{\chi}_{i,\mu}(t) &= k_\mu \sum_{j\in\mathcal{N}_i} \lceil \hat{z}_{i,0}(t) - \hat{z}_{j,0}(t) \rfloor^{\frac{m-\mu}{m+1}} + \chi_{i,\mu+1}(t), \quad 0 \le \mu < m \\ \dot{\chi}_{i,m}(t) &= k_m \sum_{j\in\mathcal{N}_i} \lceil \hat{z}_{i,0}(t) - \hat{z}_{j,0}(t) \rfloor^{0} \end{aligned} \tag{4.1}$$

Output:

$$\hat{z}_{i,\mu}(t) = z_i^{(\mu)}(t) - \chi_{i,\mu}(t).$$

where $i \in \{1,\dots,2\mathsf{N}\}$. Hence, each agent has an internal state $\boldsymbol{\chi}_i = [\chi_{i,0},\dots,\chi_{i,m}]^\top$, and shares only $\hat{z}_{i,0}$ to its neighbors, in contrast to sharing all $\hat{z}_{i,0},\dots,\hat{z}_{i,m}$ which is not necessary, reducing communication load. In this chapter, we consider the following assumption.

Assumption 4.3. *The initial conditions for* (4.1) *are set to be such that*

$$\sum_{i=1}^{\mathsf{N}} \chi_{i,\mu}(t_0) = 0, \forall \mu \in \{0,\dots,m\}.$$

Note that Assumption 4.3 is trivially satisfied without the need of any global information if all agents set $\chi_{i,\mu}(t_0) = 0, \forall \mu \in \{0,\dots,m\}$. However, this assumption does not allow connection or disconnection of agents as in an OMAS context. An extension to account for this case is considered in Chapter 5. Moreover, the algorithm depends on the the gains $k_0,\dots,k_m > 0$ which will be designed as described later in order for (4.1) to achieve EDC provided that the following assumption holds.

Assumption 4.4. *The time varying signals* $z_1(t),\dots,z_{\mathsf{N}}(t)$ *all satisfy*

$$\left| \bar{z}^{(m+1)}(t) - z_i^{(m+1)}(t) \right| \le L, \forall t \ge t_0$$

with a known $L > 0$.

Remark 4.5. *Note that* (4.1) *is a high order system with discontinuous right hand side. Hence, solutions to* (4.1) *are properly understood in the sense of Filippov [155]. This is,* (4.1) *is studied as a differential inclusion with* $\lceil 0 \rfloor^0 = [-1,1]$.

Remark 4.6. *Note that if* $m = 0$, (4.1) *resembles some of the basic results proposed in [85], hence being subsumed by the approach in this chapter.*

The following is the main result of this chapter, which states that there exist a non-empty set of possible values for the gains $k_0,\dots,k_m$ such that (4.1) achieves EDC.

Theorem 4.7. *Let Assumptions 4.3 and 4.4. Moreover, let*

$$k_\mu = \lambda_\mu k_{\mu-1}^{\frac{m-\mu}{m-(\mu-1)}}$$

for $\mu = 1, \dots, m$ with $\lambda_1, \dots, \lambda_m$ parameters chosen such that system (E.1) *is finite time stable for $\theta = 0$. Therefore, there exists sufficiently large $k_0 > 0$ and $T > 0$ such that the EDC property is achieved for* (4.1).

The proof of Theorem 4.7 can be found in Section 4.7 after some needed results which are developed in the following sections.

4.4 TOWARDS CONVERGENCE OF EDCHO

First, we provide some results which are required to show that (4.1) achieves EDC. As it will be evident latter, it is convenient to write (4.1) with a different set of gains per edge, just as a mere tool for the proof. This is, let $\mathbf{K}_\mu = \text{diag}([k_{1,\mu}, \dots, k_{\ell,\mu}]), \forall \mu \in \{0, \dots, m\}$ where ℓ is the number of edges. Then, the modified version of (4.1) is

$$\begin{aligned} \dot{\boldsymbol{\chi}}_\mu(t) &= \boldsymbol{\chi}_{\mu+1}(t) + \mathbf{D}\mathbf{K}_\mu \left\lceil \mathbf{D}^\top \hat{\mathbf{z}}_0(t) \right\rfloor^{\frac{m-\mu}{m+1}} \quad \text{for } 0 \leq \mu \leq m-1, \\ \dot{\boldsymbol{\chi}}_m(t) &= \mathbf{D}\mathbf{K}_m \left\lceil \mathbf{D}^\top \hat{\mathbf{z}}_0(t) \right\rfloor^0 \\ \hat{\mathbf{z}}_\mu(t) &= \mathbf{z}^{(\mu)}(t) - \boldsymbol{\chi}_\mu(t) \end{aligned} \tag{4.2}$$

where

$$\boldsymbol{\chi}_\mu(t) = \begin{bmatrix} \chi_{1,\mu}(t) \\ \vdots \\ \chi_{n,\mu}(t) \end{bmatrix}, \quad \hat{\mathbf{z}}_\mu(t) = \begin{bmatrix} \hat{z}_{1,\mu}(t) \\ \vdots \\ \hat{z}_{n,\mu}(t) \end{bmatrix}, \quad \mathbf{z}(t) = \begin{bmatrix} z_1(t) \\ \vdots \\ z_{\mathsf{N}}(t) \end{bmatrix}$$

and $\mathbf{D}$ is the incidence matrix of $\mathcal{G}$. Moreover, Assumption 4.4 implies

$$\mathbf{P}\mathbf{z}^{(m+1)}(t) \in [-L, L]^{\mathsf{N}}$$

with

$$\mathbf{P} = (\mathbf{I} - (1/\mathsf{N})\mathbb{1}\mathbb{1}^\top).$$

The following is an interesting property of (4.2) which basically states that under Assumption 4.3, the trajectories of (4.2) are orthogonal to $\mathbb{1}$.

Lemma 4.8. *Under Assumption 4.3, the following identity is satisfied for* (4.1)*:*

$$\mathbb{1}^\top \boldsymbol{\chi}_\mu(t) = 0, \quad \forall t \geq t_0, \quad \forall \mu \in \{0, \dots, m\}$$

Proof: Denote $s_\mu = \mathbb{1}^\top \boldsymbol{\chi}_\mu$. We proceed by induction: let $\mu = m$ as induction base

$$\dot{s}_m = \mathbb{1}^\top \dot{\boldsymbol{\chi}}_m = -\mathbb{1}^\top \mathbf{D}\mathbf{K}_m \left\lceil \mathbf{D}^\top \hat{\mathbf{z}}_0 \right\rfloor^0 = 0.$$

Hence, the value of $s_m(t) = s_m(t_0) = 0$ remains constant $\forall t \geq t_0$ under Assumption 4.3. Now, assume $s_{\mu+1}(t) = s_{\mu+1}(t_0) = 0, \forall \mu \in \{0, \dots, m-1\}$ remains constant $\forall t \geq t_0$, then,

$$\begin{aligned} \dot{s}_\mu =& \mathbb{1}^\top \dot{\boldsymbol{\chi}}_\mu = \mathbb{1}^\top \left(\boldsymbol{\chi}_{\mu+1}(t) + \mathbf{D}\mathbf{K}_\mu \left\lceil \mathbf{D}^\top \hat{\mathbf{z}}_0 \right\rfloor^{\frac{m-\mu}{m+1}} \right) \\ =& \mathbb{1}^\top \boldsymbol{\chi}_{\mu+1}(t_0) + \mathbb{1}^\top \mathbf{D}\mathbf{K}_\mu \left\lceil \mathbf{D}^\top \hat{\mathbf{z}}_0 \right\rfloor^{\frac{m-\mu}{m+1}} = 0 \end{aligned}$$

where Assumption 4.3 was used. Then, $\mathbb{1}^\top \boldsymbol{\chi}_\mu(t) = 0, \forall \mu \in \{0, \dots, m\}$, $\forall t \geq t_0$ which concludes the proof. □

It also can be shown that if protocol (4.2) converges, it will converge to a state which complies with the EDC property. To show so, let $\tilde{\mathbf{z}}_\mu(t) = \mathbf{P}\hat{\mathbf{z}}_\mu(t)$ with $\mathbf{P} = \left(\mathbf{I} - (1/\mathsf{N})\mathbb{1}\mathbb{1}^\top\right)$. Then, its dynamics are given by

$$\begin{aligned}\dot{\tilde{\mathbf{z}}}_\mu(t) &= \tilde{\mathbf{z}}_{\mu+1}(t) - \mathbf{D}\mathbf{K}_\mu \left\lceil \mathbf{D}^\top \tilde{\mathbf{z}}_0(t) \right\rfloor^{\frac{m-\mu}{m+1}} \quad \text{for } 0 \leq \mu \leq m-1, \\ \dot{\tilde{\mathbf{z}}}_m(t) &= \mathbf{P}\mathbf{z}^{(m+1)}(t) - \mathbf{D}\mathbf{K}_m \left\lceil \mathbf{D}^\top \tilde{\mathbf{z}}_0(t) \right\rfloor^0\end{aligned} \tag{4.3}$$

Corollary 4.9. *If there exists $T > 0$ such that the state $\tilde{\mathbf{z}}_\mu(t) = 0, \forall \mu \in \{0, \ldots, m\}$, $\forall t \geq t_0 + T$ is reached, then (4.2) achieves EDC.*

PROOF: If $\tilde{\mathbf{z}}_\mu(t) = 0$, then

$$\hat{\mathbf{z}}_\mu(t) = \frac{1}{\mathsf{N}}\mathbb{1}\mathbb{1}^\top \hat{\mathbf{z}}_\mu(t) = \frac{1}{\mathsf{N}}\mathbb{1}\mathbb{1}^\top(\mathbf{z}^{(\mu)}(t) - \boldsymbol{\chi}_\mu(t)) = \bar{z}^{(\mu)}\mathbb{1}$$

by Lemma 4.8. □

4.5 CONTRACTION PROPERTY OF EDCHO

In this section we show the so called contraction property as described in [96]. This property states that there exists a non-empty set of gains $\mathbf{K}_0, \ldots, \mathbf{K}_m$ such that trajectories of $\tilde{\mathbf{z}}_\mu(t), \forall \mu \in \{0, \ldots, m\}$ gather arbitrary close to the origin in an arbitrary small amount of time. First, we show that it is indeed the case for tree graphs, and then we use this result to show contraction for arbitrary connected graphs.

4.5.1 Contraction for tree graphs

According to Proposition D.4 in Appendix D, it is always possible to write $\tilde{\mathbf{z}}_\mu(t) = \mathbf{D}\boldsymbol{\sigma}_\mu(t)$ for some $\boldsymbol{\sigma}_\mu(t) \in \mathbb{R}^\ell$ where ℓ is the number of edges in $\mathcal{G}$. Now, consider that $\mathcal{G}$ is a tree graph. Note that, in this case, from Proposition D.2-(4) in Appendix D, $s_{\min}(\mathbf{D}) > 0$ since the flow space of a tree graph has dimension 0 and therefore $\mathbf{D}^\top\mathbf{D}$ is a full rank matrix. Thus, by writing $\mathbf{z}(t) = \bar{z}(t)\mathbb{1} + \mathbf{D}\tilde{\mathbf{z}}(t)$, then under Assumption 4.4,

$$\tilde{\mathbf{z}}^{(m+1)}(t) \in [-\tilde{L}, \tilde{L}]^\ell$$

with $\tilde{L} = \sqrt{\mathsf{N}}L/s_{\min}(\mathbf{D})$ by Proposition C.5 in Appendix C.

In the following, we study the behaviour of the system, introducing in (4.3) the change $\tilde{\mathbf{z}}_\mu(t) = \mathbf{D}\boldsymbol{\sigma}_\mu(t)$ to obtain

$$\begin{aligned}\dot{\boldsymbol{\sigma}}_\mu(t) &= \boldsymbol{\sigma}_{\mu+1}(t) - \mathbf{K}_\mu \left\lceil \mathbf{D}^\top\mathbf{D}\boldsymbol{\sigma}_0(t) \right\rfloor^{\frac{m-\mu}{m+1}}, \quad \text{for } 0 \leq \mu \leq m-1 \\ \dot{\boldsymbol{\sigma}}_m(t) &= \tilde{\mathbf{z}}^{(m+1)}(t) - \mathbf{K}_m \left\lceil \mathbf{D}^\top\mathbf{D}\boldsymbol{\sigma}_0(t) \right\rfloor^0\end{aligned} \tag{4.4}$$

By Corollary 4.9, if $\boldsymbol{\sigma}_\mu = 0, \forall \mu \in \{0, \ldots, m\}$ is reached, then EDC is achieved. Before showing contraction of (4.4) we provide some auxiliary results. Write (4.4) in the recursive

form

$$\mathcal{H}_m \begin{cases} \dot{\boldsymbol{\sigma}}_0(t) &= \boldsymbol{\sigma}_1(t) - \boldsymbol{\Lambda}_0 \left\lceil \mathbf{D}^\top \mathbf{D} \boldsymbol{\sigma}_0(t) \right\rfloor^{\frac{m}{m+1}} \\ \dot{\boldsymbol{\sigma}}_1(t) &= \boldsymbol{\sigma}_2(t) - \boldsymbol{\Lambda}_1 \left\lceil \boldsymbol{\sigma}_1(t) - \dot{\boldsymbol{\sigma}}_0(t) \right\rfloor^{\frac{m-1}{m}} \\ &\vdots \\ \dot{\boldsymbol{\sigma}}_\mu(t) &= \boldsymbol{\sigma}_{\mu+1}(t) - \boldsymbol{\Lambda}_\mu \left\lceil \boldsymbol{\sigma}_\mu(t) - \dot{\boldsymbol{\sigma}}_{\mu-1}(t) \right\rfloor^{\frac{m-\mu}{m-(\mu-1)}} \\ &\vdots \\ \dot{\boldsymbol{\sigma}}_m(t) &= -\boldsymbol{\Lambda}_m \left\lceil \boldsymbol{\sigma}_m(t) - \dot{\boldsymbol{\sigma}}_{m-1}(t) \right\rfloor^0 + \tilde{\mathbf{z}}^{(m+1)}(t) \end{cases} \tag{4.5}$$

by using the fact that

$$\boldsymbol{\sigma}_\mu(t) - \dot{\boldsymbol{\sigma}}_{\mu-1}(t) = \mathbf{K}_{\mu-1} \left\lceil \mathbf{D}^\top \mathbf{D} \boldsymbol{\sigma}_0(t) \right\rfloor^{\frac{m-(\mu-1)}{m+1}}$$

and defining

$$\mathbf{K}_\mu = \boldsymbol{\Lambda}_\mu \mathbf{K}_{\mu-1}^{\frac{m-\mu}{m-(\mu-1)}}$$

for $\mu = 1, \ldots, m$ and $\mathbf{K}_0 = \boldsymbol{\Lambda}_0$. This change introduces some advantages because the dynamics in $\mathcal{H}_m$ are decoupled for each component of $\boldsymbol{\sigma}_\mu(t)$, and most importantly the dynamics of each component of $\boldsymbol{\sigma}_\mu(t)$ correspond exactly to the Levant's differentiator error system in recursive form (E.1). Hence, by showing that $\dot{\boldsymbol{\sigma}}_0(t)$ resembles the properties of the signal θ in Appendix E, contraction towards the origin for $\boldsymbol{\sigma}_\mu(t), \mu \geq 1$ is guaranteed.

Lemma 4.10. *For any $\delta > 0$, $\Omega_0, \ldots, \Omega_m > 0$ and any trajectory of (4.5) starting from $\|\boldsymbol{\sigma}_\mu(t_0)\| \leq \Omega_\mu, \forall \mu \in \{0, \ldots, m\}$, there exists $K > 0$ such that*

$$\int_{t_0}^{t_0+\delta} \|\dot{\boldsymbol{\sigma}}_0(t)\| \mathrm{d}t < K.$$

Proof: Choose an arbitrary $\tau > \delta$. Hence, the trajectory of (4.5) for $t \in [t_0, t_0 + \tau]$ satisfying $\|\boldsymbol{\sigma}_\mu(t_0)\| \leq \Omega_\mu$, also satisfy that $\|\boldsymbol{\sigma}_\mu(t)\| \leq \bar{\Omega}_\mu$ in $t \in [t_0, t_0 + T]$ for some unknown bounds $\bar{\Omega}_\mu$. Moreover, denote with $s_{\boldsymbol{\Lambda}} = s_{\max}(\boldsymbol{\Lambda}_0)$. Therefore,

$$\begin{aligned} \int_{t_0}^{t_0+\delta} \|\dot{\boldsymbol{\sigma}}_0(t)\| \mathrm{d}t &\leq \int_{t_0}^{t_0+\delta} \ell^{\frac{1}{2m+2}} s_{\boldsymbol{\Lambda}} \left(s_{\max}(\mathbf{D}^\top \mathbf{D}) \right)^{\frac{m}{m+1}} \|\boldsymbol{\sigma}_0(t)\|^{\frac{m}{m+1}} + \|\boldsymbol{\sigma}_1(t)\| \mathrm{d}t \\ &\leq \int_{t_0}^{t_0+\delta} \ell^{\frac{1}{2m+2}} s_{\boldsymbol{\Lambda}} \left(s_{\max}(\mathbf{D}^\top \mathbf{D}) \right)^{\frac{m}{m+1}} \bar{\Omega}_0^{\frac{m}{m+1}} + \bar{\Omega}_1 \mathrm{d}t \\ &= \left(\ell^{\frac{1}{2m+2}} s_{\boldsymbol{\Lambda}} \left(s_{\max}(\mathbf{D}^\top \mathbf{D}) \right)^{\frac{m}{m+1}} \bar{\Omega}_0^{\frac{m}{m+1}} + \bar{\Omega}_1 \right) \delta = K \end{aligned}$$

where Corollary C.4-(1) was used with $\alpha = \frac{m}{m+1}$ and Proposition C.5 in Appendix C to introduce $s_{\boldsymbol{\Lambda}}$ and $s_{\max}(\mathbf{D}^\top \mathbf{D})$. □

The utility of Lemma 4.10 is that we can use Proposition E.1 to fix a desired bound for $\boldsymbol{\sigma}_1(t)$ in the first equation of (4.5) at least for a desired time interval $[t_0, t_0 + \delta]$. Then, we can treat $\boldsymbol{\sigma}_1(t)$ as a disturbance with known bound, and focus our attention to designing $\lambda_0 := \min \text{ diag } \boldsymbol{\Lambda}_0$ such that $\boldsymbol{\sigma}_0(t)$ reaches an arbitrarily small vicinity of the origin before that interval ends. This is shown in the following:

Lemma 4.11. *Let $\mathcal{G}$ be a tree and*

$$\dot{\boldsymbol{\sigma}}_0(t) = \mathbf{d}(t) - \boldsymbol{\Lambda}_0 \left\lceil \mathbf{D}^\top \mathbf{D}\boldsymbol{\sigma}_0 \right\rfloor^{\frac{m}{m+1}} \tag{4.6}$$

$\boldsymbol{\sigma}_0(t), \mathbf{d}(t) \in \mathbb{R}^\ell$, $\overline{d} > 0, \Omega_0 > 0$, and the bounds $\|\mathbf{D}\boldsymbol{\sigma}_0(t_0)\| \leq \Omega_0$, $\|\mathbf{d}(t)\| \leq \overline{d}, \forall t \in [t_0, +\infty)$. Then, for any $\overline{\delta} > 0$ and any $0 < \omega_0 < \Omega_0$ there exists $0 < \overline{\lambda}_0 = \min \text{ diag } \boldsymbol{\Lambda}_0$ (sufficiently big) such that $\|\mathbf{D}\boldsymbol{\sigma}_0(t)\| \leq \omega_0, \forall t \in [t_0 + \overline{\delta}, +\infty)$.

Proof: First, let the consensus error $\boldsymbol{\xi} = \mathbf{D}\boldsymbol{\sigma}_0$. Then,

$$\dot{\boldsymbol{\xi}}(t) = \mathbf{D}\mathbf{d}(t) - \mathbf{D}\boldsymbol{\Lambda}_0 \left\lceil \mathbf{D}^\top \boldsymbol{\xi}(t) \right\rfloor^{\frac{m}{m+1}}.$$

Choose the Lyapunov function candidate $V(\boldsymbol{\xi}) = (1/2)\boldsymbol{\xi}^\top \boldsymbol{\xi}$. Hence, in the interval starting from $\|\boldsymbol{\xi}(t_0)\| \geq \omega_0$ and in which $\|\boldsymbol{\xi}(t)\| \geq \omega_0$ is maintained, the following is satisfied

$$\begin{aligned}
\dot{V} &= \boldsymbol{\xi}^\top \dot{\boldsymbol{\xi}} = \boldsymbol{\xi}^\top \left(\mathbf{D}\mathbf{d} - \mathbf{D}\boldsymbol{\Lambda}_0 \left\lceil \mathbf{D}^\top \boldsymbol{\xi} \right\rfloor^{\frac{m}{m+1}} \right) \leq -\overline{\lambda}_0 \boldsymbol{\xi}^\top \mathbf{D} \left\lceil \mathbf{D}^\top \boldsymbol{\xi} \right\rfloor^{\frac{m}{m+1}} + \|\boldsymbol{\xi}\| \|\mathbf{D}\mathbf{d}\| \\
&\leq -\overline{\lambda}_0 \|\mathbf{D}^\top \boldsymbol{\xi}\|^{\frac{m}{m+1}+1} + s_{\max}(\mathbf{D}) \|\boldsymbol{\xi}\| \|\mathbf{d}\| \leq -\omega_0 \left(\overline{\lambda}_0 (s_{\min}(\mathbf{D})\omega_0)^{\frac{m}{m+1}+1} - s_{\max}(\mathbf{D})\overline{d} \right) \\
&\leq -\eta
\end{aligned}$$

where Corollary C.4-(2) and Proposition C.5 were used, and by choosing

$$\overline{\lambda}_0 \geq (s_{\min}(\mathbf{D})\omega_0)^{\frac{-m}{m+1}-1} (\eta \omega_0^{-1} + s_{\max}(\mathbf{D})\overline{d})$$

for any $\eta > 0$. Henceforth, V will decay towards the origin with rate η until the condition $\|\boldsymbol{\xi}\| \geq \omega_0$ is no longer maintained. Hence, in order to reach such condition before the interval $[t_0, t_0 + \overline{\delta}]$ ends, choose

$$\eta \geq \overline{\delta}^{-1} (V(\boldsymbol{\xi}(t_0)) - (1/2)\omega_0^2)$$

for any $\overline{\delta} > 0$. Therefore, by the comparison Lemma [129, Lemma 3.4], $\dot{V} \leq -\eta$ implies

$$V(\boldsymbol{\xi}(t)) \leq V(\boldsymbol{\xi}(t_0)) - \eta(t - t_0) \leq V(\boldsymbol{\xi}(t_0)) - \eta\overline{\delta} \leq (1/2)\omega_0^2$$

for $t \in [t_0, t_0 + T]$ where $t_0 + T \leq t_0 + \overline{\delta}$ is the moment in which $\|\boldsymbol{\xi}(t_0 + T)\| = \omega_0$. Then, the condition $\|\boldsymbol{\xi}(t)\| = \|\mathbf{D}\boldsymbol{\sigma}_0(t)\| \leq \omega_0$ will be reached and maintained $\forall t \in [t_0 + \overline{\delta}, +\infty)$ concluding the proof. □

We also show that $\dot{\boldsymbol{\sigma}}_0(t)$ can be driven towards an arbitrarily small vicinity of the origin. Hence, $\dot{\boldsymbol{\sigma}}_0$ can play the role of θ in the results from Appendix E.

Lemma 4.12. *Let $\mathcal{G}$ be a tree, consider system (4.6) under the same conditions from Lemma 4.11 and the additional condition that there exists $\tilde{d} > 0$ such that $\|\dot{\mathbf{d}}(t)\| \leq \tilde{d}, \forall t \in [t_0, +\infty)$. Then, for any $\tilde{\delta} > 0$ and any $\tilde{\omega}_0 > 0$ there exists $0 < \tilde{\lambda}_0 = \min \text{ diag } \boldsymbol{\Lambda}_0$ (sufficiently big) such that $\|\dot{\boldsymbol{\sigma}}_0(t)\| \leq \tilde{\omega}_0, \forall t \in [t_0 + \tilde{\delta}, +\infty)$.*

Proof: Let $\gamma_i(t)$ be the i-th component of $\mathbf{D}^\top \mathbf{D}\boldsymbol{\sigma}_0(t)$. Then, by the fact that

$$\frac{\mathrm{d}}{\mathrm{d}t} \left\lceil \gamma_i(t) \right\rfloor^{\frac{m}{m+1}} = \frac{m}{m+1} |\gamma_i(t)|^{\frac{m}{m+1}-1} \dot{\gamma}_i(t),$$

hence,

$$\frac{\mathrm{d}}{\mathrm{d}t}\left\lceil \mathbf{D}^\top \mathbf{D}\boldsymbol{\sigma}_0(t)\right\rfloor^{\frac{m}{m+1}} = \frac{m}{m+1}\begin{bmatrix} |\gamma_1(t)|^{\frac{m}{m+1}-1}\dot{\gamma}_1(t) \\ \vdots \\ |\gamma_\ell(t)|^{\frac{m}{m+1}-1}\dot{\gamma}_\ell(t)\end{bmatrix} = \frac{m}{m+1}\mathbf{J}(t)\mathbf{D}^\top\mathbf{D}\dot{\boldsymbol{\sigma}}_0(t)$$

where

$$\mathbf{J}(t) = \mathrm{diag}\left(\left[|\gamma_1(t)|^{\frac{m}{m+1}-1},\ldots,|\gamma_\ell(t)|^{\frac{m}{m+1}-1}\right]\right).$$

Now, let change of variables $\boldsymbol{\zeta} = \mathbf{D}\dot{\boldsymbol{\sigma}}_0$ which leads to

$$\dot{\boldsymbol{\zeta}}(t) = \mathbf{D}\dot{\mathbf{d}}(t) - \frac{m}{m+1}\mathbf{D}\boldsymbol{\Lambda}_0\mathbf{J}(t)\mathbf{D}^\top\boldsymbol{\zeta}(t)$$

Additionally let $\overline{\lambda}_0 \leq \tilde{\lambda}_0$ with $\overline{\lambda}_0$ chosen such that

$$\|\mathbf{D}^\top\mathbf{D}\boldsymbol{\sigma}_0(t)\| \leq \omega_0, \forall t \in [t_0+\overline{\delta}, +\infty)$$

from Lemma 4.11. Then, each component $\gamma_i(t)$ will satisfy

$$|\gamma_i(t)|^{\frac{m}{m+1}-1} \geq \omega_0^{\frac{m}{m+1}-1}, \forall t \in [t_0+\overline{\delta}, +\infty)$$

since $\frac{m}{m+1}-1<0$. Choose the Lyapunov function $V(\boldsymbol{\zeta}) = (1/2)\boldsymbol{\zeta}^\top\boldsymbol{\zeta}$ and an arbitrary $Z>0$. Hence, in the interval starting from $\|\boldsymbol{\zeta}(t_0)\| \geq Z$ and in which $\|\boldsymbol{\zeta}(t)\| \geq Z$ is maintained, the following is satisfied

$$\begin{aligned}
\dot{V} &= \boldsymbol{\zeta}^\top\left(\mathbf{D}\dot{\mathbf{d}} - \frac{m}{m+1}\mathbf{D}\boldsymbol{\Lambda}_0\mathbf{J}\mathbf{D}^\top\boldsymbol{\zeta}\right) \leq s_{\max}(\mathbf{D})\|\boldsymbol{\zeta}\|\|\dot{\mathbf{d}}\| - \frac{\tilde{\lambda}_0 m}{m+1}\boldsymbol{\zeta}^\top\mathbf{D}\mathbf{J}\mathbf{D}^\top\boldsymbol{\zeta} \\
&\leq s_{\max}(\mathbf{D})\tilde{d}\|\boldsymbol{\zeta}\| - \frac{\tilde{\lambda}_0 m}{m+1}c(\mathcal{G})\omega_0^{\frac{m}{m+1}-1}\|\boldsymbol{\zeta}\|^2 \leq -\|\boldsymbol{\zeta}\|\left(\frac{\tilde{\lambda}_0 m}{m+1}c(\mathcal{G})\omega_0^{\frac{m}{m+1}-1}\|\boldsymbol{\zeta}\| - s_{\max}(\mathbf{D})\tilde{d}\right) \\
&\leq -Z\left(\frac{\tilde{\lambda}_0 m}{m+1}c(\mathcal{G})\omega_0^{\frac{m}{m+1}-1}Z - s_{\max}(\mathbf{D})\tilde{d}\right) \\
&\leq -\eta
\end{aligned}$$

by the fact that

$$\boldsymbol{\zeta}(t)^\top\mathbf{D}\mathbf{J}(\mathbf{t})\mathbf{D}^\top\boldsymbol{\zeta}(t) = \omega_0^{\frac{m}{m+1}-1}\boldsymbol{\zeta}(t)\mathbf{D}\mathbf{D}^\top\boldsymbol{\zeta}(t) \geq \omega_0^{\frac{m}{m+1}-1}c(\mathcal{G})\|\boldsymbol{\zeta}(t)\|^2$$

using Proposition D.2-(2), with $c(\mathcal{G})$ as the algebraic connectivity of $\mathcal{G}$ and by choosing

$$\tilde{\lambda}_0 \geq \max\left\{\overline{\lambda}_0, \frac{m+1}{mZc(\mathcal{G})}\omega_0^{1-\frac{m}{m+1}}\left(s_{\max}(\mathbf{D})\tilde{d} + Z^{-1}\eta\right)\right\}$$

for any $\eta>0$. From this point, the proof follows exactly as the proof of Lemma 4.11 to conclude that $\|\boldsymbol{\zeta}(t)\| \leq Z$ will be reached an maintained for $t \in [t_0+\tilde{\delta}, +\infty)$ for any $Z, \tilde{\delta} > 0$. Hence, since $\mathcal{G}$ is a tree, we can choose $Z = \tilde{\omega}_0/s_{\min}(\mathbf{D})$ and obtain

$$\|\dot{\boldsymbol{\sigma}}_0(t)\| \leq s_{\min}(\mathbf{D})^{-1}\|\mathbf{D}\dot{\boldsymbol{\sigma}}_0(t)\| \leq s_{\min}(\mathbf{D})^{-1}\|\boldsymbol{\zeta}(t)\| \leq \tilde{\omega}_0,$$

which concludes the proof. □

Using these results, we provide the proof of the contraction property for tree graphs.

Lemma 4.13. *Consider* (4.4) *and* $\mathcal{G}$ *to be a tree. Then, for any* $0 < \omega_\mu < \Omega_\mu$, $T > 0$, *there exists some gain matrices* $\mathbf{K}_0, \dots, \mathbf{K}_m$ *(with sufficiently big diagonal entries) such that any trajectory of* (4.4) *satisfying* $\|\boldsymbol{\sigma}_\mu(t_0)\| \leq \Omega_\mu$ *will satisfy* $\|\boldsymbol{\sigma}_\mu(t)\| \leq \omega_\mu, \forall t \in [T, +\infty), \forall \mu \in \{0, \dots, m\}$.

Proof: Let $\dot{\boldsymbol{\sigma}}_\mu(t) = [\dot{\sigma}_{\mu,1}(t), \dots, \dot{\sigma}_{\mu,n}(t)]^\top$. Since $|\dot{\sigma}_{\mu,i}(t)| \leq \|\dot{\boldsymbol{\sigma}}_\mu(t)\|$ then, from Lemma 4.10 we know that for any δ there exists $K > 0$ such that

$$\int_{t_0}^{t_0+\delta} |\dot{\sigma}_{0,i}(t)| \mathrm{d}t \leq \int_{t_0}^{t_0+\delta} \|\dot{\boldsymbol{\sigma}}_0(t)\| \mathrm{d}t \leq K.$$

Henceforth, from Propositions E.1 and E.2 in Appendix E we can choose an arbitrary Ω'_μ with $\Psi_\mu > \Omega'_\mu > \Omega_\mu$ such that there exists $\delta > 0$ for which

$$\|\boldsymbol{\sigma}_\mu(t)\| \leq \Omega'_\mu \leq \Psi_\mu, \forall t \in [t_0, t_0 + \delta].$$

Moreover,

$$\|\dot{\boldsymbol{\sigma}}_1\| \leq \|\boldsymbol{\sigma}_2\| + s_{\max}(\mathbf{K}_1)\ell^{\frac{1}{2m+2}} \|\mathbf{D}^\top \mathbf{D}\boldsymbol{\sigma}_0\|^{\frac{m-1}{m+1}} \leq \Psi_2 + s_{\max}(\mathbf{K}_1)\ell^{\frac{1}{2m+2}} (s_{\max}(\mathbf{D}^\top \mathbf{D})\Psi_1)^{\frac{m-1}{m+1}}.$$

Therefore, both $\|\boldsymbol{\sigma}_1(t)\|$ and $\|\dot{\boldsymbol{\sigma}}_1(t)\|$ remain bounded in the interval $t \in [t_0, t_0+\delta)$ and will remain bounded (by the same bounding constants) in $t \in [t_0+\delta, +\infty)$ by Proposition E.2 provided that $\|\dot{\boldsymbol{\sigma}}_0(t)\| \leq \tilde{\omega}_0, \forall t \in [t_0+\delta, +\infty)$ and sufficiently small $\tilde{\omega}_0$. Choose $0 < \tilde{\delta} < \delta$. Hence, we can identify $\mathbf{d}(t) = \boldsymbol{\sigma}_1$ from Lemma 4.11 and Lemma 4.12 and choose $\tilde{\lambda}_0 = \min \operatorname{diag} \boldsymbol{\Lambda}_0 = \min \operatorname{diag} \mathbf{K}_0$ big enough such to obtain $\|\dot{\boldsymbol{\sigma}}_0(t)\| \leq \tilde{\omega}_0, \forall t \in [t_0 + \delta, +\infty)$ and $\|\mathbf{D}\boldsymbol{\sigma}_0(t)\| \leq \omega_0, \forall t \in [t_0 + \delta, +\infty)$, obtaining the contraction for $\|\boldsymbol{\sigma}_0\|$. Contraction for $\|\boldsymbol{\sigma}_\mu(t)\|, \mu > 0$ follows directly from Proposition E.2, since $\tilde{\mathbf{z}}^{(m+1)} \in [-\tilde{L}, \tilde{L}]^\ell$, and by adjusting $\tilde{\omega}_0 < \overline{\theta}$, concluding the proof . □

4.5.2 Contraction for general connected graphs

Now, in order to show the same contraction property but for general graphs, consider the following setting. Let $\mathcal{G}_A$ and $\mathcal{G}_B$ be two graphs with corresponding incidence matrices $\mathbf{D}_A = [\tilde{\mathbf{D}}_A, \mathbf{D}_s]$ and $\mathbf{D}_B = [\tilde{\mathbf{D}}_B, \mathbf{D}_s]$ where $\mathbf{D}_s$ corresponds to the edges which appear in both $\mathcal{G}_A$ and $\mathcal{G}_B$. Suppose that protocol (4.2) works for each of the graphs. Then, we aim to conclude that the protocol works for their union by means of switching between them, and applying the averaging principle. However, in average, the edges that appear in both graphs contribute twice to the protocol. Hence, we take advantage of the different gains per-edge to attenuate such contribution. This is, choose some gain matrices $\mathbf{K}^A_\mu$ and $\mathbf{K}^B_\mu, \forall \mu \in \{0, \dots, m\}$ for each graph to implement protocol (4.2) in the following form: $\mathbf{K}^A_\mu = \operatorname{blockdiag}(2\tilde{\mathbf{K}}^A_\mu, \mathbf{K}^s_\mu)$ and $\mathbf{K}^B_\mu = \operatorname{blockdiag}(2\tilde{\mathbf{K}}^B_\mu, \mathbf{K}^s_\mu)$ where $\mathbf{K}^s_\mu$ corresponds to gains for the edges in $\mathbf{D}_s$ and $2\tilde{\mathbf{K}}^A_\mu, 2\tilde{\mathbf{K}}^B_\mu$ for $\tilde{\mathbf{D}}_A, \tilde{\mathbf{D}}_B$ respectively. Now, to study the switching between $\mathcal{G}_A$ and $\mathcal{G}_B$, let

$$\mathbf{F}_\mu(t, \tilde{\mathbf{z}}_0; \varepsilon) = \begin{cases} \mathbf{D}_A \mathbf{K}^A_\mu \left\lceil \mathbf{D}_A^\top \tilde{\mathbf{z}}_0 \right\rfloor^{\frac{m-\mu}{m+1}}, & t - t_0 \in [0, \varepsilon/2) \\ \mathbf{D}_B \mathbf{K}^B_\mu \left\lceil \mathbf{D}_B^\top \tilde{\mathbf{z}}_0 \right\rfloor^{\frac{m-\mu}{m+1}}, & t - t_0 \in [\varepsilon/2, \varepsilon) \\ \mathbf{F}_\mu(t - \varepsilon, \tilde{\mathbf{z}}_0; \varepsilon), & t - t_0 \geq \varepsilon \end{cases}$$

and write the dynamics of $\tilde{\mathbf{z}}_\mu$ for this switching protocol as a differential inclusion,

$$\begin{aligned}\dot{\tilde{\mathbf{z}}}_\mu(t) &= \tilde{\mathbf{z}}_{\mu+1}(t) - \mathbf{F}_\mu(t, \tilde{\mathbf{z}}_0(t); \varepsilon), \quad \text{for } 0 \leq \mu \leq m-1 \\ \dot{\tilde{\mathbf{z}}}_m(t) &\in [-L, L]^{\mathsf{N}} - \mathbf{F}_m(t, \tilde{\mathbf{z}}_0(t); \varepsilon)\end{aligned} \tag{4.7}$$

since $\mathbf{P}\mathbf{z}^{(m+1)} \in [-L, L]^{\mathsf{N}}$ by Assumption 4.4. Note that explicit dependence of time in (4.7) comes only from the terms of the form $\mathbf{F}_\mu(t, \tilde{\mathbf{z}}_0; \varepsilon)$ and therefore from switching. Now, we obtain the average system, by averaging the right hand side of (4.7) in the interval $t - t_0 \in [0, \varepsilon)$. Note that terms of the form $\mathbf{F}_\mu(t, \tilde{\mathbf{z}}_0; \varepsilon)$ are averaged as

$$\begin{aligned}\frac{1}{\varepsilon}\int_{t_0}^{t_0+\varepsilon} \mathbf{F}_\mu(t, \tilde{\mathbf{z}}_0; \varepsilon)\mathrm{d}t &= \frac{1}{2}\left(\mathbf{D}_A \mathbf{K}_\mu^A \left\lceil \mathbf{D}_A^\top \tilde{\mathbf{z}}_0 \right\rfloor^{\frac{m-\mu}{m+1}} + \mathbf{D}_B \mathbf{K}_\mu^B \left\lceil \mathbf{D}_B^\top \tilde{\mathbf{z}}_0 \right\rfloor^{\frac{m-\mu}{m+1}}\right) \\ &= \mathbf{D}_{AB} \mathbf{K}_\mu^{AB} \left\lceil \mathbf{D}_{AB}^\top \tilde{\mathbf{z}}_0 \right\rfloor^{\frac{m-\mu}{m+1}}\end{aligned}$$

where $\mathbf{D}_{AB} = [\tilde{\mathbf{D}}_A, \tilde{\mathbf{D}}_B, \mathbf{D}_s]$ is the incidence matrix of the superposition of $\mathcal{G}_A$ and $\mathcal{G}_B$, and $\mathbf{K}_\mu^{AB} = \mathrm{blockdiag}(\tilde{\mathbf{K}}_\mu^A, \tilde{\mathbf{K}}_\mu^B, K_\mu^s)$. From, this we obtain the following conclusion about (4.7) with respect to the averaged version of it.

Lemma 4.14. *Let $\tilde{\mathbf{z}}_\mu(t), 0 \leq \mu \leq m$ be a solution of (4.7) with initial conditions $\tilde{\mathbf{z}}_\mu(t_0)$ and let the averaged system*

$$\begin{aligned}\dot{\tilde{\mathbf{z}}}_\mu^a(t) &= \tilde{\mathbf{z}}_{\mu+1}^a(t) + \mathbf{D}_{AB}\mathbf{K}_\mu^{AB} \left\lceil \mathbf{D}_{AB}^\top \tilde{\mathbf{z}}_0^a(t) \right\rfloor^{\frac{m-\mu}{m+1}} \quad \text{for } 0 \leq \mu \leq m-1, \\ \dot{\tilde{\mathbf{z}}}_m^a(t) &\in [-L, L]^{\mathsf{N}} + \mathbf{D}_{AB}\mathbf{K}_m^{AB} \left\lceil \mathbf{D}_{AB}^\top \tilde{\mathbf{z}}_0^a(t) \right\rfloor^0\end{aligned} \tag{4.8}$$

with $\tilde{\mathbf{z}}_\mu^a(t_0) = \tilde{\mathbf{z}}_\mu(t_0)$. Then, for any $r > 0$, there exists $R > 0, \varepsilon > 0$ such that

$$\|\tilde{\mathbf{z}}_\mu(t) - \tilde{\mathbf{z}}_\mu^a(t)\| \leq r, \quad \forall t \in [t_0, t_0 + R/\varepsilon)$$

Proof: Note that the right hand sides of (4.7) and (4.8) are locally Lipschitz. Following from [155], every locally Lipschitz function at a point is one-sided Lipshitz in a neighborhood of such point. Hence, rigorous justification of the averaging argument comes from the Bogoliubov's first theorem for one-sided Lipschitz differential inclusions [156, Section 2.2]. □

Using this result, we show contraction for general graphs.

Lemma 4.15. *Consider (4.4) and $\mathcal{G}$ to an arbitrary connected graph. Then, for any $0 < \omega_\mu < \Omega_\mu$, $T > 0$, there exists some gain matrices $\mathbf{K}_0, \ldots, \mathbf{K}_m$ (sufficiently big) such that any trajectory of (4.4) satisfying $\|\tilde{\mathbf{z}}_\mu(t_0)\| \leq \Omega_\mu$ will satisfy $\|\tilde{\mathbf{z}}_\mu(t)\| \leq \omega_\mu, \forall t \in [T, +\infty), \forall \mu \in \{0, \ldots, m\}$.*

Proof: In order to show the result for graphs $\mathcal{G}$ let N be the dimension of its flow space and proceed by induction. The induction base with $N = 0$ is shown in Lemma 4.13. Now, assume that the result is true for graphs with flow space of dimension $N-1$ with contraction neighborhood of radius $\omega_\mu(N-1)$ and $\mathcal{G}_{AB}$ be any graph with flow space dimension N. Then, by Proposition D.3 there exists two connected graphs $\mathcal{G}_A$ and $\mathcal{G}_B$ with flow space dimension $N-1$ whose union corresponds to $\mathcal{G}_{AB}$. Choose $\omega_\mu(N-1) = \omega_\mu(N) - r$ with arbitrary $0 < r < \omega_\mu(N)$ such that there exists $\mathbf{K}_\mu^A$ and

$\mathbf{K}_\mu^B$ for $\mathcal{G}_A$ and $\mathcal{G}_B$ respectively and $\|\tilde{\mathbf{z}}_\mu(t)\| \leq \omega_\mu(N-1), \forall t \geq T$ with $T \leq \varepsilon/2$ for each of the two networks by the assumption about the $N-1$ case. Hence, since both schemes contract to an arbitrarily small neighborhood of the origin before the switching instants at $t-t_0 = \varepsilon/2$ and $t-t_0 = \varepsilon$, the same conclusion applies for the switching system (4.7) before $t - t_0 = \varepsilon$. Contraction for $\mathcal{G}_{AB}$ comes from Lemma 4.14 since (4.8) corresponds to the dynamics in (4.3) for such graph, and the bound $\|\tilde{\mathbf{z}}_\mu^a(t)\| \leq \omega_\mu(N-1) + r = \omega_\mu(N)$. □

4.6 PARAMETER DESIGN FOR EDCHO

By inspecting the results from the previous sections, in particular the proof of Lemma 4.13, it can be noticed that the parameters needed for (4.1) to reach consensus are closely related to the parameters used for a Levant's differentiator to converge. In fact, all parameters, except for k_0 can be found using this reasoning, simplifying the parameter design methodology. This is shown in the following Corollary:

Corollary 4.16. *Let $\lambda_1, \ldots, \lambda_m$ be parameters chosen such that* (E.1) *is finite time stable for $\theta = 0$. Thus, there exists $k_0 > 0$ large enough such that the conclusion of Lemma 4.15 follows with $k_\mu = \lambda_\mu k_{\mu-1}^{\frac{m-\mu}{m-(\mu-1)}}$ for $\mu = 1, \ldots, m$.*

PROOF: First, consider the case of tree graphs. The proof follows directly from the fact that (4.4) can be written recursively as (4.5). The result is then a consequence of the reasoning in Section 4.5.1, where the last m equations of (4.5) correspond precisely to a vector form of (E.1). This leaves only the condition that $k_0 > 0$ needs to be large enough. The case of general graphs is no different, since the gains used in such scheme can be chosen the same as the ones for tree graphs, as long as the contraction time is small enough from the arguments of the proof of Lemma 4.15. However, increasing k_0 decreases such contraction time too, which concludes the proof. □

Note that finding feasible sequences of parameters $\lambda_1, \ldots, \lambda_\mathsf{N}$ for (E.1) is by now a well studied topic in the literature. In fact, not only in the original work [96] some feasible parameters were found by computer simulation for $L = 1$, but also other works such as [157] give different possible values by means of a Lyapunov function condition. Hence, these parameters can be consulted and used directly, scaled appropriately for any $L > 0$. Moreover, motivated by the methodology in [96], k_0 can be found by computer simulation for a concrete topology, by incrementally searching for an appropriate $k_0 > 0$ until convergence is obtained.

4.7 CONVERGENCE OF EDCHO

In this section, we show the proof of Theorem 4.7. The proof follows by contraction and by noticing that trajectories of (4.2) are invariant to a particular transformation, referred as the homogeneity in [96].

Lemma 4.17. *Let $\eta > 0$. Then, the trajectories of* (4.3) *are preserved by the transformation $(t, \tilde{\mathbf{z}}_\mu) \mapsto (\eta t, \eta^{m-(\mu-1)} \tilde{\mathbf{z}}_\mu)$.*

PROOF: Let $t' = \eta t$ and $\tilde{\mathbf{z}}'_\mu(t') = \eta^{m-(\mu-1)}\tilde{\mathbf{z}}_\mu(\eta t)$. Then, for $\mu = 0, \ldots, m-1$,

$$\begin{aligned}\frac{\mathrm{d}\tilde{\mathbf{z}}'_\mu}{\mathrm{d}t'} &= \eta^{m-(\mu-1)}\frac{\mathrm{d}\tilde{\mathbf{z}}_\mu}{\mathrm{d}t'} = \eta^{m-\mu}\dot{\tilde{\mathbf{z}}}_\mu \\ &= \eta^{m-\mu}\left(\eta^{-(m-\mu)}\tilde{\mathbf{z}}'_{\mu+1} - \mathbf{D}\mathbf{K}_\mu\left\lceil \mathbf{D}^\top \eta^{-(m+1)}\tilde{\mathbf{z}}'_0 \right\rfloor^{\frac{m-\mu}{m+1}}\right) \\ &= \tilde{\mathbf{z}}'_{\mu+1} - \mathbf{D}\mathbf{K}_\mu\left\lceil \mathbf{D}^\top \tilde{\mathbf{z}}'_0 \right\rfloor^{\frac{m-\mu}{m+1}}\end{aligned}$$

Similarly, for $\mu = m$ it is obtained

$$\frac{\mathrm{d}\tilde{\mathbf{z}}'_m}{\mathrm{d}t'} \in [-L, L]^{\mathsf{N}} - \mathbf{D}\mathbf{K}_m\left\lceil \mathbf{D}^\top \tilde{\mathbf{z}}'_0 \right\rfloor^0.$$

Then, trajectories $\tilde{\mathbf{z}}'_\mu(t')$ are equivalent to $\tilde{\mathbf{z}}_\mu(t)$. □

Similarly as the work in [96], both contraction and homogeneity of (4.4) can be used to produce sequential contractions towards an equilibrium point, reaching it in a finite amount of time.

PROOF: (Of Theorem 4.7) Let (4.2) with $\mathbf{K}_\mu = k_\mu \mathbf{I}$ recovering (4.1). Then, Lemma 4.15 implies that for sufficiently large $k_0, \ldots, k_m > 0$ there exists a finite time $T_c > 0$ such that if $\|\tilde{\mathbf{z}}_\mu(t_0)\| \leq \Omega_\mu$ then $\|\tilde{\mathbf{z}}_\mu(t_0+T_c)\| < \kappa\Omega_\mu$ for any $0 \leq \kappa < 1$. Then, from Lemma 4.17 the similar contraction follows: if $\|\tilde{\mathbf{z}}_\mu(t_0)\| \leq \eta^{m-\mu+1}\Omega_\mu$ then $\|\tilde{\mathbf{z}}_\mu(t_0+\eta T_c)\| \leq \kappa\eta^{m-\mu+1}\Omega_\mu$. Hence, for $0 < \eta < 1$ choose $\kappa = \eta^{m-\mu+1}$. Therefore, convergence is shown in a sequence of countable steps for $\nu = 0, 1, \ldots$ of sequential contraction. For $\nu = 0$, let $\|\tilde{\mathbf{z}}_\mu(t_0)\| \leq \Omega_\mu$ be contracted to

$$\|\tilde{\mathbf{z}}_\mu(t_0 + T_c)\| \leq \eta^{m-\mu+1}\Omega_\mu.$$

Then, for $\nu = 1$,

$$\|\tilde{\mathbf{z}}_\mu(t_0 + T_c)\| \leq \eta^{m-\mu+1}$$

is contracted to

$$\|\tilde{\mathbf{z}}_\mu(t_0 + T_c + \eta T_c)\| \leq \eta^{2(m-\mu+1)}\Omega_\mu.$$

Furthermore, for any ν, the contraction

$$\left\|\tilde{\mathbf{z}}_\mu(t_0 + T_c(1 + \eta + \eta^2 + \cdots + \eta^\nu))\right\| \leq \eta^{(\nu+1)(m-\mu+1)}\Omega_\mu$$

is obtained. Hence, $\lim_{\nu\to+\infty}\eta^{(\nu+1)(m-\mu+1)} = 0$ and

$$\lim_{\nu\to+\infty} T_c(1 + \eta + \eta^2 + \cdots + \eta^\nu) = \frac{T_c}{1-\eta}$$

using the geometric series. Therefore, with $T = T_c/(1-\eta)$, $\|\tilde{\mathbf{z}}_\mu(t)\| = 0, \forall t \in [t_0+T, +\infty)$. Moreover, convergence towards the EDC property follows from the conclusion of Corollary 4.9. Finally, using the parameter design from Corollary 4.16 concludes the proof. □

4.8 SIMULATION EXAMPLES

For the purpose of demonstrating the advantages of the proposal, a simulation scenario is described here with the following configuration. There are $n = 8$ agents connected by a graph $\mathcal{G}$ shown in Figure 4.2. The EDCHO algorithm (4.1) was implemented using explicit Euler method with time step $h = 10^{-4}$. In this example we use $m = 3$ and the gains k_μ are chosen as $7.5, 19.25, 17.75, 7$ for all agents.

Moreover, consider initial conditions $\chi_{i,\mu}(0) = 0,\ \forall \mu > 0$ and $\chi_{i,0}(0)$ given by [18.69, −4.17, −2.02, −1.49, −4.65, −4.52, 0.16, −2.00] respectively. Note that this initial conditions comply with Assumption 4.3.

In the first experiment, each agent has internal reference signals

$$z_i(t) = a_i \cos(\omega_i t + \phi_i), i = 1, \ldots, 8$$

with amplitudes a_i of [0.99, 0.27, 0.02, 0.48, 0.18, 0.24, 0.65, 0.50], frequencies ω_i of [2.44, 1.70, 1.12, 0.26, 0.68, 1.73, 2.33, 0.02] and phases ϕ_i of [1.25, 1.92, 6.25, 3.82, 0.70, 7.63, 9.93, 6.80]. The individual trajectories for this experiment are shown in Figure 4.3, as well as the target $\bar{z}(t)$ and its derivatives in red. Note that all agents are able to track not only $\bar{z}(t)$ but also $\dot{\bar{z}}(t), \ddot{\bar{z}}(t)$ and $\bar{z}^{(3)}(t)$. The magnitudes of $\tilde{\mathbf{z}}_0, \ldots, \tilde{\mathbf{z}}_3$ are shown in 4.4-(Up), where it can be noted that exact convergence is achieved. Compare this with the behaviour of a linear protocol [67, Equation (11)] and the first order sliding mode (FOSM) protocol in [85, Equation (5)] under the same conditions. As shown in Figure 4.4-(Up), the linear protocol achieves only bounded steady state error whereas the FOSM achieves exact convergence in the input and its derivatives. However, both linear and FOSM approaches are only able to track $\bar{z}(t)$ and not its derivatives.

Now, consider the reference signals as $z_i(t) = a_i t^3, i = 1, \ldots, 8$ with a_i of [0.99, 0.27, 0.02, 0.48, 0.18, 0.24, 0.65, 0.50]. The error convergence is shown in Figure 4.4-(Bottom) where EDCHO achieves exact convergence whereas both the linear and the FOSM protocol errors grow to infinity.

All previous experiments were conducted by simulating the algorithms using Euler's discretization with small step size of $\Delta t = 10^{-6}$ in order to inspect their performance as close as possible to their continuous time theoretical version. Increasing the value of Δt have an effect on the performance of both EDCHO and FOSM protocols due to discretization errors and chattering. Thus, we repeated the experiment with sinusoidal reference signals using $\Delta t = 10^{-3}$ instead, in order to make this effects more apparent. The results of this experiment are shown in Figure 4.5 for both $\Delta t = 10^{-6}$ and $\Delta t = 10^{-3}$. The parameters of the FOSM were chosen such to roughly match the settling time of EDCHO for the sake of fairness. Note that in both cases the error signal $\|\tilde{\mathbf{z}}_0(t)\|$ for EDCHO is almost one order of magnitude less than the FOSM protocol. Moreover, it can be noted that the chattering effect is almost negligible for EDCHO when compared to FOSM which greatly suffers from it. Additionally, the accuracy in steady state degrades as Δt is increased, specially for the higher order signals $\|\tilde{\mathbf{z}}_\mu(t)\|, \mu > 0$ as expected from HOSM systems [96, Theorem 7].

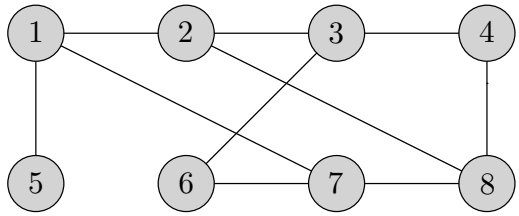

Figure 4.2: The graph $\mathcal{G}$ considered in the examples

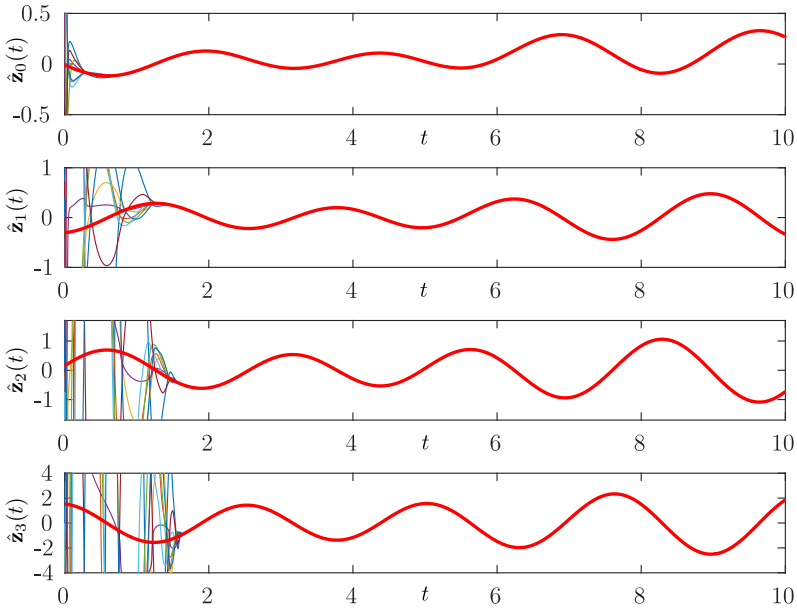

Figure 4.3: Components of the vectors $\hat{\mathbf{z}}_0(t), \ldots, \hat{\mathbf{z}}_3(t)$ as well as $\bar{z}(t), \dot{\bar{z}}(t), \ddot{\bar{z}}(t)$ and $\bar{z}^{(3)}(t)$ in red.

4.9 DISCUSSION

In this chapter, the EDCHO algorithm has been presented, where the agents are able to maintain zero steady-state consensus error, when tracking the average of time-varying signals and its derivatives. EDCHO works under reasonable assumptions about the initial conditions and bounds of certain high order derivatives of the reference signals. An in-depth study on its stability characteristics was provided from which a simple design procedure arises. The simulation scenario presented here, exposes the effectiveness of our approach in addition to show its advantages when compared to other approaches.

EDCHO serves as a main building block for cooperation and information fusion in further developments in this thesis. While the current assumptions for EDCHO are still restrictive, we will consider robustness to agent connection and disconnection, noise, asynchronous discrete-time communication, and delays in subsequent chapters. Similarly, in Chapters 7 and 8, we will explore how EDCHO fits in the context of perception latency

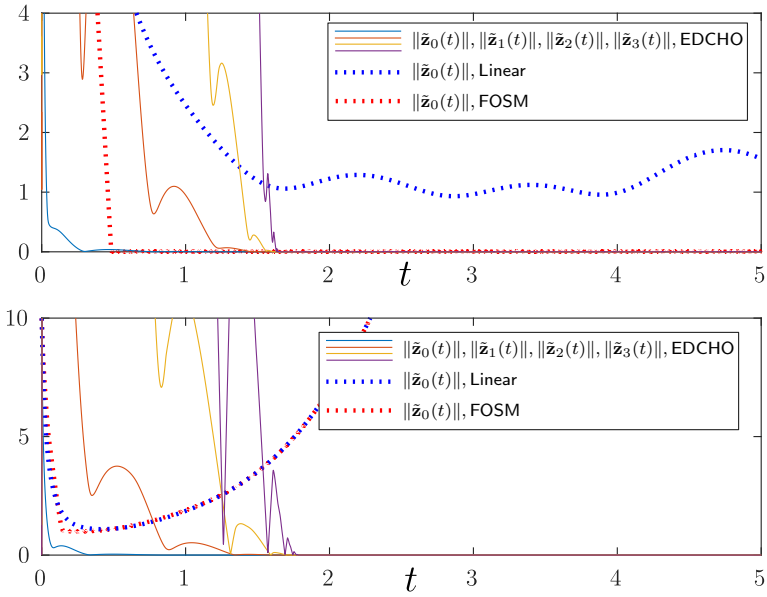

Figure 4.4: Comparison of the magnitude of $\tilde{\mathbf{z}}_\mu(t)$ for EDCHO with a similar measure for a linear protocol and a first order sliding mode (FOSM) in the case of (Up) sinusoidal and (Bottom) polynomial references signals.

scheduling.

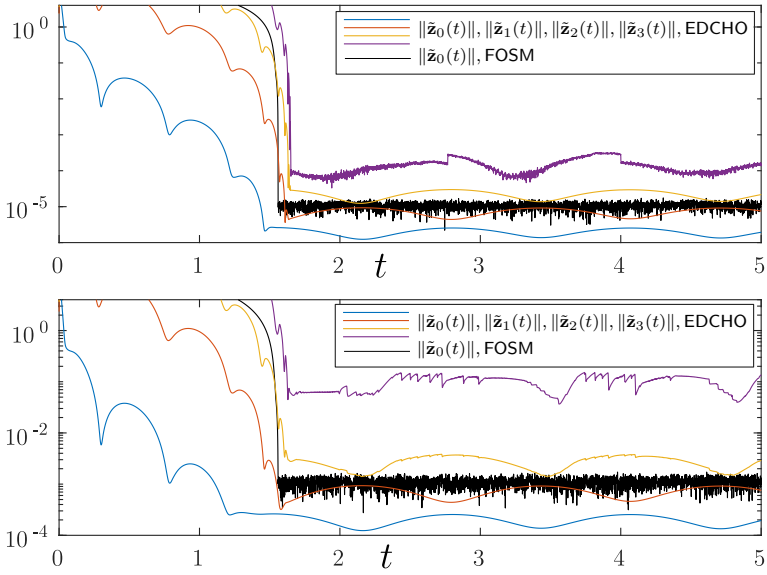

Figure 4.5: Comparison of the magnitude of $\tilde{\mathbf{z}}_\mu(t)$ for EDCHO with a similar measure for the first order sliding mode (FOSM) in the case of sampling steps of $\Delta t = 10^{-6}$ (Up) and $\Delta t = 10^{-3}$ (Bottom) sinusoidal references signals. To improve clarity, this figure is plotted in logarithmic vertical axis

Chapter Five

Robust EDC For Open Networks

In this chapter, we build upon the discussion initiated in Chapter 4 concerning the development of exact dynamic consensus algorithms. As mentioned previously, the EDCHO protocol successfully achieves EDC under certain conditions, such as noiseless time-varying signals, continuous-time communications, and a static graph $\mathcal{G}$. However, these conditions are limiting, motivating the need for further design in more general settings.

In particular, this chapter focuses on scenarios where agents can connect or disconnect from the network at arbitrary points in time, enabling an OMAS setting. The challenge in this setting is that Assumption 4.3 from Chapter 4, which pertains to the initial conditions of EDCHO, is no longer valid. With agents joining or leaving, new initial conditions and a transient for consensus emerge.

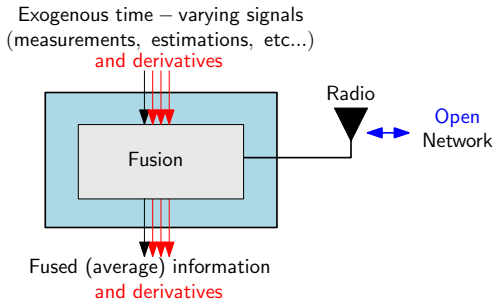

Figure 5.1: Individual agent architecture for distributed information averaging.

The goal of this chapter is to extend EDCHO into what we term Robust EDCHO (REDCHO). The objective is to eliminate the need for Assumption 4.3 from Chapter 4 and accommodate the connection and disconnection of agents from the network. Similarly as in Chapter 4, we provide a formal stability analysis for REDCHO as well as illustrative simulation examples. The contributions of this chapter were also published in [2].

5.1 RELATED WORK AND PROBLEM STATEMENT

First, we extend the definition of EDC in Chapter 4 in order to emphasize the features of an OMAS we are interested in this chapter:

Definition 5.1 (Robust Exact Dynamic Consensus)**.** *Let the average signal* $\bar{z}(t) := \frac{1}{\mathsf{N}}\sum_{i=1}^{\mathsf{N}} z_i(t)$*. Then, the multi-agent system is said to achieve robust EDC, if the individual output signals for each agent comply*

$$\lim_{t\to\infty}\left|\hat{z}_{i,\mu}(t) - \bar{z}^{(\mu)}(t)\right| = 0,$$

for all $\mu \in \{0,\dots,m\}, i \in \{1,\dots,\mathsf{N}\}$ *regardless of isolated events of spontaneous connection or disconnection of agents.*

It is important to note that the problem of dealing with robustness to connection and disconnection of agents, hereby violating Assumption 4.3 is known in the literature for particular protocols. In fact, dedicated discussions for this issue can be found for linear protocols in [67] and FOSM in [85].

The proposed approach involves adding linear damping terms using the internal memory variable ($\chi_{i,\mu}(t)$ in this thesis), which enables the asymptotic vanishing of the initial condition mismatch. The convergence analysis with these damping terms can be carried out using standard techniques for linear systems such as pole placement analysis. Additionally, the convergence analysis for the nonlinear FOSM protocol is feasible using a simple Lyapunov function due to the first-order nature of the algorithm.

However, it is important to note that these convergence analysis ideas cannot be directly applied to EDCHO, as demonstrated in this chapter. While REDCHO introduces linear damping terms into EDCHO to achieve robustness to spontaneous connection or disconnection of agents from the network, similar to solutions in the literature, the main contribution of this chapter lies in its non-trivial convergence analysis. The complexity in the analysis results from the high-order and nonlinear nature of the proposed protocol, making the application of linear systems theory or the identification of an appropriate Lyapunov function challenging.

Therefore, REDCHO offers similar advantages as EDCHO when compared to linear protocols and FOSM. Additionally, REDCHO introduces a new feature that enables basic OMAS by allowing for connection or disconnection of agents from the network.

5.2 REDCHO

We propose a new algorithm, REDCHO, which manages to achieve robust EDC under mild assumptions on the reference signals. To present the algorithm, first let the auxiliary matrices

$$\mathbf{\Gamma} = \begin{bmatrix} -\gamma_0 & 1 & 0 & \cdots & 0 \\ 0 & -\gamma_1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & -\gamma_{m-1} & 1 \\ 0 & \cdots & \cdots & 0 & -\gamma_m \end{bmatrix}, \mathbf{G} = \begin{bmatrix} \mathbf{C} \\ \mathbf{C\Gamma} \\ \vdots \\ \mathbf{C\Gamma}^m \end{bmatrix} \tag{5.1}$$

for design parameters $\gamma_0, \ldots, \gamma_m > 0$ and $\mathbf{C} = [1, 0, \ldots, 0] \in \mathbb{R}^{1\times(m+1)}$. The structure of the REDCHO algorithm is proposed as

Protocol:

$$\begin{aligned}
\dot{\chi}_{i,\mu}(t) &= k_\mu \theta^{\mu+1} \sum_{j\in\mathcal{N}_i} \lceil \hat{z}_{i,0}(t) - \hat{z}_{j,0}(t) \rfloor^{\frac{m-\mu}{m+1}} + \chi_{i,\mu+1}(t) - \gamma_\mu \chi_{i,\mu}(t), \quad 0 \le \mu < m \\
\dot{\chi}_{i,m}(t) &= k_m \theta^{m+1} \sum_{j\in\mathcal{N}_i} \lceil \hat{z}_{i,0}(t) - \hat{z}_{j,0}(t) \rfloor^{0} - \gamma_m \chi_{i,m}(t)
\end{aligned}$$

Output:

$$\hat{z}_{i,\mu}(t) = z_i^{(\mu)}(t) - \sum_{\nu=0}^{m} \mathbf{G}_{\mu+1,\nu+1} \chi_{i,\nu}(t). \tag{5.2}$$

where $\mathbf{G}_{\mu+1,\nu+1}$ with $\mu, \nu \in \{0, \ldots, m\}$ are the components of the matrix $\mathbf{G}$ and $\theta \ge 1$ is a design parameter. Furthermore, we consider the following assumption:

Assumption 5.2. *Let*

$$s_i(t) = \left(\bar{z}^{(m+1)}(t) - z_i^{(m+1)}(t)\right) + \sum_{\mu=0}^{m} l_\mu \left(\bar{z}^{(\mu)}(t) - z_i^{(\mu)}(t)\right)$$

where $l_0, \ldots, l_m$ are the coefficients of the polynomial

$$(\lambda + \gamma_0) \cdots (\lambda + \gamma_m) = \lambda^{m+1} + \sum_{\mu=0}^{m} l_\mu \lambda^\mu.$$

Thus, $|s_i(t)| \le L, \forall t \ge t_0$ for fixed $\gamma_0, \ldots, \gamma_m$ and known $L > 0$.

It is easy to show that EDCHO is a particular limiting case of REDCHO and can be recovered by choosing $\gamma_0 = \cdots = \gamma_m = 0$ and $\theta = 1$.

5.3 TOWARDS CONVERGENCE OF REDCHO

The REDCHO algorithm (5.2) can be written in partially vectorized form as:

$$\begin{aligned}
\dot{\boldsymbol{\chi}}_\mu(t) &= \boldsymbol{\chi}_{\mu+1}(t) + k_\mu \theta^{\mu+1} \mathbf{D} \left\lceil \mathbf{D}^\top \hat{\mathbf{z}}_0(t) \right\rfloor^{\frac{m-\mu}{m+1}} - \gamma_\mu \boldsymbol{\chi}_\mu(t) \quad \text{for } 0 \le \mu < m, \\
\dot{\boldsymbol{\chi}}_m(t) &= k_m \theta^{m+1} \mathbf{D} \left\lceil \mathbf{D}^\top \hat{\mathbf{z}}_0(t) \right\rfloor^{0} - \gamma_m \boldsymbol{\chi}_m(t) \\
\hat{\mathbf{z}}_\mu(t) &= \mathbf{z}^{(\mu)}(t) - \sum_{\nu=0}^{m} \mathbf{G}_{\mu+1,\nu+1} \boldsymbol{\chi}_\nu(t), \forall \mu \in \{0, \ldots, m\}
\end{aligned}$$

where we define

$$\boldsymbol{\chi}_\mu(t) := \begin{bmatrix} x_{1,\mu}(t) \\ \vdots \\ x_{n,\mu}(t) \end{bmatrix}, \quad \hat{\mathbf{z}}_\mu(t) := \begin{bmatrix} \hat{z}_{1,\mu}(t) \\ \vdots \\ \hat{z}_{\mathsf{N},\mu}(t) \end{bmatrix}, \quad \mathbf{z}(t) := \begin{bmatrix} z_1(t) \\ \vdots \\ z_{\mathsf{N}}(t) \end{bmatrix}$$

and $\mathbf{D}$ is the incidence matrix of $\mathcal{G}$. Moreover, let

$$\boldsymbol{\chi}(t) := \begin{bmatrix} \boldsymbol{\chi}_0(t) \\ \vdots \\ \boldsymbol{\chi}_m(t) \end{bmatrix}, \quad \hat{\mathbf{z}}(t) := \begin{bmatrix} \hat{\mathbf{z}}_0(t) \\ \vdots \\ \hat{\mathbf{z}}_m(t) \end{bmatrix}, \quad \mathbf{z}_{\mathsf{e}}(t) := \begin{bmatrix} \left(\mathbf{z}^{(0)}(t)\right) \\ \vdots \\ \left(\mathbf{z}^{(m)}(t)\right) \end{bmatrix}$$

and

$$\mathbf{F}(\hat{\mathbf{z}}_0(t);\theta) = \begin{bmatrix} k_0\theta\mathbf{D}\left\lceil \mathbf{D}^\top \hat{\mathbf{z}}_0(t)\right\rfloor^{\frac{m}{m+1}} \\ \vdots \\ k_m\theta^{m+1}\mathbf{D}\left\lceil \mathbf{D}^\top \hat{\mathbf{z}}_0(t)\right\rfloor^{0} \end{bmatrix}$$

Using this notation we obtain the fully vectorized form of the algorithm:

$$\begin{aligned} \dot{\boldsymbol{\chi}}(t) &= (\boldsymbol{\Gamma} \otimes \mathbf{I}_{\mathsf{N}})\boldsymbol{\chi}(t) + \mathbf{F}(\hat{\mathbf{z}}_0(t);\theta) \\ \hat{\mathbf{z}}(t) &= \mathbf{z}_{\mathsf{e}}(t) - (\mathbf{G} \otimes \mathbf{I}_{\mathsf{N}})\boldsymbol{\chi}(t) \end{aligned}$$

Both partial and fully vectorized versions of the algorithm will be used throughout this chapter. Moreover, note that $\mathbf{G}$ is the observability matrix of the pair $(\boldsymbol{\Gamma}, \mathbf{C})$ which is invertible. Then, the dynamics of $\hat{\mathbf{z}}(t)$ result in

$$\begin{aligned} \dot{\hat{\mathbf{z}}}(t) &= \dot{\mathbf{z}}_{\mathsf{e}}(t) - (\mathbf{G} \otimes \mathbf{I}_{\mathsf{N}})(\boldsymbol{\Gamma} \otimes \mathbf{I}_{\mathsf{N}})\boldsymbol{\chi}(t) - (\mathbf{G} \otimes \mathbf{I}_{\mathsf{N}})\mathbf{F}(\hat{\mathbf{z}}_0(t);\theta) \\ &= \dot{\mathbf{z}}_{\mathsf{e}}(t) + (\mathbf{G} \otimes \mathbf{I}_{\mathsf{N}})(\boldsymbol{\Gamma} \otimes \mathbf{I}_{\mathsf{N}})(\mathbf{G}^{-1} \otimes \mathbf{I}_{\mathsf{N}})(\hat{\mathbf{z}}(t) - \mathbf{z}_{\mathsf{e}}(t)) + (\mathbf{G} \otimes \mathbf{I}_{\mathsf{N}})\mathbf{F}(\hat{\mathbf{z}}_0(t);\theta) \\ &= \dot{\mathbf{z}}_{\mathsf{e}}(t) + (\mathbf{G}\boldsymbol{\Gamma}\mathbf{G}^{-1} \otimes \mathbf{I}_{\mathsf{N}})(\hat{\mathbf{z}}(t) - \mathbf{z}_{\mathsf{e}}(t)) - (\mathbf{G} \otimes \mathbf{I}_{\mathsf{N}})\mathbf{F}(\hat{\mathbf{z}}_0(t);\theta) \end{aligned} \tag{5.3}$$

since $\boldsymbol{\chi}(t) = -(\mathbf{G}^{-1} \otimes \mathbf{I}_{\mathsf{N}})(\hat{\mathbf{z}}(t) - \mathbf{z}_{\mathsf{e}}(t))$, and $(\mathbf{G} \otimes \mathbf{I}_{\mathsf{N}})(\boldsymbol{\Gamma} \otimes \mathbf{I}_{\mathsf{N}})(\mathbf{G}^{-1} \otimes \mathbf{I}_{\mathsf{N}}) = (\mathbf{G}\boldsymbol{\Gamma}\mathbf{G}^{-1} \otimes \mathbf{I}_{\mathsf{N}})$.

We will show that $\hat{\mathbf{z}}(t)$ converges towards the average consensus vector

$$\begin{bmatrix} \bar{z}(t)\mathbb{1}_{\mathsf{N}} \\ \vdots \\ \bar{z}^{(m)}(t)\mathbb{1}_{\mathsf{N}} \end{bmatrix}$$

asymptotically achieving EDC. This analysis is performed by decomposing

$$\hat{\mathbf{z}}(t) = (\mathbf{I}_{m+1} \otimes \mathbb{1}_{\mathsf{N}})\boldsymbol{\zeta}(t) + \tilde{\mathbf{z}}(t) \in \mathbb{R}^{(m+1)n}$$

in the consensus component

$$\boldsymbol{\zeta}(t) = (\mathbf{I}_{m+1} \otimes \mathbb{1}_{\mathsf{N}}^\top/\mathsf{N})\hat{\mathbf{z}}(t) \in \mathbb{R}^{m+1}$$

and in the consensus error $\tilde{\mathbf{z}}(t) = (\mathbf{I}_{m+1} \otimes \mathbf{P})\hat{\mathbf{z}}(t) \in \mathbb{R}^{(m+1)n}$ with $\mathbf{P} = (\mathbf{I}_{\mathsf{N}} - (1/\mathsf{N})\mathbb{1}_{\mathsf{N}}\mathbb{1}_{\mathsf{N}}^\top)$. Therefore, convergence of $\hat{\mathbf{z}}(t)$ can be established by means of showing that $\boldsymbol{\zeta}(t)$ converges exponentially to $[\bar{z}(t), \ldots, \bar{z}^{(m)}(t)]^\top$ and $\tilde{\mathbf{z}}(t)$ converges in finite time to the origin as we do in the following sections. First, we provide some results regarding structural properties of the matrices $\boldsymbol{\Gamma}, \mathbf{G}$ and their relation to the signals $z_i(t)$ and $s_i(t)$ from Assumption 5.2. These notions will be useful in subsequent proofs.

Lemma 5.3. *Let the change of variables*

$$\mathbf{s}(t) = \begin{bmatrix} \mathbf{s}_0(t) \\ \vdots \\ \mathbf{s}_m(t) \end{bmatrix} = (\mathbf{G}^{-1} \otimes \mathbf{I}_{\mathsf{N}})\mathbf{z}_{\mathsf{e}}(t)$$

with $\mathbf{s}_\mu(t) \in \mathbb{R}^{\mathsf{N}}, \forall \mu \in \{0, \dots, m\}$. *Moreover, define*

$$\mathbf{s}_{m+1}(t) = \mathbf{z}^{(m+1)}(t) + \sum_{\mu=0}^{m} l_\mu \mathbf{z}^{(\mu)}(t) \tag{5.4}$$

where $l_0, \dots, l_m$ *are the coefficients of the polynomial*

$$(\lambda + \gamma_0) \cdots (\lambda + \gamma_m) = \lambda^{m+1} + \sum_{\mu=0}^{m} l_\mu \lambda^\mu.$$

Then, we conclude that

$$\dot{\mathbf{z}}_{\mathsf{e}}(t) = (\tilde{\mathbf{\Gamma}} \otimes \mathbf{I}_{\mathsf{N}}) \mathbf{z}_{\mathsf{e}}(t) + (\mathbf{B} \otimes \mathbf{I}_{\mathsf{N}}) \mathbf{s}_{m+1}(t) \tag{5.5}$$

and

$$\dot{\mathbf{s}}(t) = (\mathbf{\Gamma} \otimes \mathbf{I}_{\mathsf{N}}) \mathbf{s}(t) + (\mathbf{B} \otimes \mathbf{I}_{\mathsf{N}}) \mathbf{s}_{m+1}(t) \tag{5.6}$$

with $\mathbf{B} = [0, \dots, 0, 1]^\top \in \mathbb{R}^{(m+1)\times 1}$ *and*

$$\tilde{\mathbf{\Gamma}} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ -l_0 & -l_1 & \cdots & \cdots & -l_m \end{bmatrix}. \tag{5.7}$$

Proof:

First, let $\mathbf{z}_\mu(t) := \mathbf{z}^{(\mu)}(t)$ and note that

$$\dot{\mathbf{z}}_\mu(t) = \mathbf{z}_{\mu+1}(t)$$

for $0 \leq \mu < m$ and

$$\dot{\mathbf{z}}_m(t) = \mathbf{s}_{m+1}(t) - \sum_{\mu=0}^{m} l_\mu \mathbf{z}_\mu(t)$$

from the definition in (5.4). Writing this in complete vector form leads to (5.5) directly. Now, rewrite (5.4) as

$$\mathbf{s}_{m+1}(t) = \left(\frac{d^{m+1}}{dt^{m+1}} + \sum_{\mu=0}^{m} l_\mu \frac{d^\mu}{dt^\mu} \right) \mathbf{z}(t) = \left(\frac{\mathrm{d}}{\mathrm{d}t} + \gamma_m \right) \cdots \left(\frac{\mathrm{d}}{\mathrm{d}t} + \gamma_0 \right) \mathbf{z}(t) \tag{5.8}$$

by the relation between the coefficients $l_0, \dots, l_m$ and $\gamma_0, \dots, \gamma_m$. Thus, define $\mathbf{v}_0(t) = \mathbf{z}_0(t)$ and recursively

$$\mathbf{v}_{\mu+1}(t) = \left(\frac{\mathrm{d}}{\mathrm{d}t} + \gamma_\mu \right) \mathbf{v}_\mu(t)$$

for $0 \leq \mu < m$ from which

$$\mathbf{s}_{m+1}(t) = \left(\frac{\mathrm{d}}{\mathrm{d}t} + \gamma_m \right) \mathbf{v}_m(t)$$

is obtained using (5.8). Equivalently, we have

$$\dot{\mathbf{v}}_\mu(t) = \mathbf{v}_{\mu+1}(t) - \gamma_\mu \mathbf{v}_\mu(t),\ 0 \le \mu < m$$

and $\dot{\mathbf{v}}_m(t) = \mathbf{s}_{m+1}(t) - \gamma_m \mathbf{v}_m(t)$. Written in vector form, $\mathbf{v}_\mathsf{e}(t) := [\mathbf{v}_0(t)^\top, \ldots, \mathbf{v}_m(t)^\top]^\top$ satisfies

$$\dot{\mathbf{v}}_\mathsf{e}(t) = (\mathbf{\Gamma} \otimes \mathbf{I}_\mathsf{N})\mathbf{v}_\mathsf{e}(t) + (\mathbf{B} \otimes \mathbf{I}_\mathsf{N})\mathbf{s}_{m+1}(t).$$

Now, we obtain the matrix which maps $\mathbf{v}_\mathsf{e}(t)$ to $\mathbf{z}_\mathsf{e}(t)$ by noting that with

$$\mathbf{v}_0(t) \equiv \mathbf{z}_0(t) = (\mathbf{C} \otimes \mathbf{I}_\mathsf{N})\mathbf{v}_\mathsf{e}(t)$$

it follows that

$$\mathbf{z}_1(t) = \dot{\mathbf{z}}_0(t) = (\mathbf{C\Gamma} \otimes \mathbf{I}_\mathsf{N})\mathbf{v}_\mathsf{e}(t) + (\mathbf{CB} \otimes \mathbf{I}_\mathsf{N})\mathbf{s}_{m+1}(t) = (\mathbf{C\Gamma} \otimes \mathbf{I}_\mathsf{N})\mathbf{v}_\mathsf{e}(t)$$

since $\mathbf{CB} = 0$ and continuing this procedure to obtain $\mathbf{z}_\mu(t) = \dot{\mathbf{z}}_{\mu-1}(t) = (\mathbf{C\Gamma}^\mu \otimes \mathbf{I}_\mathsf{N})\mathbf{v}_\mathsf{e}(t)$ since $\mathbf{C\Gamma}^{\mu-1}\mathbf{B} = 0$ for $0 \le \mu < m$. This can be written as $\mathbf{z}_\mathsf{e}(t) = (\mathbf{G} \otimes \mathbf{I}_\mathsf{N})\mathbf{v}_\mathsf{e}(t)$ or equivalently $\mathbf{v}_\mathsf{e}(t) = (\mathbf{G}^{-1} \otimes \mathbf{I}_\mathsf{N})\mathbf{z}_\mathsf{e}(t) = \mathbf{s}(t)$. Therefore, $\mathbf{s}(t)$ satisfy (5.6) concluding the proof. □

Corollary 5.4. *Let $\tilde{\mathbf{\Gamma}}, \mathbf{B}$ be defined as in Lemma 5.3. Then, $\tilde{\mathbf{\Gamma}} = \mathbf{G\Gamma G}^{-1}$ and $\mathbf{GB} = \mathbf{B}$.*

PROOF: Let the change of variables from Lemma 5.3 as $\mathbf{z}_\mathsf{e}(t) = (\mathbf{G} \otimes \mathbf{I}_\mathsf{N})\mathbf{s}(t)$. Then,

$$\dot{\mathbf{z}}_\mathsf{e} = (\mathbf{G} \otimes \mathbf{I}_\mathsf{N})(\mathbf{\Gamma} \otimes \mathbf{I}_\mathsf{N})\mathbf{s} + (\mathbf{G} \otimes \mathbf{I}_\mathsf{N})(\mathbf{B} \otimes \mathbf{I}_\mathsf{N})\mathbf{s}_{m+1} = (\mathbf{G\Gamma G}^{-1} \otimes \mathbf{I}_\mathsf{N})\mathbf{z}_\mathsf{e} + (\mathbf{GB} \otimes \mathbf{I}_\mathsf{N})\mathbf{s}_{m+1}$$

Comparing with (5.5) completes the proof. □

5.4 CONVERGENCE OF THE CONSENSUS COMPONENTS OF REDCHO

In this section we show the behaviour of

$$\boldsymbol{\zeta}(t) = \left(\mathbf{I}_{m+1} \otimes \frac{\mathbb{1}_\mathsf{N}^\top}{\mathsf{N}}\right)\hat{\mathbf{z}}(t)$$

as given in the following result.

Lemma 5.5. *Let $\hat{\mathbf{z}}(t) \in \mathbb{R}^{n(m+1)}$. Then, with any $\gamma_0, \ldots, \gamma_m > 0$ and any initial conditions $\hat{\mathbf{z}}(t_0)$ for (5.3) it is satisfied that $\boldsymbol{\zeta}(t)$ converge asymptotically towards*

$$\bar{\mathbf{z}}(t) = \left(\mathbf{I}_{m+1} \otimes \frac{\mathbb{1}_\mathsf{N}^\top}{\mathsf{N}}\right)\mathbf{z}_\mathsf{e}(t) = \begin{bmatrix} \bar{z}(t)\mathbb{1}_\mathsf{N} \\ \vdots \\ \bar{z}^{(m)}(t)\mathbb{1}_\mathsf{N} \end{bmatrix}.$$

PROOF: First, recall that $\mathbf{G\Gamma G}^{-1} = \tilde{\mathbf{\Gamma}}$ from Corollary 5.4 and obtain the dynamics of $\boldsymbol{\zeta}(t)$ by multiplying (5.3) by $(\mathbf{I}_{m+1} \otimes \mathbb{1}_\mathsf{N}^\top/\mathsf{N})$ from the left:

$$\begin{aligned}\dot{\boldsymbol{\zeta}}(t) &= \dot{\bar{\mathbf{z}}}(t) + (\mathbf{I}_{m+1} \otimes \mathbb{1}_\mathsf{N}^\top/\mathsf{N})(\tilde{\mathbf{\Gamma}} \otimes \mathbf{I}_\mathsf{N})(\hat{\mathbf{z}}(t) - \mathbf{z}_\mathsf{e}(t)) + (\mathbf{I}_{m+1} \otimes \mathbb{1}_\mathsf{N}^\top/\mathsf{N})(\mathbf{G} \otimes \mathbf{I}_\mathsf{N})\mathbf{F}(\hat{\mathbf{z}}_0(t); \theta) \\ &= \dot{\bar{\mathbf{z}}}(t) + \tilde{\mathbf{\Gamma}}(\boldsymbol{\zeta}(t) - \bar{\mathbf{z}}(t)) + \mathbf{G}(\mathbf{I}_{m+1} \otimes \mathbb{1}_\mathsf{N}^\top/\mathsf{N})\mathbf{F}(\hat{\mathbf{z}}_0(t); \theta)\end{aligned}$$

since

$$(\mathbf{I}_{m+1} \otimes \mathbb{1}_{\mathsf{N}}^{\top}/\mathsf{N})(\tilde{\mathbf{\Gamma}} \otimes \mathbf{I}_{\mathsf{N}}) = (\tilde{\mathbf{\Gamma}} \otimes \mathbb{1}_{\mathsf{N}}^{\top}/\mathsf{N}) = (\tilde{\mathbf{\Gamma}} \otimes \mathbf{I}_1)(\mathbf{I}_{m+1} \otimes \mathbb{1}_{\mathsf{N}}^{\top}/\mathsf{N}) = \tilde{\mathbf{\Gamma}}(\mathbf{I}_{m+1} \otimes \mathbb{1}_{\mathsf{N}}^{\top}/\mathsf{N}).$$

Moreover,

$$\mathbf{G}(\mathbf{I}_{m+1} \otimes \mathbb{1}_{\mathsf{N}}^{\top}/\mathsf{N})\mathbf{F}(\hat{\mathbf{z}}_0(t);\theta) = 0$$

since $\mathbb{1}_{\mathsf{N}}^{\top}\mathbf{D} = 0$ [158, Page 280]. Hence, we obtain

$$\dot{\boldsymbol{\zeta}}(t) = \dot{\bar{\mathbf{z}}}(t) + \tilde{\mathbf{\Gamma}}(\boldsymbol{\zeta}(t) - \bar{\mathbf{z}}(t)).$$

Define the error $\mathbf{e}(t) := \boldsymbol{\zeta}(t) - \bar{\mathbf{z}}(t)$ to obtain $\dot{\mathbf{e}}(t) = \tilde{\mathbf{\Gamma}}\mathbf{e}(t)$. Finally, note that $\tilde{\mathbf{\Gamma}}$ is given in (5.7) and has characteristic polynomial

$$\prod_{\mu=0}^{m}(\lambda + \gamma_\mu) = 0.$$

Then, with $\gamma_0, \ldots, \gamma_m > 0$, $\tilde{\mathbf{\Gamma}}$ has negative eigenvalues and $\boldsymbol{\zeta}(t) - \bar{\mathbf{z}}(t)$ asymptotically converge to the origin. □

5.5 CONVERGENCE OF THE CONSENSUS ERROR

In this section we show the behaviour of $\tilde{\mathbf{z}}(t) = (\mathbf{I}_{m+1} \otimes \mathbf{P})\hat{\mathbf{z}}(t)$ where $\mathbf{P} = (\mathbf{I}_{\mathsf{N}} - (1/\mathsf{N})\mathbb{1}_{\mathsf{N}}\mathbb{1}_{\mathsf{N}}^{\top})$. First, we obtain how the dynamics of $\tilde{\mathbf{z}}(t)$ relate to EDCHO.

Lemma 5.6. *Let* $\tilde{\mathbf{z}}(t) = (\mathbf{I}_{m+1} \otimes \mathbf{P})\hat{\mathbf{z}}(t)$ *where* $\mathbf{P} = (\mathbf{I}_{\mathsf{N}} - (1/\mathsf{N})\mathbb{1}_{\mathsf{N}}\mathbb{1}_{\mathsf{N}}^{\top})$ *and* $\hat{\mathbf{z}}(t)$ *satisfy* (5.3). *Moreover, let*

$$\mathbf{s}(t) := \begin{bmatrix} \mathbf{s}_0(t) \\ \vdots \\ \mathbf{s}_m(t) \end{bmatrix} = (\mathbf{G}^{-1} \otimes \mathbf{I}_{\mathsf{N}})\mathbf{z}_{\mathsf{e}}(t)$$

with $\mathbf{s}_\mu(t) \in \mathbb{R}^{\mathsf{N}}, \forall \mu \in \{0, \ldots, m\}$ *and* $\mathbf{\Theta} = diag(1, \theta^{-1}, \ldots, \theta^{-m})$. *Then,*

$$\mathbf{y}(t) := \begin{bmatrix} \mathbf{y}_0(t) \\ \vdots \\ \mathbf{y}_m(t) \end{bmatrix} = (\mathbf{\Theta}\mathbf{G}^{-1} \otimes \mathbf{I}_{\mathsf{N}})\tilde{\mathbf{z}}(t)$$

with $\mathbf{y}_\mu(t) \in \mathbb{R}^{\mathsf{N}}$, *satisfy*

$$\begin{aligned} \dot{\mathbf{y}}_\mu(t) &= \theta\Big(\mathbf{y}_{\mu+1}(t) - k_\mu \mathbf{D}\left\lceil \mathbf{D}^{\top}\mathbf{y}_0(t)\right\rfloor^{\frac{m-\mu}{m+1}} - (\gamma_\mu/\theta)\mathbf{y}_\mu(t)\Big) \text{ for } 0 \le \mu \le m-1, \\ \dot{\mathbf{y}}_m(t) &= \theta\Big(\mathbf{P}\mathbf{s}_{m+1}(t)/\theta^{m+1} - k_m \mathbf{D}\left\lceil \mathbf{D}^{\top}\mathbf{y}_0(t)\right\rfloor^{0} - (\gamma_m/\theta)\mathbf{y}_m(t)\Big) \end{aligned} \tag{5.9}$$

with $\mathbf{s}_{m+1}(t)$ *defined in* (5.4).

Proof: First, obtain the dynamics of $\tilde{\mathbf{z}}(t)$ using (5.3) and $\tilde{\mathbf{\Gamma}} = \mathbf{G}\mathbf{\Gamma}\mathbf{G}^{-1}$ from Corollary 5.4:

$$\frac{\mathrm{d}\tilde{\mathbf{z}}(t)}{\mathrm{d}t} = (\mathbf{I}_{m+1} \otimes \mathbf{P})\dot{\mathbf{z}}_{\mathsf{e}}(t) + (\tilde{\mathbf{\Gamma}} \otimes \mathbf{I}_{\mathsf{N}})\tilde{\mathbf{z}}(t) - (\tilde{\mathbf{\Gamma}} \otimes \mathbf{P})\mathbf{z}_{\mathsf{e}}(t) - (\mathbf{G} \otimes \mathbf{P})\mathbf{F}(\hat{\mathbf{z}}_0(t);\theta)$$

However, note that $(\mathbf{I}_{m+1}\otimes\mathbf{P})\dot{\mathbf{z}}_{\mathsf{e}}(t) = (\tilde{\mathbf{\Gamma}}\otimes\mathbf{P})\mathbf{z}_{\mathsf{e}}(t)+(\mathbf{B}\otimes\mathbf{P})\mathbf{s}_{m+1}(t)$ from (5.5) in Lemma 5.3. Then,

$$\frac{\mathrm{d}\tilde{\mathbf{z}}(t)}{\mathrm{d}t} = (\tilde{\mathbf{\Gamma}}\otimes\mathbf{I}_{\mathsf{N}})\tilde{\mathbf{z}}(t) - (\mathbf{G}\otimes\mathbf{P})\mathbf{F}(\hat{\mathbf{z}}_0(t);\theta) + (\mathbf{B}\otimes\mathbf{P})\mathbf{s}_{m+1}(t)$$

Moreover,

$$(\mathbf{G}\otimes\mathbf{P})\mathbf{F}(\hat{\mathbf{z}}_0(t);\theta) = (\mathbf{G}\otimes\mathbf{I}_{\mathsf{N}})\mathbf{F}(\hat{\mathbf{z}}_0(t);\theta)$$

since $\mathbf{PD} = (\mathbf{I}_{\mathsf{N}} - (1/\mathsf{N})\mathbb{1}_{\mathsf{N}}\mathbb{1}_{\mathsf{N}}^{\top})\mathbf{D} = \mathbf{D}$. Furthermore, the dynamics of $\mathbf{y}(t)$ are

$$\begin{aligned}\dot{\mathbf{y}}(t) &= (\mathbf{\Theta}\mathbf{G}^{-1}\otimes\mathbf{I}_{\mathsf{N}})\frac{\mathrm{d}\tilde{\mathbf{z}}(t)}{\mathrm{d}t}\\ &= (\mathbf{\Theta}\mathbf{G}^{-1}\tilde{\mathbf{\Gamma}}\mathbf{G}\mathbf{\Theta}^{-1}\otimes\mathbf{I}_{\mathsf{N}})\mathbf{y}(t) - (\mathbf{\Theta}\otimes\mathbf{I}_{\mathsf{N}})\mathbf{F}(\hat{\mathbf{z}}_0(t);\theta) + (\mathbf{\Theta}\mathbf{G}^{-1}\mathbf{B}\otimes\mathbf{P})\mathbf{s}_{m+1}(t)\\ &= (\mathbf{\Theta}\mathbf{\Gamma}\mathbf{\Theta}^{-1}\otimes\mathbf{I}_{\mathsf{N}})\mathbf{y}(t) - (\mathbf{\Theta}\otimes\mathbf{I}_{\mathsf{N}})\mathbf{F}(\hat{\mathbf{z}}_0(t);\theta) + \theta^{-m}(\mathbf{B}\otimes\mathbf{I}_{\mathsf{N}})\mathbf{P}\mathbf{s}_{m+1}(t)\end{aligned}$$

where Corollary 5.4 was used and

$$(\mathbf{\Theta}\mathbf{B}\otimes\mathbf{P}) = \theta^{-m}(\mathbf{B}\otimes\mathbf{I}_{\mathsf{N}})(\mathbf{I}_1\otimes\mathbf{P}) = \theta^{-m}(\mathbf{B}\otimes\mathbf{I}_{\mathsf{N}})P.$$

In addition, note that

$$(\mathbf{\Theta}\otimes\mathbf{I}_n)\mathbf{F}(\hat{\mathbf{z}}_0(t);\theta) = \begin{bmatrix}\mathbf{I}_{\mathsf{N}} & 0 & \cdots & 0\\ 0 & \mathbf{I}_{\mathsf{N}}\theta^{-1} & \ddots & \vdots\\ \vdots & \ddots & \ddots & 0\\ 0 & \cdots & 0 & \mathbf{I}_{\mathsf{N}}\theta^{-m}\end{bmatrix}\begin{bmatrix} k_0\theta\mathbf{D}\left\lceil\mathbf{D}^{\top}\hat{\mathbf{z}}_0(t)\right\rfloor^{\frac{m}{m+1}}\\ \vdots\\ k_m\theta^{m+1}\mathbf{D}\left\lceil\mathbf{D}^{\top}\hat{\mathbf{z}}_0(t)\right\rfloor^{0}\end{bmatrix} = \theta\mathbf{F}(\hat{\mathbf{z}}_0(t);1)$$

To simplify the $\mathbf{\Theta}\mathbf{\Gamma}\mathbf{\Theta}^{-1}$ term, denote the nilpotent matrix $\mathbf{A}_0 := \mathbf{\Gamma} + \mathrm{diag}(\gamma)$ which does not depend on any of the parameters $\boldsymbol{\gamma} = [\gamma_0,\ldots,\gamma_m]^{\top}$ and for which it can be verified $\mathbf{\Theta}\mathbf{A}_0 = \theta\mathbf{A}_0\mathbf{\Theta}$. Hence,

$$\mathbf{\Theta}\mathbf{\Gamma}\mathbf{\Theta}^{-1} = (\mathbf{\Theta}\mathbf{A}_0 - \mathbf{\Theta}\mathrm{diag}(\gamma))\mathbf{\Theta}^{-1} = (\theta\mathbf{A}_0\mathbf{\Theta} - \mathrm{diag}(\gamma)\mathbf{\Theta})\mathbf{\Theta}^{-1} = \theta(\mathbf{A}_0 - \mathrm{diag}(\gamma/\theta)) = \theta\mathbf{\Gamma}_{\theta}$$

Where $\mathbf{\Gamma}_{\theta} := \mathbf{A}_0 - \mathrm{diag}(\gamma/\theta)$. Moreover, the first row of $\mathbf{G}\mathbf{\Theta}^{-1}$ is $\mathbf{C}$ which leads to $\mathbf{y}_0(t) = \hat{\mathbf{z}}_0(t)$. Hence, combining all these facts,

$$\begin{aligned}\dot{\mathbf{y}}(t) &= (\theta\mathbf{\Gamma}_{\theta}\otimes\mathbf{I}_{\mathsf{N}})\mathbf{y}(t) - \theta\mathbf{F}(\mathbf{y}_0(t);1) + \theta^{-m}(\mathbf{B}\otimes\mathbf{I}_{\mathsf{N}})\mathbf{P}\mathbf{s}_{m+1}(t)\\ &= \theta\left((\mathbf{\Gamma}_{\theta}\otimes\mathbf{I}_{\mathsf{N}})\mathbf{y}(t) - \mathbf{F}(\mathbf{y}_0(t);1) + \theta^{-(m+1)}(\mathbf{B}\otimes\mathbf{I}_{\mathsf{N}})\mathbf{P}\mathbf{s}_{m+1}(t)\right)\end{aligned}$$

Writing this equation in partially vectorized form we recover (5.9), which completes the proof. □

Now, if we show that $\mathbf{y}(t)$ converge to the origin in finite time, the same conclusion will apply to $\tilde{\mathbf{z}}(t)$. Note that for given $\gamma_0,\ldots,\gamma_m$,

$$\mathbf{P}\mathbf{s}_{m+1}(t)/\theta^{m+1} \in [-L,L]^{\mathsf{N}}/\theta^{m+1} \subseteq [-L,L]^{\mathsf{N}}$$

under Assumption 5.2 and $\theta \geq 1$. Therefore, comparing (5.9) with the EDCHO error system (4.3) in Chapter 4, it would be the case that $\mathbf{y}(t)$ reaches the origin if $\gamma_0 = \cdots =$

$\gamma_m = 0$. In the following we will use homogeneity to show that even with those terms, stability of REDCHO will still be valid locally. To do so, we will decompose the right hand side of (5.9) in two parts, one similar to the right hand side of the EDCHO error system and the other with the remaining linear terms. Let $\mathbf{h} : \mathbb{R}^{n(m+1)} \rightrightarrows \mathbb{R}^{n(m+1)}$ and $\mathbf{q} : \mathbb{R}^{n(m+1)} \to \mathbb{R}^{n(m+1)}$ be defined as

$$\mathbf{h}(\mathbf{y}) = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_m \\ [-L, L]^{\mathsf{N}} \end{bmatrix} - \mathbf{F}(\mathbf{y}_0; 1), \quad \mathbf{q}(\mathbf{y}) = \begin{bmatrix} -(\gamma_0/\theta)\mathbf{y}_0 \\ \vdots \\ -(\gamma_m/\theta)\mathbf{y}_m \end{bmatrix} \tag{5.10}$$

Then, (5.9) is equivalent to the differential inclusion

$$\dot{\mathbf{y}}(t) \in \theta(\mathbf{h}(\mathbf{y}(t)) + \mathbf{q}(\mathbf{y}(t))).$$

Moreover, let $\mathbf{r} = [r_0 \mathbb{1}^\top, \ldots, r_m \mathbb{1}^\top]$ with $r_\mu := m + 1 - \mu, \forall \mu \in \{0, \ldots, m\}$. Then, it can be verified that $\mathbf{h}(\bullet)$ and $\mathbf{q}(\bullet)$ are $\mathbf{r}$-homogeneous of degrees -1 and 0 respectively in the sense of in the sense of Definition F.2 in Appendix F:

Lemma 5.7. *Let* $\mathbf{r} = [r_0 \mathbb{1}^\top, \ldots, r_m \mathbb{1}^\top]$ *with* $r_\mu = m + 1 - \mu, \forall \mu \in \{0, \ldots, m\}$. *Then,* $\mathbf{h}$ *defined in* (5.10) *and the identity map* $\mathsf{Id}(\mathbf{y}) = \mathbf{y}$ *are* $\mathbf{r}$*-homogeneous in the sense of Definition F.2 in Appendix F of degrees* -1 *and* 0 *respectively.*

Proof: First, note that $r_0 \frac{m-\mu}{m+1} = r_{\mu+1}$ and $r_{\mu+1} = r_\mu - 1$. Moreover, note that the dilation

$$\boldsymbol{\Delta}_{\mathbf{r}}(\lambda)\mathbf{y} = \begin{bmatrix} \lambda^{r_0}\mathbf{y}_0 \\ \vdots \\ \lambda^{r_m}\mathbf{y}_m \end{bmatrix}.$$

Furthermore, by splitting

$$\mathbf{h}(\mathbf{y}; \rho) = \begin{bmatrix} \mathbf{h}_0(\mathbf{y}; \rho) \\ \vdots \\ \mathbf{h}_m(\mathbf{y}; \rho) \end{bmatrix}$$

then,

$$\begin{aligned} \mathbf{h}_\mu(\boldsymbol{\Delta}_{\mathbf{r}}(\lambda)\mathbf{y}; \rho) &= \lambda^{r_{\mu+1}} - \rho^{\mu+1} k_\mu \mathbf{D} \left\lceil \mathbf{D}^\top \lambda^{r_0} \mathbf{y}_0(t) \right\rfloor^{\frac{m-\mu}{m+1}} \\ &= \lambda^{r_{\mu+1}} \mathbf{y}_\mu - \lambda^{r_{\mu+1}} \rho^{\mu+1} k_\mu \mathbf{D} \left\lceil \mathbf{D}^\top \lambda^{r_0} \mathbf{y}_0(t) \right\rfloor^{\frac{m-\mu}{m+1}} \\ &= \lambda^{-1+r_\mu} \left(\mathbf{y}_\mu - \rho^{\mu+1} k_\mu \mathbf{D} \left\lceil \mathbf{D}^\top \lambda^{r_0} \mathbf{y}_0(t) \right\rfloor^{\frac{m-\mu}{m+1}} \right) \\ &= \lambda^{-1} \lambda^{r_\mu} \mathbf{h}_\mu(\mathbf{y}; \rho) \end{aligned}$$

for $\mu \in \{0, \ldots, m-1\}$. For $\mathbf{h}_m$, Note that $r_m = 1$ and

$$\mathbf{h}_m(\boldsymbol{\Delta}_{\mathbf{r}}(\lambda)\mathbf{y}; \rho) = \mathbf{h}_m(\mathbf{y}; \rho) = \lambda^{-1} \lambda^{r_m} \mathbf{h}_m(\mathbf{y}; \rho).$$

Hence,

$$\mathbf{h}(\boldsymbol{\Delta}_{\mathbf{r}}(\lambda)\mathbf{y}; \rho) = \lambda^{-1} \boldsymbol{\Delta}_{\mathbf{r}}(\lambda) \mathbf{h}(\mathbf{y}; \rho).$$

Then, $\mathbf{h}$ is $\mathbf{r}$-homogeneous in the sense of Definition F.2 of degree -1. For $\mathsf{Id}(\mathbf{y}) = [\mathsf{Id}_0^\top(\mathbf{y}), \dots, \mathsf{Id}_m^\top(\mathbf{y})]^\top$ with $\mathsf{Id}_\mu(\mathbf{y}) = \mathbf{y}_\mu$,

$$\mathsf{Id}_\mu(\mathbf{\Delta}_\mathbf{r}(\lambda)\mathbf{y}) = -\gamma_\mu \lambda^{r_\mu}\mathbf{y}_\mu = \lambda^{r_\mu}\mathsf{Id}_\mu(\mathbf{y})$$

Hence, $\mathsf{Id}(\mathbf{\Delta}_\mathbf{r}(\lambda)\mathbf{y}) = \lambda^0 \mathbf{\Delta}_\mathbf{r}(\lambda)\mathsf{Id}(\mathbf{y})$ and Id is $\mathbf{r}$-homogeneous in the sense of Definition F.2 of degree 0. □

Lemma 5.8. *Let $\mathcal{G}$ be a connected graph and the pair $\mathbf{h}(\bullet)$, $\mathbf{q}(\bullet)$ defined in (5.10). Moreover, let some fixed $\gamma_0, \dots, \gamma_m > 0$ so that there exists L for which Assumption 5.2 is complied and $k_0, \dots, k_m$ chosen as in Theorem 4.7 in Chapter 4 for such L. Then, there exists a neighborhood $\mathcal{R}_0 \subset \mathbb{R}^{n(m+1)}$ of the origin such that if $\mathbf{y}(t_0) \in \mathcal{R}_0$, then the solution of*

$$\dot{\mathbf{y}}(t) \in \theta(\mathbf{h}(\mathbf{y}(t)) + \mathbf{q}(\mathbf{y}(t)))$$

converge to the origin in finite-time. Moreover, $\mathcal{R}_0$ can be made arbitrarily big by increasing θ.

Proof: Consider $\gamma_0 = \cdots = \gamma_m = 0, \theta = 1$. Then, $\dot{\mathbf{y}} \in \mathbf{h}(\mathbf{y})$ is globally finite-time stable towards the origin by Theorem 4.7 from Chapter 4. Moreover, recall that $\mathbf{h}(\bullet)$ is $\mathbf{r}$-homogeneous of degree -1. Hence, by Proposition F.5 in Appendix F there exists scalar functions $V(\mathbf{y}), W(\mathbf{y})$ which are $\mathbf{r}$-homogeneous of degrees k and $k-1$ respectively and comply

$$\dot{V}(\mathbf{y}(t)) = L_\mathbf{h} V(\mathbf{y}(t)) \leq -W(\mathbf{y}(t)) \leq -\beta_m V(\mathbf{y}(t))^{\frac{k-1}{k}}$$

with

$$0 < \beta_m = \inf\{W(\mathbf{y}) : V(\mathbf{y}) = 1\}$$

using Proposition F.3. Now consider $\gamma_0, \dots, \gamma_m > 0$ with arbitrary $\theta \geq 1$ and the same Lyapunov function $V(\mathbf{y})$ as before. In this case

$$\dot{V}(\mathbf{y}(t)) = \theta\left(L_\mathbf{h} V(\mathbf{y}(t)) + L_\mathbf{q} V(\mathbf{y}(t))\right) \leq \theta\left(-\beta_m V(\mathbf{y}(t))^{\frac{k-1}{k}} + L_\mathbf{q} V(\mathbf{y}(t))\right)$$

Note that from Proposition F.4

$$L_\mathbf{q} V(\mathbf{y}) = \frac{\partial V}{\partial \mathbf{y}}\mathbf{q}(\mathbf{y})$$

is $\mathbf{r}$-homogeneous of degree k since $\mathbf{q}(\bullet)$ is $\mathbf{r}$-homogeneous of degree 0. Hence, $L_\mathbf{q} V(\mathbf{y}) \leq \beta_M V(\mathbf{y})$ by Proposition F.3 and

$$\beta_M = \sup\{L_\mathbf{q} V(\mathbf{y}) : V(\mathbf{y}) = 1\}.$$

Note that $L_\mathbf{q} V(\mathbf{y})$ may be positive for some $\mathbf{y}$. Therefore, β_M may be positive too. Moreover,

$$\dot{V}(\mathbf{y}(t)) \leq \theta\left[-\beta_m V(\mathbf{y}(t))^{\frac{k-1}{k}} + \beta_M V(\mathbf{y}(t))\right] \leq -\theta\left[\beta_m - \beta_M V(\mathbf{y}(t))^{\frac{k}{k-1}}\right] V(\mathbf{y}(t))^{\frac{k-1}{k}}$$

Denote with $\mathcal{R}_0 \subseteq \mathbb{R}^{n(m+1)}$ any region in which

$$-\left[\beta_m - \beta_M V(\mathbf{y})^{\frac{k}{k-1}}\right] \leq -c \tag{5.11}$$

is complied for some $c > 0$ so that

$$\dot{V}(\mathbf{y}(t)) \leq -\theta c V(\mathbf{y}(t))^{\frac{k-1}{k}}$$

for any $\mathbf{y}(t) \in \mathcal{R}_0$. If $\beta_M \leq 0$, then (5.11) is complied for $c = \beta_m$ regardless of $\mathbf{y}$ so that we can set $\mathcal{R}_0 = \mathbb{R}^{n(m+1)}$. On the other hand, if $\beta_M > 0$ then (5.11) is complied when

$$V(\mathbf{y})^{\frac{k}{k-1}} \leq (\beta_m - c)/\beta_M$$

which is possible only for $c \in (0, \beta_m)$. Then, choose $\mathcal{R}_0$ with $c = \beta_m/2$ so that (5.11) is complied whenever $V(\mathbf{y}) \leq \beta_m/(2\beta_M)$. We can write explicitly

$$\mathcal{R}_0 = \{\mathbf{y} \in \mathbb{R}^{n(m+1)} : V(\mathbf{y}) \leq \beta_m/(2\beta_M)\}$$

so that

$$\dot{V}(\mathbf{y}(t)) \leq -(\theta\beta_m/2)V(\mathbf{y}(t))^{\frac{k-1}{k}}$$

for any $\mathbf{y}(t) \in \mathcal{R}_0$ regardless of β_M. Note that due to Proposition F.5, $V(\mathbf{y})$ is continuously differentiable and we can write

$$\beta_M = \sup\left\{-\sum_{\mu=0}^{m}(\gamma_\mu/\theta)\frac{\partial V}{\partial \mathbf{y}_\mu}\mathbf{y}_\mu : V(\mathbf{y}) = 1\right\} = \tilde{\beta}_M/\theta$$

where

$$\tilde{\beta}_M := \sup\left\{-\sum_{\mu=0}^{m}\gamma_\mu\frac{\partial V}{\partial \mathbf{y}_\mu}\mathbf{y}_\mu : V(\mathbf{y}) = 1\right\}$$

is a constant for fixed $\gamma_0, \ldots, \gamma_m$. Thus,

$$\beta_m/(2\beta_M) = \theta(\beta_m/(2\tilde{\beta}_M))$$

can be made arbitrarily big by increasing θ, so that $\mathcal{R}_0$ can be made arbitrarily big as well. Finally, note that since $(k-1)/k \in (0,1)$, then $V(\mathbf{y})$ will reach the origin in finite time [159, Corolary 4.25] and so will $\mathbf{y}(t)$ whenever $\mathbf{y}(t_0) \in \mathcal{R}_0$, completing the proof. □

The previous result shows that trajectories of

$$\dot{\mathbf{y}}(t) \in \theta(\mathbf{h}(\mathbf{y}(t)) + \mathbf{q}(\mathbf{y}(t)))$$

reach the origin if $\mathbf{y}(t_0) \in \mathcal{R}_0$ which motivates to study if diverging trajectories can be obtained for some $\mathbf{y}(t_0) \notin \mathcal{R}_0$. In the following, we show that this is not possible and only a terminal bounded error is allowed.

Lemma 5.9. *Let the conditions of Lemma 5.8 be satisfied. Thus, for any initial conditions $\mathbf{y}(t_0) \in \mathbb{R}^{n(m+1)}$, there exists $T > 0$ and a bounded neighborhood of the origin $\mathcal{R}_\infty$ such that solution of*

$$\dot{\mathbf{y}}(t) \in \theta(\mathbf{h}(\mathbf{y}(t)) + \mathbf{q}(\mathbf{y}(t)))$$

comply $\mathbf{y}(t) \in \mathcal{R}_\infty$ for $t \geq T + t_0$. Moreover, such neighborhood can be made arbitrarily big by increasing θ.

PROOF: We proceed very similarly to the proof of Lemma 5.8. Consider only $\dot{\mathbf{y}}(t) = \theta\mathbf{q}(\mathbf{y}(t))$ using (5.10) and a Lyapunov function

$$V(\mathbf{y}) = \sum_{\mu=0}^{m} (\mathbf{y}_\mu^\top \mathbf{y}_\mu)^{\frac{m+1}{2r_\mu}}$$

with

$$r_\mu := m + 1 - \mu, \forall \mu \in \{0, \dots, m\}$$

obtaining

$$\dot{V}(\mathbf{y}(t)) = \sum_{\mu=0}^{m} \frac{m+1}{r_\mu} (\mathbf{y}_\mu(t)^\top \mathbf{y}_\mu(t))^{\frac{m+1}{2r_\mu}-1} \mathbf{y}_\mu(t)^\top \left(-\gamma_\mu \mathbf{y}_\mu(t)\right) \leq -\gamma_{\min} V(\mathbf{y}(t))$$

since $\gamma_{\min} := \min\{\gamma_0, \dots, \gamma_m\} \leq \gamma_\mu$ and $1 \leq \frac{m+1}{r_\mu}$, $\forall \mu \in \{0, \dots, m\}$. Moreover, note that

$$V(\mathbf{\Delta}_\mathbf{r}(\lambda)\mathbf{y}) = \sum_{\mu=0}^{m} (\lambda^{2r_\mu} \mathbf{y}_\mu^\top \mathbf{y}_\mu)^{\frac{m+1}{2r_\mu}} = \lambda^{m+1} V(\mathbf{y})$$

and thus $V(\mathbf{y})$ is $\mathbf{r}$-homogenenous of degree $m+1$ under the dilation

$$\mathbf{\Delta}_\mathbf{r}(\lambda)\mathbf{y} = \begin{bmatrix} \lambda^{r_0}\mathbf{y}_0 \\ \vdots \\ \lambda^{r_m}\mathbf{y}_m \end{bmatrix}.$$

Let the same Lyapunov function for

$$\dot{\mathbf{y}}(t) \in \theta(\mathbf{h}(\mathbf{y}(t)) + \mathbf{q}(\mathbf{y}(t)))$$

so that

$$\dot{V}(\mathbf{y}(t)) = \theta \left(L_\mathbf{q} V(\mathbf{y}(t)) + L_\mathbf{h} V(\mathbf{y}(t))\right) \leq -\gamma_{\min} V(\mathbf{y}(t)) + \theta L_\mathbf{h} V(\mathbf{y}(t)).$$

Note that from Proposition F.4 $L_\mathbf{h} V(\mathbf{y})$ is $\mathbf{r}$-homogeneous of degree m since $\mathbf{h}(\bullet)$ is $\mathbf{r}$-homogeneous of degree -1. Hence,

$$L_\mathbf{h} V(\mathbf{y}) \leq \beta'_M V(\mathbf{y})^{\frac{m}{m+1}}$$

by Proposition F.3 and

$$\beta'_M = \sup\{L_\mathbf{h} V(\mathbf{y}) : V(\mathbf{y}) = 1\}.$$

Thus,

$$\dot{V}(\mathbf{y}(t)) \leq -\left[\gamma_{\min} - \theta\beta'_M V(\mathbf{y}(t))^{-\frac{1}{m+1}}\right] V(\mathbf{y}(t)).$$

Denote with $\mathcal{R}^c_\infty \subset \mathbb{R}^{n(m+1)}$ any region in which

$$-\theta\left[\frac{\gamma_{\min}}{\theta} - \beta'_M V^{-\frac{1}{m+1}}\right] \leq -\theta c' \tag{5.12}$$

for some $c' > 0$ so that

$$\dot{V}(\mathbf{y}(t)) \leq -\theta c' V(\mathbf{y}(t))$$

for any $\mathbf{y}(t) \in \mathcal{R}_\infty^c$. If $\beta'_M \leq 0$, then we can set $\mathcal{R}_\infty^c = \mathbb{R}^{n(m+1)}$ with $c' = \gamma_{\min}/\theta$. On the other hand if $\beta'_M > 0$, (5.12) is equivalent to

$$V(\mathbf{y})^{1/(m+1)} \geq \beta'_M/(\gamma_{\min}/\theta - c').$$

Then, choose $c' = \gamma_{\min}/(2\theta)$ so that we can write

$$\mathcal{R}_\infty^c = \{\mathbf{y} \in \mathbb{R}^{n(m+1)} : V(\mathbf{y})^{1/k} \geq 2\theta\beta'_M/\gamma_{\min}\}$$

so that

$$\dot{V}(\mathbf{y}(t)) \leq -(\gamma_{\min}/2)V(\mathbf{y}(t))$$

for any $\mathbf{y}(t) \in \mathcal{R}_\infty^c$. Therefore, for any initial condition $\mathbf{y}(t_0) \in \mathbb{R}^{n(m+1)}$ the trajectory of $V(\mathbf{y}(t))$ will converge to

$$\mathcal{R}_\infty := \mathbb{R}^{n(m+1)} \setminus \mathcal{R}_\infty^c$$

after a finite time $T + t_0$ and comply $\mathbf{y}(t) \in \mathcal{R}_\infty$ for all $t \geq T + t_0$. Finally, note that $\mathcal{R}_\infty$ can be made arbitrarily large by increasing θ.

□

5.6 CONVERGENCE OF REDCHO

In this section we formally state the main result of this chapter.

Theorem 5.10. *Let $\mathcal{G}$ be a connected graph and the pair $\mathbf{h}(\bullet)$, $\mathbf{q}(\bullet)$ defined in* (5.10)*. Moreover, let some fixed $\gamma_0, \ldots, \gamma_m > 0$ so that there exists L for which Assumption 5.2 is complied and $k_0, \ldots, k_m$ chosen as in Theorem 4.7 in Chapter 4 for such L. Then, there exists neighborhoods $\mathcal{R}, \mathcal{R}' \subset \mathbb{R}^{n(m+1)}$ around consensus such that if the initial conditions comply $[\hat{z}_{1,0}(t_0), \ldots, \hat{z}_{n,m}(t_0)]^\top \in \mathcal{R}$, the REDCHO algorithm in* (5.2) *achieves robust EDC. On the other hand, if $[\hat{z}_{1,0}(t_0), \ldots, \hat{z}_{n,m}(t_0)]^\top \notin \mathcal{R}$,* (5.2) *will achieve at most a uniformly bounded terminal error $[\hat{z}_{1,0}(t), \ldots, \hat{z}_{n,m}(t)]^\top \in \mathcal{R}', \forall t \geq T$ around dynamic consensus after some finite time $T > 0$. Moreover, the neighborhoods $\mathcal{R}, \mathcal{R}'$ can be made arbitrarily big by increasing $\theta \geq 1$.*

Proof: First, decompose

$$\hat{\mathbf{z}}(t) = (I \otimes \mathbb{1}_\mathsf{N})(\mathbf{I}_{m+1} \otimes \mathbb{1}_\mathsf{N}^\top/\mathsf{N})\hat{\mathbf{z}}(t) + (\mathbf{I}_{m+1} \otimes \mathbf{P})\hat{\mathbf{z}}(t) = (\mathbf{I}_{m+1} \otimes \mathbb{1})\boldsymbol{\zeta}(t) + \tilde{\mathbf{z}}(t).$$

Now, note that Lemma 5.8 implies the existence of a neighborhood $\mathcal{R}_0$ such that if $\mathbf{y}(t_0) \in \mathcal{R}_0$ with

$$\mathbf{y}(t) := (\boldsymbol{\Theta}\mathbf{G}^{-1} \otimes \mathbf{I}_\mathsf{N})\tilde{\mathbf{z}}(t)$$

then convergence of $\mathbf{y}(t)$ is achieved towards the origin. Hence, consider the biggest ball of radius $R(\theta)$,

$$\mathcal{B}_0(\theta) = \{\mathbf{y} \in \mathbb{R}^{n(m+1)} : \mathbf{y}^\top\mathbf{y} \leq R(\theta)^2\}$$

such that $\mathcal{B}_0(\theta) \subseteq \mathcal{R}_0$ and $R(\theta)$ is an increasing function of θ due to the last part of Lemma 5.8. Now,

$$R(\theta)^2 \geq \mathbf{y}^\top\mathbf{y} = \tilde{\mathbf{z}}^\top(\boldsymbol{\Theta}(\mathbf{G}^{-1})^\top\mathbf{G}^{-1}\boldsymbol{\Theta} \otimes \mathbf{I}_\mathsf{N})\tilde{\mathbf{z}} \geq \theta^{-2m}\tilde{\mathbf{z}}^\top((\mathbf{G}^{-1})^\top\mathbf{G}^{-1} \otimes \mathbf{I}_\mathsf{N})\tilde{\mathbf{z}}$$

This implies that convergence of $\tilde{\mathbf{z}}(t)$ towards the origin happens for any

$$\tilde{\mathbf{z}}(t_0) \in \{\tilde{\mathbf{z}} \in \mathbb{R}^{n(m+1)} : \tilde{\mathbf{z}}^\top ((\mathbf{G}^{-1})^\top \mathbf{G}^{-1} \otimes \mathbf{I}_{\mathsf{N}})\tilde{\mathbf{z}} \leq (\theta^m R(\theta))^2\}$$

where the previous region can be made arbitrarily big by increasing θ. This implies the existence of $\mathcal{R} \in \mathbb{R}^{n(m+1)}$ as required by the theorem. An identical argument can be made for $\mathcal{R}' \in \mathbb{R}^{n(m+1)}$ but using Lemma 5.9 instead to conclude uniformly bounded trajectories for $\tilde{\mathbf{z}}(t)$ implying uniformly bounded steady state error around dynamic consensus. Furthermore, Lemma 5.5 imply that $\boldsymbol{\zeta}(t)$ converge asymptotically towards $\bar{\mathbf{z}}(t) = (\mathbf{I}_{m+1} \otimes \mathbb{1}_{\mathsf{N}}^\top/\mathsf{N})\mathbf{z}_{\mathsf{e}}(t) = \left[\bar{z}(t)\mathbb{1}_{\mathsf{N}}^\top, \ldots, \bar{z}^{(m)}(t)\mathbb{1}_{\mathsf{N}}^\top\right]^\top$. Hence, when $\hat{\mathbf{z}}(t_0) \in \mathcal{R}$, then $\hat{\mathbf{z}}(t)$ converge asymptotically towards $(\mathbf{I}_{m+1} \otimes \mathbb{1}_{\mathsf{N}})\bar{\mathbf{z}}(t)$. Equivalently, $\hat{\mathbf{z}}_\mu(t) \to \bar{z}^{(\mu)}(t)\mathbb{1}_{\mathsf{N}}$. Since no initialization condition is required, then (5.2) achieves robust EDC. □

Remark 5.11. *Note that since the $\gamma_0, \ldots, \gamma_m$ are fixed, the class of signals for which Assumption 5.2 is complied can be checked before-hand, so that the method remains fully distributed. On the other hand, showing the same stability properties in the case when all agents have different parameters $\gamma_0^i, \ldots, \gamma_m^i > 0, i \in \{1, \ldots, n\}$ require more complicated computations, but is straightforward using similar arguments as in this chapter. Thus, only the case with a single set of parameters for all agents is provided here for simplicity.*

5.7 SIMULATION EXAMPLES

In the following we show some simulation scenarios designed to show the properties of the REDCHO protocol. The simulations were implemented using explicit Euler method with time step $\Delta t = 10^{-6}$ over (5.2).

In order to show the convergence properties as described in the analysis from the previous sections, we simulated (5.2) for the network topology $\mathcal{G}$ shown in Figure 5.2. Moreover,the number of agents is $n = 8$ and the signals $z_i(t) = a_i \cos(\omega_i t)$ with amplitudes

$$a_i = 0.95, 0.34, 0.58, 0.22, 0.75, 0.25, 0.50, 0.69$$

and frequencies

$$\omega_i = 0.70, 0.75, 0.27, 0.67, 0.65, 0.16, 0.11, 0.49.$$

Thus, we choose $m = 2$, $\gamma_0 = \gamma_1 = \gamma_2 = 3$, $k_0 = 6, k_1 = 11, k_2 = 6, \theta = 1.5$. Furthermore, initial conditions for (5.2) were generated from a normal distribution with mean 1 and variance $r = 1$. Figure 5.3 shows the convergence of $\hat{\mathbf{z}}_0(t), \hat{\mathbf{z}}_1(t), \hat{\mathbf{z}}_2(t)$ towards the signals $\bar{z}(t)\mathbb{1}, \dot{\bar{z}}(t)\mathbb{1}, \ddot{\bar{z}}(t)\mathbb{1}$ in the first column. Moreover, by letting $\boldsymbol{\zeta}(t) = [\boldsymbol{\zeta}_0(t), \ldots, \boldsymbol{\zeta}_m(t)]^\top$, we show the average consensus error $\mathbf{e}_\mu(t) := \boldsymbol{\zeta}_\mu(t) - \bar{z}^{(\mu)}(t)\mathbb{1}, \mu = 0, 1, 2$ in the second column of Figure 5.3, which converges asymptotically to the origin as expected from Lemma 5.5. The third column of Figure 5.3 shows how the consensus errors $\tilde{\mathbf{z}}_0(t), \tilde{\mathbf{z}}_1(t), \tilde{\mathbf{z}}_2(t)$ converge to the origin in finite time as expected from Lemma 5.8. This same experiment was repeated for different values of r for which the norm of the average consensus errors $\|\hat{\mathbf{z}}_\mu(t) - \bar{z}^{(\mu)}(t)\mathbb{1}\|, \mu = 0, 1, 2$ is shown in Figure 5.4. This experiment shows that even for big initial conditions, the algorithm manages to converge to EDC. This is consistent with Lemma 5.9 which implies that no diverging error trajectories are possible.

In order to show the robustness properties of the REDCHO protocol when the topology suffers from sudden changes, we simulated (5.2) for the network topology shown in Figure

5.5 which changes from $\mathcal{G}_{t<5}$ to $\mathcal{G}_{t\geq 5}$ at $t = 5$. Consider the same configuration and signals as in the previous example. Figure 5.6 shows how the outputs of the REDCHO algorithm converge to EDC approximately at $t = 0.5$. At $t = 5$ the topology changes, but the REDCHO protocol manages to make all agents, the first four and the new ones, converge to EDC again even when the states did not comply neither $\sum_{i=1}^{4} \chi_{i,\mu}(0) = 0$ nor $\sum_{i=1}^{8} \chi_{i,\mu}(5) = 0$ as required in EDCHO. For comparison, consider EDCHO obtained by setting $\gamma_0 = \gamma_1 = \gamma_2 = 0$ in REDCHO. Moreover, initial conditions are changed so that $\sum_{i=1}^{4} \chi_{i,\mu}(0) = 0$. Figure 5.7 shows the trajectories for the protocol in this case, where EDC is achieved before $t = 5$. However, when the new agents merge to the network, the agents output converge to consensus towards a signal that diverges.

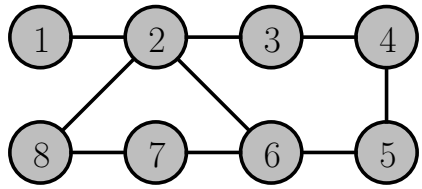

Figure 5.2: The network topology $\mathcal{G}$ considered in the first example of Section 5.7.

In the previous examples we showed the effectiveness of REDCHO against its non-robust version (EDCHO). In the following we compare against other state of the art dynamic consensus methods, with particular focus on terminal precision for the EDC goal.

In this example, we compare REDCHO with the Boundary-layer (B-layer) approach from [154] and the High-Order Linear protocol (HOL) from [84]. Both previous methods are able to achieve consensus towards the average signal and its derivatives, but are not robust and require to share a whole vector between agents. In addition, we compare with the First-Order Linear (FOL) protocol in [67] and the First-Order Sliding Mode (FOSM) protocol in [85]. Both protocols are robust, but cannot obtain the derivatives of the average signal by construction. Thus, a robust exact differentiator [96] is applied locally at each agent to obtain derivatives of the average signal.

In this setting, consider $\mathcal{G}$ constructed as a ring topology of $n = 20$ agents. Similarly as before, consider signals $z_i(t) = a_i \cos(\omega_i t)$ where the a_i, ω_i are not shown for brevity. Note that all approaches can handle these type of signals, with at least a bounded terminal consensus error regardless of the order of the algorithm. An order of $m = 2$ was used for REDCHO, B-layer and HOL, and an exact differentiator of order m is applied for FOL and FOSM. Hence, all algorithms are able to obtain up to the second derivative of the average signal. All algorithms were implemented with parameters of similar magnitude, chosen such to roughly match same settling time for the sake of fairness. Moreover, we show the resulting consensus errors for all algorithms with $h = 10^{-6}$ as shown in Figure 5.8 and $\Delta t = 10^{-3}$ in Figure 5.9 to show how they degrade as the discretization becomes coarser. In addition, we simulated that agent 1 fails at $t = 25$ and resets its state, allowing us to evaluate robustness of the algorithms.

As it can be observed, HOL and FOL methods have similar low precision in all cases before $t = 25$. The reason is that neither of these methods are able to achieve exact convergence for sinusoidal signals. However, their performance does not degrade significantly

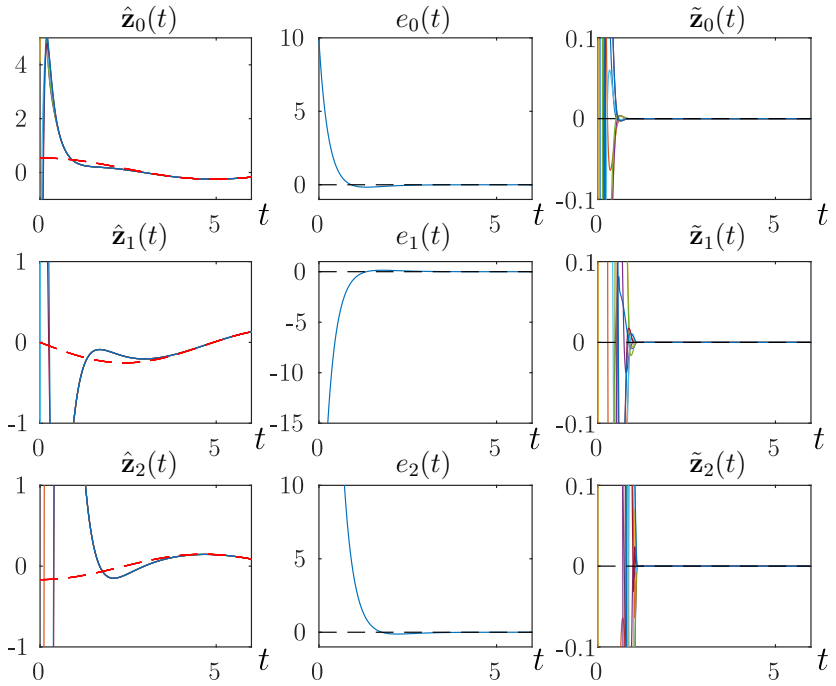

Figure 5.3: Convergence of the REDCHO algorithm for the scenario considered in the first example of Section 5.7. The first column shows convergence of $\hat{\mathbf{z}}_0(t), \hat{\mathbf{z}}_1(t), \hat{\mathbf{z}}_2(t)$ towards the signals $\bar{z}(t)\mathbb{1}, \dot{\bar{z}}(t)\mathbb{1}, \ddot{\bar{z}}(t)\mathbb{1}$. The second column shows the average consensus error $e_\mu(t) := \mathbb{1}^\top \mathbf{e}_\mu(t)/\mathsf{N}$ with $\mathbf{e}_\mu(t) := \boldsymbol{\zeta}_\mu(t) - \bar{z}^{(\mu)}\mathbb{1}(t), \mu = 0, 1, 2$. The third column shows the consensus errors $\tilde{\mathbf{z}}_0(t), \tilde{\mathbf{z}}_1(t), \tilde{\mathbf{z}}_2(t)$.

when the time step is increased. On the other hand, it can be noted that the FOSM approach have better performance than the linear approaches when $\Delta t = 10^{-6}$ due to its theoretically exact convergence. However, it degrades significantly when h is increased as shown in Figure 5.9. The reason is that this method suffers from the chattering effect which is amplified for the higher order derivatives due to the exact differentiator. Note that the B-layer and REDCHO approaches have similar performance before $t = 25$ with both sampling step sizes and outperform the other methods with at least one order of magnitude of precision improvement when $\Delta t = 10^{-3}$ as shown in Figure 5.9. However, after $t = 25$ both B-layer and HOL converge to consensus only up to a constant error due to their lack of robustness as shown by the $\|\hat{\mathbf{z}}_0(t) - \bar{z}(t)\mathbb{1}\|$ curves in both Figures 5.8 and 5.9. Although other methods manage to recover from the failure of agent 1, REDCHO is the one with the best performance in all cases.

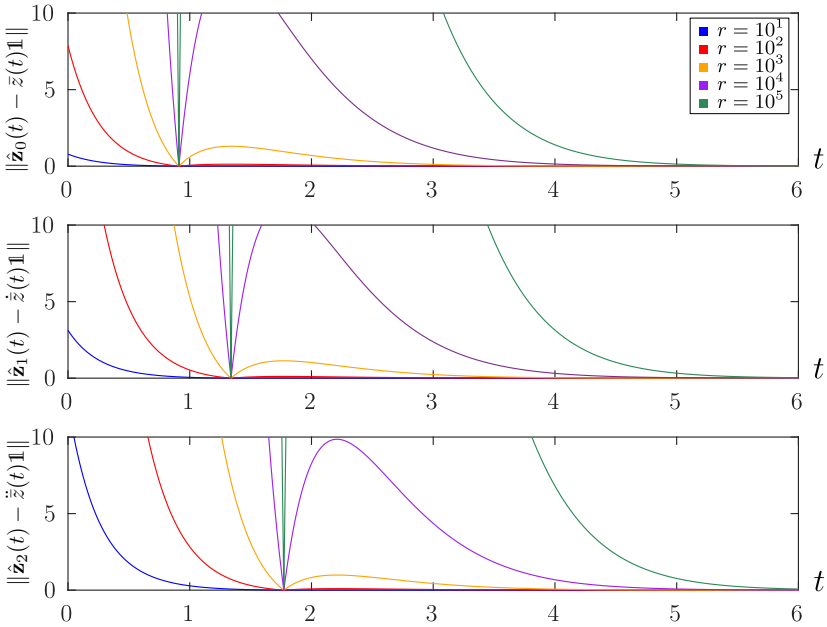

Figure 5.4: Convergence of the REDCHO algorithm for different magnitudes of initial conditions considered in the first example of Section 5.7.

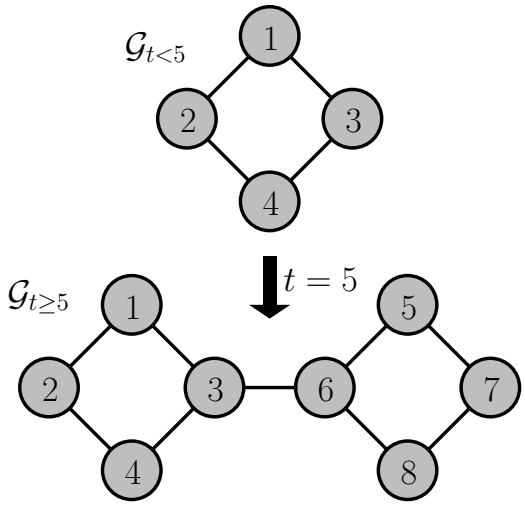

Figure 5.5: The network $\mathcal{G}$ considered in the network merging example in Section 5.7.

5.8 DISCUSSION

In this chapter we proposed the REDCHO protocol. This new protocol achieves exact consensus towards the average of time varying signals and its derivatives distributed

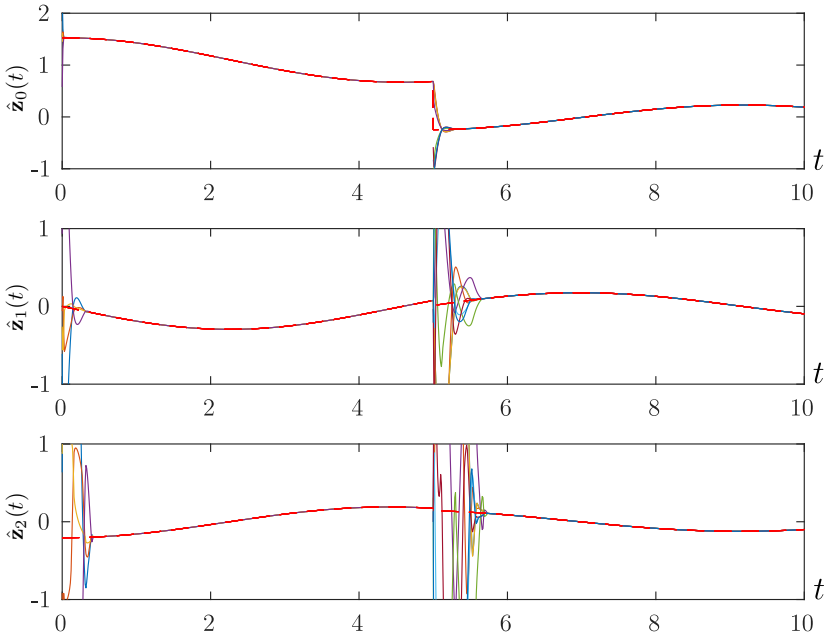

Figure 5.6: Trajectories for the REDCHO protocol in the scenario of Figure 5.5.

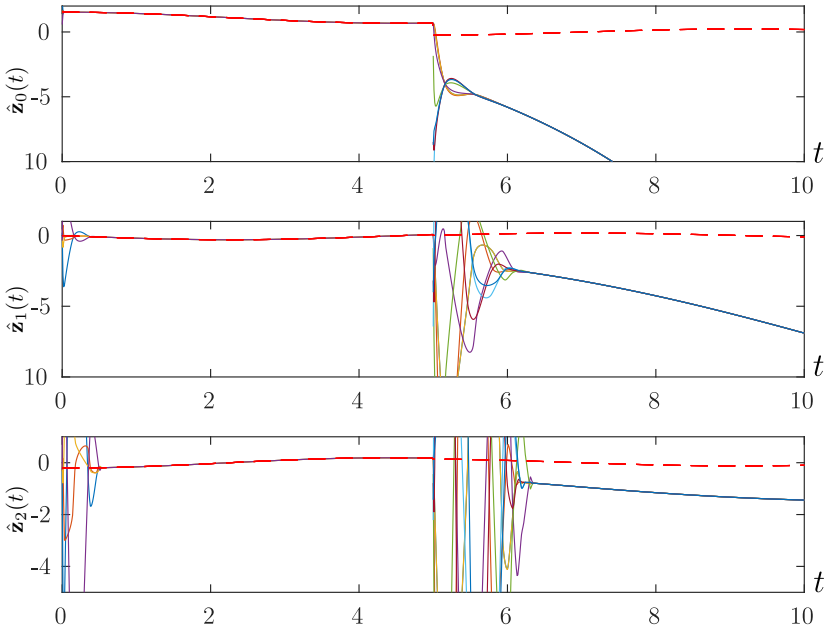

Figure 5.7: Trajectories for EDCHO in the scenario of Figure 5.5.

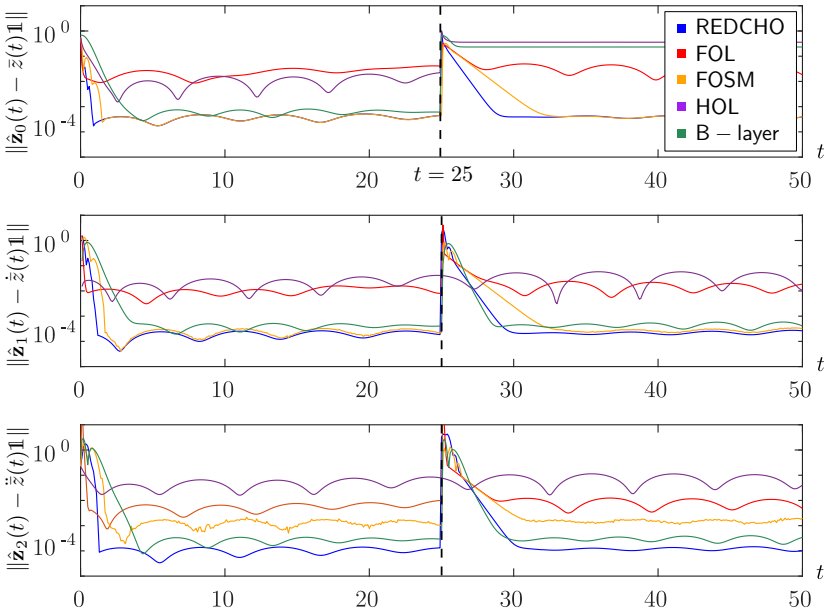

Figure 5.8: Comparison of the magnitude of the average consensus errors for REDCHO, FOL, FOSM, HOL and B-layer in the case of a sampling step $\Delta t = 10^{-6}$.

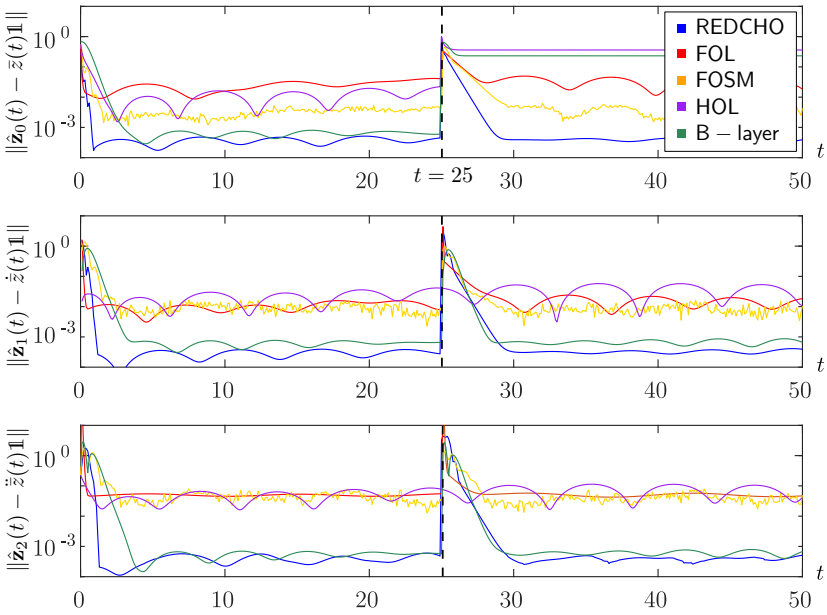

Figure 5.9: Comparison of the magnitude of the average consensus errors for REDCHO, FOL, FOSM, HOL and B-layer in the case of a sampling step $\Delta t = 10^{-3}$.

through a network. Proofs of convergence of the algorithm are given even when agents connect or disconnect from the network. Simulation scenarios were designed to confirm the advantages of the proposed protocol.

This extension of EDCHO allows to consider a mild class of OMAS systems, where changes in the connectivity between agents occur as infrequent isolated events. Considering general switching topologies is currently beyond the scope of this thesis. However, REDCHO still provides many advantages that make it applicable in various practical scenarios. In particular, in subsequent chapters, the ideas presented in this chapter will be extended to the context of asynchronous discrete-time communication, delays, and measurement noise. Moreover, REDCHO is directly applied to a formation control problem under perception latency in Chapter 7.

Chapter Six

Distributed Differentiation Protocol

The problem of online derivative computation or simply *differentiation* is fundamental in this thesis. To give an illustrative example of why differentiation algorithms are important, consider the dynamics of a single coordinate $x(t)$ for an arbitrary robot with second order dynamics $\ddot{x}(t) = u(t)$ where $u(t)$ is a scalar control input. If a reference $x_{\text{ref}}(t)$ is constructed for $x(t)$ to follow, we can design a controller as

$$u(t) = \ddot{x}_{\text{ref}}(t) - \kappa_1(x(t) - x_{\text{ref}}(t)) - \kappa_2(\dot{x}(t) - \dot{x}_{\text{ref}}(t)) \tag{6.1}$$

for design parameters $\kappa_1, \kappa_2 > 0$. The error $e(t) = x(t) - x_{\text{ref}}(t)$ has closed-loop dynamics $\ddot{e}(t) = -\kappa_1 e(t) - \kappa_2 \dot{e}(t)$ with stable origin, effectively achieving trajectory tracking.

However, even if the robot is able to measure $x(t)$, $x_{\text{ref}}(t)$, the values of $\dot{x}(t)$, $\dot{x}_{\text{ref}}(t)$, $\ddot{x}_{\text{ref}}(t)$ required in (6.1), have to be computed numerically in general. For instance, consider $x_{\text{ref}}(t) = \hat{x}_{\text{target}}(t) + d$ where $\hat{x}_{\text{target}}(t)$ is an estimated target coordinate at time t and $d \geq 0$ is a fixed displacement. Since estimations $\hat{x}_{\text{target}}(t)$ are produced by sensors, there is no explicit expression of $x_{\text{ref}}(t)$ so that its derivatives cannot be computed in closed form beforehand. Instead, $\dot{x}_{\text{ref}}(t), \ddot{x}_{\text{ref}}(t)$ are estimated at time t using prior and current data for $x_{\text{ref}}(t)$. Such numerical procedure is called a *differentiator*.

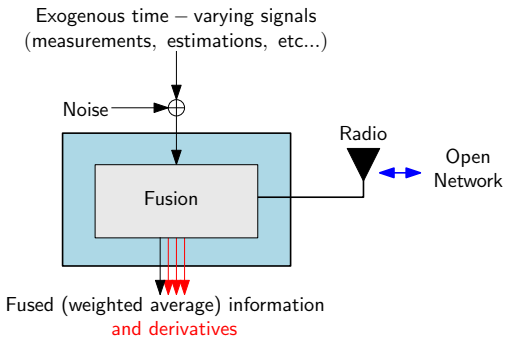

Figure 6.1: Individual agent architecture. No knowledge of input exogenous signal's derivatives is assumed. Measurement noise at the input is considered.

In a multi-agent setting, it can be beneficial to compute derivatives of quantities for which only local information is available. For example, computing derivatives of the consensus signal $\bar{z}(t)$ in Chapters 4 and 5. As will be discussed in Chapter 7, such solution can enable distributed target tracking and formation control. While EDCHO and REDCHO appear to compute the first m derivatives $\bar{z}(t)$, they still have important limitations in more practical scenarios. In particular, both EDCHO and REDCHO make some restrictive assumptions:

1. Having perfect knowledge of the local signal $z_i(t)$ at agent $i \in \mathcal{I}$.
2. Having knowledge of the m derivatives of $z_i(t)$ at each agent $i \in \mathcal{I}$ (see the output equations for EDCHO and REDCHO in (4.1) and (5.2) respectively).

Assuming perfect knowledge of $z_i(t)$ is very restrictive in the context of perception latency, where such signals may originate from perception measurements which are prone to noise. For the second assumption, estimating high-order derivatives of sensor data can be very challenging in practice, specially since an inappropriate differentiation method may amplify the measurement noise.

Henceforth, in this chapter we focus on a setting as in Figure 6.1. We present two main contributions in this setting which account for the two main drawbacks of prior EDC techniques mentioned before. First, we analyze techniques to mitigate measurement noise at the inputs through weighted averaging. Second, in order to avoid additional noise amplification, we develop a workaround to compute derivatives of the consensus signal without requiring knowledge of the derivatives of the inputs. To the best of our knowledge this is the first dynamic consensus protocol with a noise analysis of this kind. The contributions of this chapter were also published in [6].

6.1 RELATED WORK

The problem of numerical differentiation is closely related to estimating the state of a chain of integrators linear system with an unknown input [95]. This connection motivates the use Kalman filtering to estimate derivatives. Hence, the problem of distributed differentiation can be recasted as a distributed Kalman filtering problem for which many solutions exist in the literature [87, 94, 142, 160]. However, these type of protocols won't be able to compute the exact derivatives of the average signal even without measurement noise under persistently varying signals, due to their linear nature. A similar kind of differentiators are the linear High Gain Observers (HGO) [161] for the case of a single agent. In this case, arbitrarily small differentiation error can be achieved by increasing a gain in the algorithm for the noiseless case. However, this same feature makes it sensitive in the presence of noise. Similar linear strategies to the HGO have been extended for dynamics consensus such as in [84, 98, 162]. Nonetheless, these approaches still suffer from the problem that they cannot attain exact convergence even without noise, and may be sensitive to noise if high gains are used.

On the other hand, Robust Exact Differentiators (RED) [96] are non-linear observers that attain exact differentiation without noise for a class input signals with bounded high-order derivative. It was shown that the RED reaches optimally shaped error bounds. Despite this, there does not exist a distributed differentiator in the literature generalizing the capabilities of the RED. It is precisely this gap the one tackled in this chapter, presenting the first of its kind exact distributed differentiator.

6.2 PROBLEM STATEMENT

Consider the same network model as in Chapters 4 and 5 with the exception that each agent i has access to a local noisy signal

$$z_i^{\varepsilon}(t) = z_i(t) + \varepsilon_i(t)$$

with $(m+1)$-times differentiable nominal signal $z_i(t)$ and noise $\varepsilon_i(t)$. The goal of the distributed differentiation problem is for the agents to estimate the weighted average signal

$$\bar{z}(t) = \frac{w_1 z_1(t) + \cdots + w_{\mathsf{N}} z_{\mathsf{N}}(t)}{w_1 + \cdots + w_{\mathsf{N}}} \tag{6.2}$$

for locally known $w_1, \ldots, w_{\mathsf{N}} > 0$, as well as its first $m \in \mathbb{N}$ derivatives

$$\bar{z}^{(1)}(t), \ldots, \bar{z}^{(m)}(t),$$

in a decentralized fashion.

Remark 6.1. *Computing* (6.2) *is useful for robust estimation under noisy measurements or imperfect communication since w_i can be assigned with a lower value depending on the quality of node i. For example, in a sensor network measuring the same quantity $\bar{z}(t) = z_1(t) = \cdots = z_{\mathsf{N}}(t)$ with noise $\varepsilon_i(t) \sim \mathcal{N}(0, \sigma_i^2)$, one can choose $w_i = 1/\sigma_i^2$ such that* (6.2) *corresponds to the optimal estimate for $\bar{z}(t)$ due to the inverse-covariance average structure.*

In the following section we propose a Distributed Differentiation Protocol (DDP) in continuous time, able to estimate (6.2).

6.3 THE PROTOCOL

The proposed DDP has the following structure. We construct a new virtual graph $\mathcal{G}^{\mathsf{a}} = (\mathcal{I}^{\mathsf{a}}, \mathcal{E}^{\mathsf{a}})$ with the same local information as $\mathcal{G}$ in the following way. First, $\mathcal{I}^{\mathsf{a}} = \mathcal{I} \cup \{\mathsf{N}+1, \ldots, 2\mathsf{N}\}$ where N virtual nodes are created, one for each $i \in \mathcal{I}$. Second, $\mathcal{E} \subset \mathcal{E}^{\mathsf{a}}$ with the only difference being that any virtual node $i + \mathsf{N} \in \{\mathsf{N}+1, \ldots, 2\mathsf{N}\}$ is only connected to node $i \in \mathcal{I}$. In this sense, we say that agent $i \in \mathcal{I}$ owns both $i, i+\mathsf{N} \in \mathcal{I}^{\mathsf{a}}$ virtual nodes. Moreover, the resulting adjacency matrix $\mathbf{A}^{\mathsf{a}}$ for $\mathcal{G}^{\mathsf{a}}$ is

$$\mathbf{A}^{\mathsf{a}} = \begin{bmatrix} \mathbf{A} & \mathbf{I}_{\mathsf{N}} \\ \mathbf{I}_{\mathsf{N}} & \mathbf{0}_{\mathsf{N}\times\mathsf{N}} \end{bmatrix}$$

where $\mathbf{A} \in \{0,1\}^{\mathsf{N}\times\mathsf{N}}$ is the adjacency matrix of $\mathcal{G}$.

The DDP is based on the new graph $\mathcal{G}^{\mathsf{a}}$. An arbitrary agent $i \in \mathcal{I}$ has internal variables $\{\chi_{i,\mu}(t), \chi_{i+\mathsf{N},\mu}(t)\}_{\mu=0}^{m}$ and output estimations $\{\hat{z}_{i,\mu}(t)\}_{\mu=0}^{m}$ for the derivatives

$\{\bar{z}^{(\mu)}(t)\}_{\mu=0}^{m}$. The internal dynamics of the DDP and its output are defined according to

Protocol:

$$\begin{aligned}
\dot{\chi}_{i,\mu}(t) &= (k_\mu/w_i)\theta^{\mu+1}\sum_{j=1}^{2\mathsf{N}}\mathbf{A}^{\mathsf{a}}_{ij}F_\mu(y_i(t)-y_j(t)) + \chi_{i,\mu+1}(t) - \gamma_\mu\chi_{i,\mu}(t), \quad 0\leq\mu<m\\
\dot{\chi}_{i,m}(t) &= (k_m/w_i)\theta^{m+1}\sum_{j=1}^{2\mathsf{N}}\mathbf{A}^{\mathsf{a}}_{ij}F_m(y_i(t)-y_j(t)) - \gamma_m\chi_{i,m}(t)\\
y_i(t) &= \begin{cases} z_i^{\varepsilon}(t)-\chi_{i,0}(t) & i\in\{1,\ldots,\mathsf{N}\}\\ -\chi_{i,0}(t) & i\in\{\mathsf{N}+1,\ldots,2\mathsf{N}\}\end{cases}
\end{aligned}$$

Output:

$$\hat{z}_{i,\mu}(t) = -2\sum_{\nu=0}^{m}\mathbf{G}_{\mu+1,\nu+1}\chi_{i+\mathsf{N},\nu}(t) \tag{6.3}$$

where $i\in\{1,\ldots,2\mathsf{N}\}$, $F_\mu(\bullet)=\lceil\bullet\rfloor^{\frac{m-\mu}{m+1}}$, $\mu\in\{0,\ldots,m\}$, $\mathbf{A}^{\mathsf{a}}_{ij}$ are the components of $\mathbf{A}^{\mathsf{a}}$, $\mathbf{G}_{\mu+1,\nu+1}$ are the components of the matrix $\mathbf{G}$ in (5.1), the $\theta,\{\gamma_\mu,k_\mu\}_{\mu=0}^{m}$ are design parameters and $w_{i+\mathsf{N}}=w_i, i\in\{1,\ldots,\mathsf{N}\}$.

Note from (6.3) that an arbitrary node $i\in\{1,\ldots,2\mathsf{N}\}$ only requires knowledge of its local weight value w_i, and no condition is imposed on these weights other than being strictly positive.

Remark 6.2. *Some important comments must be made for the DDP in* (6.3)*. First, it can be observed that each agent $i\in\mathcal{I}$ only needs to communicate $y_i(t)$ to its neighbors. On the other hand, each agent $i\in\mathcal{I}$ must run the dynamics for both $\{\chi_{i,\mu}(t)\}_{\mu=0}^{m}$ and $\{\chi_{i+\mathsf{N},\mu}(t)\}_{\mu=0}^{m}$. Moreover, the output of the DDP in* (6.3) *does not require local estimations for the derivatives of $z_i(t)$. This allows the accuracy of* (6.3) *under noise to depend only on the protocol parameters and the graph $\mathcal{G}$.*

Intuitively, the extra node $i+\mathsf{N}$ added to the virtual graph $\mathcal{G}^{\mathsf{a}}$ act as a differentiator estimating $z_i^{(\mu)}(t)$, in a way that its performance is tightly coupled to the rest of the protocol.

Theorem 6.3. *Let, $w_1,\ldots,w_{\mathsf{N}}>0$, $\mathcal{G}$ be a connected graph and the noise signals satisfy $|\varepsilon_i(t)|\leq\varepsilon,\forall t\geq T$, $\varepsilon\in[0,\varepsilon_{\max}]$ for some $\varepsilon_{\max}>0$. Then, for any $\varepsilon_{\max}>0$ there exist $\gamma_{\max}>0$ sufficiently small with $\gamma_0,\ldots,\gamma_m\in(0,\gamma_{\max}]$ such that if Assumption 5.2 holds, there exists:*

- *convergence time $T>0$ and sufficiently big $\theta,\{k_\mu\}_{\mu=0}^{m}$,*
- *admissible balls $\mathcal{R}_0,\ldots,\mathcal{R}_m\subset\mathbb{R}$ for the initial conditions $\{\chi_{i,\mu}(0)\}_{\mu=0}^{m}$,*
- *m-times differentiable consensus signal $\zeta(t)$,*

such that the DDP in (6.3) *complies:*

1. *$|\zeta^{(\mu)}(t)-\hat{z}_{i,\mu}(t)|\leq c_\mu\varepsilon^{\frac{m-\mu+1}{m+1}}$, $\forall t\geq T$ and all $i\in\{1,\ldots,N\}$ with constants $c_\mu>0$ which only depend on the protocol parameters, $\mathcal{G}$ and $\varepsilon_{\max}$.*

2. $\lim_{t\to\infty} |\zeta(t) - \bar{z}(t)| = 0$.

3. *The balls* $\mathcal{R}_0, \ldots, \mathcal{R}_m$ *can be made arbitrarily large by increasing* θ.

Remark 6.4. *The best possible differentiator accuracy under noise of magnitude* $\varepsilon > 0$ *must be proportional to* $\varepsilon^{\frac{m-\mu+1}{m+1}}$ *when the clear-of-noise signal has bounded* m*-th derivative [96]. This is the case of item 1 of Theorem 6.3 under Assumption 5.2. Moreover, as will be shown in Section 6.5, a linear version of* (6.3) *with* $F_\mu(\bullet) = (\bullet)$ *will have error bounds growing linearly with* ε*. This implies that the proposed DDP can outperform the linear protocols in the presence of noise.*

6.4 PROTOCOL CONVERGENCE

The convergence proof strategy for the DDP is the following. First, we show convergence with $\varepsilon = 0$ and $\gamma_0 = \cdots = \gamma_m = 0$. Then, we extend the result to the case with $\gamma_0, \ldots, \gamma_m > 0$. Finally, we extend the result for the case with $\varepsilon > 0$. These arguments follow a very similar reasoning as in Chapters 4 and 5 and are organized in the following technical lemmas.

Lemma 6.5. *Let* $\mathcal{G}$ *be connected, fixed* $\varepsilon = 0, \theta = 1$*, Assumption 5.2 hold with* $\gamma_0 = \cdots = \gamma_m = 0$ *and initial conditions for* (6.3) *comply*

$$\sum_{i=1}^{2\mathsf{N}} w_i \chi_{i,\mu}(0) = 0.$$

Then, there exists $T > 0$ *and sufficiently big* $\{k_\mu\}_{\mu=0}^m$ *such that*

$$|\hat{z}_{i,\mu}(t) - \bar{z}^{(\mu)}(t)| = 0, \forall t \geq T$$

Proof: First, we write (6.3) in compact form as follows. Define $z_i(t) = 0, \forall i \in \{\mathsf{N}+1, \ldots, 2\mathsf{N}\}$. Let

$$\boldsymbol{\chi}_\mu(t) = \begin{bmatrix} \chi_{1,\mu}(t) \\ \vdots \\ \chi_{2\mathsf{N},\mu}(t) \end{bmatrix}, \quad \mathbf{z}(t) = \begin{bmatrix} z_1(t) \\ \vdots \\ z_{2\mathsf{N}}(t) \end{bmatrix}, \quad \mathbf{y}_\mu(t) = \mathbf{z}^{(\mu)}(t) - \boldsymbol{\chi}_\mu(t).$$

In this notation, (6.3) with $\varepsilon = \gamma_0 = \cdots = \gamma_m = 0, \theta = 1$ can be written as:

$$\begin{aligned} \dot{\boldsymbol{\chi}}_\mu(t) &= \boldsymbol{\chi}_{\mu+1}(t) + k_\mu \mathbf{W}^{-1}\mathbf{D} \left\lceil \mathbf{D}^\top \mathbf{y}_0(t) \right\rfloor^{\frac{m-\mu}{m+1}} \quad \text{for } 0 \leq \mu \leq m-1, \\ \dot{\boldsymbol{\chi}}_m(t) &= k_m \mathbf{W}^{-1}\mathbf{D} \left\lceil \mathbf{D}^\top \mathbf{y}_0(t) \right\rfloor^{0} \end{aligned} \tag{6.4}$$

with $\mathbf{W} = \mathrm{diag}(\mathbf{w}), \mathbf{w} = [w_1, \ldots, w_{2\mathsf{N}}]^\top$ and $\mathbf{D}$ the incidence matrix of $\mathcal{G}^{\mathsf{a}}$. Moreover, the assumption on the initial conditions can be written as $\mathbf{w}^\top \boldsymbol{\chi}_\mu(0) = 0$. Note that $\mathbf{w}^\top \dot{\boldsymbol{\chi}}_m(t) = 0$ since $\mathbf{w}^\top \mathbf{W}^{-1}\mathbf{D} = \mathbb{1}^\top \mathbf{D} = 0$ [158, Page 280]. Hence, the quantity $\mathbf{w}^\top \boldsymbol{\chi}_m(t) = 0$ remains invariant $\forall t \geq 0$. This implies $\mathbf{w}^\top \boldsymbol{\chi}_\mu(t) = 0, \forall t \geq 0, \forall \mu \in$

$\{0,\dots,m\}$ by induction over μ. Now, let the consensus error $\tilde{\mathbf{y}}_\mu(t) = \mathbf{P}\mathbf{y}_\mu(t)$ with $\mathbf{P} = (\mathbf{I}_{\mathsf{N}} - \mathbb{1}\mathbf{w}^\top/(\mathbb{1}^\top\mathbf{w}))$. The dynamics of $\tilde{\mathbf{y}}_\mu(t)$ can be written as:

$$\begin{aligned}\dot{\tilde{\mathbf{y}}}_\mu(t) &= \tilde{\mathbf{y}}_{\mu+1}(t) - k_\mu\mathbf{W}^{-1}\mathbf{D}\left\lceil\mathbf{D}^\top\tilde{\mathbf{y}}_0(t)\right\rfloor^{\frac{m-\mu}{m+1}} \quad \text{for } 0\leq\mu\leq m-1,\\ \dot{\tilde{\mathbf{y}}}_m(t) &= \tilde{\mathbf{z}}^{(m+1)}(t) - k_m\mathbf{W}^{-1}\mathbf{D}\left\lceil\mathbf{D}^\top\tilde{\mathbf{y}}_0(t)\right\rfloor^{0}\end{aligned}$$

with $\tilde{\mathbf{z}}(t) = \mathbf{P}\mathbf{z}(t)$. Note that $\tilde{\mathbf{z}}^{(m+1)}(t)$ is uniformly bounded for all $t \geq 0$ due to Assumption 5.2. Hence, it follows that there exists $T > 0$ and sufficiently big $\{k_\mu\}_{\mu=0}^m$ such that $\tilde{\mathbf{y}}_\mu(t) = 0, \forall t \geq T, \forall\mu \in \{0,\dots,m\}$ as per Theorem 4.7 in Chapter 4. Therefore, $\mathbf{y}_\mu(t) = \bar{y}_\mu(t)\mathbb{1}, \forall t \geq T$ and some scalar functions $\bar{y}_0(t),\dots,\bar{y}_m(t)$. Moreover,

$$\bar{y}_\mu(t) = \frac{\mathbf{w}^\top\mathbf{y}_\mu(t)}{\mathbf{w}^\top\mathbb{1}} = \frac{\mathbf{w}^\top\mathbf{z}^{(\mu)}(t)}{\mathbf{w}^\top\mathbb{1}} = \frac{\sum_{i=1}^{\mathsf{N}} w_i z_i^{(\mu)}(t)}{2\sum_{i=1}^{\mathsf{N}} w_i} = \frac{\bar{z}^{(\mu)}(t)}{2} \tag{6.5}$$

due to $\mathbf{w}^\top\boldsymbol{\chi}_\mu(t) = 0$ and $z_{i+\mathsf{N}}(t) = 0, w_{i+\mathsf{N}} = w_i$ for $i \in \{1,\dots,\mathsf{N}\}$. Finally, $\gamma_0 = \cdots = \gamma_m = 0$ implies $\mathbf{G} = \mathbf{I}_m$ such that $\hat{z}_{i,\mu}(t) = -2\chi_{i+\mathsf{N},\mu}(t) = 2y_{i+\mathsf{N},\mu}(t)$ from (6.3) where $\{y_{i,\mu}(t)\}_{i=1}^{2\mathsf{N}}$ are the components of $\mathbf{y}_\mu(t)$. Hence, the result follows since

$$|\hat{z}_{i,\mu}(t) - \bar{z}^{(\mu)}(t)| = |2y_{i+\mathsf{N},\mu}(t) - \bar{z}^{(\mu)}(t)| = |2\bar{y}_\mu(t) - \bar{z}^{(\mu)}(t)| = 0, \forall t \geq T.$$

□

Lemma 6.6. *Theorem 6.3 is true for* $\varepsilon_{\max} = 0$.

Proof: First, consider the recursive sequence of filters

$$\mathbf{s}_0(t) = \mathbf{z}(t), \quad \mathbf{s}_{\mu+1}(t) = \left(\frac{\mathrm{d}}{\mathrm{d}t} + \gamma_\mu\right)\mathbf{s}_\mu(t) \tag{6.6}$$

for $\mu \in \{0,\dots,m\}$ with $\mathbf{z}(t) = [z_1(t),\dots,z_{\mathsf{N}}(t),0,\dots,0]^\top$. Note that

$$\mathbf{s}_{m+1}(t) = \left(\frac{\mathrm{d}}{\mathrm{d}t} + \gamma_m\right)\cdots\left(\frac{\mathrm{d}}{\mathrm{d}t} + \gamma_0\right)\mathbf{z}(t) = \mathbf{z}^{(m+1)}(t) + \sum_{\mu=0}^{m} l_\mu\mathbf{z}^{(\mu)}(t)$$

where $l_0,\dots,l_m$ are defined in Assumption 5.2. Now, define $\mathbf{y}_\mu(t) = \mathbf{s}_\mu(t) - \boldsymbol{\chi}_\mu(t)$ with $\boldsymbol{\chi}_\mu(t) = [\chi_{1,\mu}(t),\dots,\chi_{2\mathsf{N},\mu}(t)]^\top$. Decompose $\mathbf{y}_\mu(t) = \bar{y}_\mu(t)\mathbb{1} + \tilde{\mathbf{y}}_\mu(t)$ in the consensus component $\bar{y}_\mu(t) = (\mathbf{w}^\top/(\mathbb{1}^\top\mathbf{w}))\mathbf{y}_\mu(t)$ and the consensus error $\tilde{\mathbf{y}}_\mu(t) = \mathbf{P}\mathbf{y}_\mu(t)$ with $\mathbf{P} = (\mathbf{I}_{\mathsf{N}} - \mathbb{1}\mathbf{w}^\top/(\mathbb{1}^\top\mathbf{w}))$. Then, the dynamics of the consensus error $\tilde{\mathbf{y}}_\mu(t)$ complies

$$\begin{aligned}\dot{\tilde{\mathbf{y}}}_\mu(t) &= \mathbf{P}\dot{\mathbf{s}}_\mu(t) - k_\mu\theta^{\mu+1}\mathbf{P}\mathbf{W}^{-1}\mathbf{D}\left\lceil\mathbf{D}^\top\mathbf{y}_0(t)\right\rfloor^{\frac{m-\mu}{m+1}} - \mathbf{P}\boldsymbol{\chi}_{\mu+1}(t) + \gamma_\mu\mathbf{P}\boldsymbol{\chi}_\mu(t)\\ &= -k_\mu\theta^{\mu+1}\mathbf{W}^{-1}\mathbf{D}\left\lceil\mathbf{D}^\top\tilde{\mathbf{y}}_0(t)\right\rfloor^{\frac{m-\mu}{m+1}} + \mathbf{P}(\mathbf{s}_{\mu+1}(t) - \boldsymbol{\chi}_{\mu+1}(t)) - \gamma_\mu\mathbf{P}(\mathbf{s}_\mu(t) - \boldsymbol{\chi}_\mu(t))\\ &= -k_\mu\theta^{\mu+1}\mathbf{W}^{-1}\mathbf{D}\left\lceil\mathbf{D}^\top\tilde{\mathbf{y}}_0(t)\right\rfloor^{\frac{m-\mu}{m+1}} + \tilde{\mathbf{y}}_{\mu+1}(t) - \gamma_\mu\tilde{\mathbf{y}}_\mu(t)\\ \dot{\tilde{\mathbf{y}}}_m(t) &= -k_m\theta^{m+1}\mathbf{W}^{-1}\mathbf{D}\left\lceil\mathbf{D}^\top\tilde{\mathbf{y}}_0(t)\right\rfloor^{0} + \mathbf{P}\mathbf{s}_{m+1}(t) - \gamma_m\tilde{\mathbf{y}}_m(t)\end{aligned} \tag{6.7}$$

for $\mu \in \{0,\dots,m-1\}$ by using $\mathbf{P}\mathbf{W}^{-1}\mathbf{D}^\top = \mathbf{W}^{-1}\mathbf{D}^\top$, $\mathbf{D}^\top\mathbf{P} = \mathbf{D}^\top$, $\dot{\mathbf{s}}_\mu(t) = \mathbf{s}_{\mu+1}(t) - \gamma_\mu\mathbf{s}_\mu(t)$ from (6.6) and that $\mathbf{y}_0(t) = \mathbf{z}(t) - \boldsymbol{\chi}_0(t)$. Note that (6.7) reduces to (6.4) when

$\gamma_0 = \cdots = \gamma_m = 0, \theta = 1$. Hence, from Lemma 6.5 and an homogeneity argument [2, 163], existence of $\theta, \{k_\mu\}_{\mu=0}^m$ is guaranteed for fixed $\gamma_0, \ldots, \gamma_m > 0$ such that $\tilde{\mathbf{y}}_\mu(t) = 0, \forall t \geq T$ provided that the initial conditions $\tilde{\mathbf{y}}_\mu(0)$ are sufficiently close to the origin. This ensures the existence of the admissible balls $\{\mathcal{R}_\mu\}_{\mu=0}^m$ for the initial conditions $\{\chi_{i,\mu}(0)\}_{\mu=0}^m$ and its dependence with respect to θ.

Now, the dynamics of the consensus component $\bar{y}_\mu(t)$ can be obtained in a similar fashion as:

$$\begin{aligned} \dot{\bar{y}}_\mu(t) &= \bar{y}_{\mu+1}(t) - \gamma_\mu \bar{y}_\mu(t) \\ \dot{\bar{y}}_m(t) &= \mathbf{w}^\top \mathbf{s}_{m+1}(t)/(\mathbb{1}^\top \mathbf{w}) - \gamma_m \bar{y}_m(t) \end{aligned} \tag{6.8}$$

Note that $\mathbf{w}^\top \mathbf{z}(t)/(\mathbb{1}^\top \mathbf{w}) = \bar{z}(t)/2$ from (6.5). By the definition of the matrix $\mathbf{G}$, the transformation

$$\hat{z}_\mu(t) := 2 \sum_{\nu=0}^{m} \mathbf{G}_{\mu+1,\nu+1} \bar{y}_\mu(t)$$

transforms the linear system (6.8) into the canonical form:

$$\begin{aligned} \dot{\hat{z}}_\mu(t) &= \hat{z}_{\mu+1}(t) \\ \dot{\hat{z}}_m(t) &= \bar{z}^{(m+1)}(t) - \sum_{\mu=0}^{m} l_\mu (\hat{z}_\mu(t) - \bar{z}^{(\mu)}(t)) \end{aligned} \tag{6.9}$$

by the definition of $\mathbf{s}_{m+1}(t)$. Let $\hat{z}(t) := \hat{z}_0(t)$ such that $\hat{z}^{(\mu)} = \hat{z}_\mu(t), \mu \leq m$ is obtained from (6.9). Hence, defining the error $\tilde{z}(t) = \hat{z}(t) - \bar{z}(t)$ one obtains

$$\tilde{z}^{(m+1)}(t) = -\sum_{\mu=0}^{m} l_\mu \tilde{z}^{(\mu)}(t)$$

which by definition of $\{l_\mu\}_{\mu=0}^m$ is a linear system with poles in $-\gamma_0, \ldots, -\gamma_m < 0$ and thus converge to the origin asymptotically. Therefore, $\hat{z}^{(\mu)}(t)$ converge towards $\bar{z}^{(\mu)}(t)$ asymptotically. Similarly as before, due to the definition of $\mathbf{G}$, it follows that

$$\mathbf{z}^{(\mu)}(t) = \sum_{\nu=0}^{m} \mathbf{G}_{\mu+1,\nu+1} \mathbf{s}_\nu(t)$$

is the inverse transformation of (6.6). Then, define the consensus derivative estimation as

$$\begin{aligned} \hat{\mathbf{z}}_\mu(t) &:= 2 \left(\mathbf{z}^{(\mu)}(t) - \sum_{\nu=0}^{m} \mathbf{G}_{\mu+1,\nu+1} \boldsymbol{\chi}_\nu(t) \right) \\ &= 2 \sum_{\nu=0}^{m} \mathbf{G}_{\mu+1,\nu+1} \mathbf{y}_\nu(t) = \hat{z}^{(\mu)}(t) \mathbb{1} + 2 \sum_{\nu=0}^{m} \mathbf{G}_{\mu+1,\nu+1} \tilde{\mathbf{y}}_\nu(t) \end{aligned} \tag{6.10}$$

using the decomposition of

$$\mathbf{y}_\mu(t) = \bar{y}_\mu(t) \mathbb{1} + \tilde{\mathbf{y}}_\mu(t)$$

as well as from the definition

$$\hat{z}^{(\mu)}(t) = \hat{z}_\mu(t) = 2 \sum_{\nu=0}^{m} \mathbf{G}_{\mu+1,\nu+1} \bar{y}_\nu(t)$$

given before. Hence, for $t \geq T$, $\hat{\mathbf{z}}_\mu(t) = \hat{z}^{(\mu)}(t)\mathbb{1}$ from which the outputs $\hat{z}_{i,\mu}(t)$ in (6.3) can be identified with the lower half of the components of $\hat{\mathbf{z}}_\mu(t)$ due to $z_i(t) = 0, i \in \{\mathsf{N}+1, \dots, 2\mathsf{N}\}$ completing the proof. □

Lemma 6.7. *Let* $\mathbf{W} = \mathit{diag}(\mathbf{w}), \mathbf{w} = [w_1, \dots, w_{2\mathsf{N}}]^\top$ *and the differential inclusion:*

$$\begin{aligned}\dot{\tilde{\mathbf{y}}}_\mu(t) &= -k_\mu \theta^{\mu+1} \mathbf{W}^{-1} \mathbf{D} \left\lceil \mathbf{D}^\top (\tilde{\mathbf{y}}_0(t) + \boldsymbol{\varepsilon}(t)) \right\rfloor^{\frac{m-\mu}{m+1}} + \tilde{\mathbf{y}}_{\mu+1}(t) - \gamma_\mu \tilde{\mathbf{y}}_\mu(t) \\ &\qquad \text{for } 0 \leq \mu \leq m-1, \\ \dot{\tilde{\mathbf{y}}}_m(t) &\in -k_m \theta^{m+1} \mathbf{W}^{-1} \mathbf{D} \left\lceil \mathbf{D}^\top (\tilde{\mathbf{y}}_0(t) + \boldsymbol{\varepsilon}(t)) \right\rfloor^{0} + [-L', L']^{2\mathsf{N}} - \gamma_m \tilde{\mathbf{y}}_m(t)\end{aligned} \tag{6.11}$$

for $\tilde{\mathbf{y}}_\mu(t) \in \mathbb{R}^{2\mathsf{N}}$, $L' > 0$. *Moreover, let* $\theta, \{k_\mu\}_{\mu=0}^m$ *be chosen such that* (6.11) *is finite time stable with* $\varepsilon = 0$ *and any* $\gamma_0, \dots, \gamma_m \in [0, 1)$. *Then, for every* $\varepsilon_{\max} > 0$ *there exists* $\gamma_{\max} > 0$ *such that if* $\gamma_0, \dots, \gamma_m \in [0, \gamma_{\max})$ *and the noise is uniformly bounded as* $\|\boldsymbol{\varepsilon}(t)\|_\infty \leq \varepsilon$ *with* $\varepsilon \in [0, \varepsilon_{\max})$ *then there exist* $T, \tilde{c}_0, \dots, \tilde{c}_m > 0$ *such that* $\|\tilde{\mathbf{y}}_\mu(t)\| \leq \tilde{c}_\mu \varepsilon^{\frac{m-\mu+1}{m+1}}, \forall t \geq T$.

Proof: Note that for fixed $\gamma = [\gamma_0, \dots, \gamma_m]^\top \in [0, 1)^{m+1}$, existence of sufficiently big parameters $\theta^\gamma, \{k_\mu^\gamma\}_{\mu=0}^m$ such that (6.11) is finite time stable with $\varepsilon = 0$ is guaranteed by Lemma 6.6 since (6.11) is equivalent to (6.7) under Assumption 5.2. Hence, the parameters $\sup_\gamma \theta^\gamma, \{\sup_\gamma k_\mu^\gamma\}_{\mu=0}^m$ exist and ensure finite time stability of (6.11) for $\varepsilon = 0$ and any $\gamma_0, \dots, \gamma_m \in [0, 1)$. Now, pick some $\varepsilon' > 0$ and let $\eta = (\varepsilon'/\varepsilon)^{1/(m+1)}$. Then, set $\gamma_{\max} = (\varepsilon'/\varepsilon_{\max})^{1/(m+1)}$. Consider the change of coordinates

$$t' = \eta t, \quad \tilde{\mathbf{y}}'_\mu(t) = \eta^{m-\mu+1} \tilde{\mathbf{y}}_\mu(t), \quad \boldsymbol{\varepsilon}'(t) = \eta^{m+1} \boldsymbol{\varepsilon}(t)$$

Hence,

$$\begin{aligned}\frac{\mathrm{d}\tilde{\mathbf{y}}'_\mu}{\mathrm{d}t'} &= \eta^{-1} \dot{\tilde{\mathbf{y}}}'_\mu \\ &= \eta^{m-\mu} \dot{\tilde{\mathbf{y}}}_\mu = \eta^{m-\mu} \tilde{\mathbf{y}}_{\mu+1} - \gamma_\mu \eta^{m-\mu} \tilde{\mathbf{y}}_\mu - k_\mu \theta^{\mu+1} \mathbf{W}^{-1} \eta^{m-\mu} \mathbf{D} \left\lceil \mathbf{D}^\top (\tilde{\mathbf{y}}_0 + \boldsymbol{\varepsilon}) \right\rfloor^{\frac{m-\mu}{m+1}} \\ &= \tilde{\mathbf{y}}'_{\mu+1} - (\gamma_\mu/\eta) \tilde{\mathbf{y}}'_\mu - k_\mu \theta^{\mu+1} \mathbf{W}^{-1} \mathbf{D} \left\lceil \mathbf{D}^\top (\eta^{m+1} \tilde{\mathbf{y}}_0 + \eta^{m+1} \boldsymbol{\varepsilon}) \right\rfloor^{\frac{m-\mu}{m+1}} \\ &= \tilde{\mathbf{y}}'_{\mu+1} - (\gamma_\mu/\eta) \tilde{\mathbf{y}}'_\mu - k_\mu \theta^{\mu+1} \mathbf{W}^{-1} \mathbf{D} \left\lceil \mathbf{D}^\top (\tilde{\mathbf{y}}'_0 + \boldsymbol{\varepsilon}') \right\rfloor^{\frac{m-\mu}{m+1}} \\ \frac{\mathrm{d}\tilde{\mathbf{y}}'_m}{\mathrm{d}t'} &\in -k_m \theta^{m+1} \mathbf{W}^{-1} \mathbf{D} \left\lceil \mathbf{D}^\top (\tilde{\mathbf{y}}'_0 + \boldsymbol{\varepsilon}') \right\rfloor^{0} + [-L', L']^{2\mathsf{N}} - (\gamma_m/\eta) \tilde{\mathbf{y}}'_m.\end{aligned}$$

Therefore, the dynamics of $\tilde{\mathbf{y}}'_\mu(t')$ are equivalent to the ones in (6.11) with the new noise $\boldsymbol{\varepsilon}'(t')$ and the new gains

$$\gamma_0/\eta, \dots, \gamma_m/\eta < \gamma_{\max}/\eta = (\varepsilon/\varepsilon_{\max})^{1/(m+1)} \leq 1.$$

Moreover, the new noise comply

$$|\boldsymbol{\varepsilon}'(t')| = \eta^{m+1} |\boldsymbol{\varepsilon}(t')| \leq \eta^{m+1} \varepsilon = \varepsilon'$$

by the definition of η. If we pick $\varepsilon' = 0$ in the noiseless case, then by assumption $\tilde{\mathbf{y}}'_\mu(t')$ is finite time stable towards the origin since all $\gamma_\mu/\eta \in (0, 1)$. Hence, there exist sufficiently small $\varepsilon' > 0$ and constants $T', c'_0, \dots, c'_\mu$ such that

$$\|\tilde{\mathbf{y}}'_\mu(t')\| \leq c'_\mu, \forall t' \geq T'$$

by continuity of solutions. This means that there exists $T = T'/\eta > 0$ such that by setting

$$\tilde{c}_\mu = c'_\mu (\varepsilon')^{\frac{\mu-m+1}{m+1}}$$

and using $\|\tilde{\mathbf{y}}'_\mu(t)\| \leq c'_\mu$, $\eta = (\varepsilon'/\varepsilon)^{1/(m+1)}$ we have that (6.11) complies the following $\forall t \geq T$:

$$\|\tilde{\mathbf{y}}_\mu(t)\| = \eta^{\mu-m-1}\|\tilde{\mathbf{y}}'_\mu(t)\| \leq \tilde{c}_\mu \varepsilon^{\frac{m-\mu+1}{m+1}},$$

completing the proof. □

Lemma 6.8. *Let fixed $\tilde{c}_0, \ldots, \tilde{c}_m > 0$ and define*

$$I_\mu(\varepsilon) := 2\sum_{\nu=0}^{m} \tilde{c}_\mu \mathbf{G}_{\mu+1,\nu+1} \varepsilon^{\frac{m-\nu+1}{m+1}}.$$

Then, for any $\varepsilon_{\max} > 0$ there exists $c_\mu > 0$ such that $\forall \varepsilon \in [0, \varepsilon_{\max}]$ it follows that $I_\mu(\varepsilon) \leq c_\mu \varepsilon^{\frac{m-\mu+1}{m+1}}$.

Proof: First, note that $\mathbf{G}$ is a lower triangular matrix with only ones in the diagonal. Hence,

$$\begin{aligned} I_\mu(\varepsilon) &= 2\tilde{c}_\mu \varepsilon^{\frac{m-\mu+1}{m+1}} + 2\sum_{\nu=0}^{\mu-1} \tilde{c}_\nu \mathbf{G}_{\mu+1,\nu+1} \varepsilon^{\frac{m-\nu+1}{m+1}} \\ &\leq 2\varepsilon^{\frac{m-\mu+1}{m+1}} \left(\tilde{c}_\mu + \sum_{\nu=0}^{\mu-1} \tilde{c}_\nu \mathbf{G}_{\mu+1,\nu+1} \varepsilon^{\frac{\mu-\nu}{m+1}} \right) \end{aligned}$$

where $\mathbf{G}_{\mu+1,\mu+1} = 1$ and $\mathbf{G}_{\mu+1,\nu+1} = 0, \nu > \mu$ was used. Thus, since $\varepsilon \in [0, \varepsilon_{\max}]$ and $\mu > \nu$ then, the result follows with

$$c_\mu = 2\left(\tilde{c}_\mu + \sum_{\nu=0}^{\mu-1} \tilde{c}_\nu \mathbf{G}_{\mu+1,\nu+1} (\varepsilon_{\max})^{\frac{\mu-\nu}{m+1}} \right) > 0.$$

□

6.4.1 Proof of Theorem 6.3

Similar to the proof of Lemma 6.6, define $\mathbf{y}_\mu(t) = \mathbf{s}_\mu(t) - \boldsymbol{\chi}_\mu(t)$ with

$$\boldsymbol{\chi}_\mu(t) = \begin{bmatrix} \chi_{1,\mu}(t) \\ \vdots \\ \chi_{2N,\mu}(t) \end{bmatrix}$$

and compute the dynamics of the consensus error $\tilde{\mathbf{y}}_\mu(t) = \mathbf{P}\mathbf{y}_\mu(t)$ with $\mathbf{P} = (\mathbf{I}_N - \mathbb{1}\mathbf{w}^\top/(\mathbb{1}^\top \mathbf{w}))$:

$$\begin{aligned} \dot{\tilde{\mathbf{y}}}_\mu(t) &= -k_\mu \theta^{\mu+1} \mathbf{W}^{-1}\mathbf{D} \left\lceil \mathbf{D}^\top(\tilde{\mathbf{y}}_0(t) + \boldsymbol{\varepsilon}(t)) \right\rfloor^{\frac{m-\mu}{m+1}} + \tilde{\mathbf{y}}_{\mu+1}(t) - \gamma_\mu \tilde{\mathbf{y}}_\mu(t) \\ \dot{\tilde{\mathbf{y}}}_m(t) &= -k_m \theta^{m+1} \mathbf{W}^{-1}\mathbf{D} \left\lceil \mathbf{D}^\top(\tilde{\mathbf{y}}_0(t) + \boldsymbol{\varepsilon}(t)) \right\rfloor^{0} + \mathbf{P}\mathbf{s}_{m+1}(t) - \gamma_m \tilde{\mathbf{y}}_m(t) \end{aligned}$$

This is, the dynamics of $\tilde{\mathbf{y}}_\mu(t)$ are equivalent to (6.11) since by Assumption 5.2, we have that $\mathbf{Ps}_{m+1}(t)$ is bounded in $[-L', L']^{2\mathsf{N}}$ for some $L' > 0$. Hence, Lemma 6.6 ensures the existence of gains $\theta, \{k_\mu\}_{\mu=0}^m$ such to make (6.11) converge to the origin when $\varepsilon = 0$. Thus, given $\varepsilon_{\max} > 0$ pick $\gamma_{\max} > 0$ and gains as in Lemma 6.6 to conclude that

$$\|\tilde{\mathbf{y}}_\mu(t)\| \leq \tilde{c}_\mu \varepsilon^{\frac{m-\mu+1}{m+1}}, \forall t \geq T$$

for some $T, \tilde{c}_0, \ldots, \tilde{c}_m > 0$ using Lemma 6.7. Let $\hat{\mathbf{z}}_\mu(t)$ defined as in (6.10). Then, the consensus derivative estimation $\mathbf{P}\hat{\mathbf{z}}_\mu(t)$ complies

$$\|\mathbf{P}\hat{\mathbf{z}}_\mu(t)\| \leq 2\sum_{\nu=0}^{m} \mathbf{G}_{\mu+1,\nu+1}\|\tilde{\mathbf{y}}_\nu(t)\| \leq 2\sum_{\nu=0}^{m} \tilde{c}_\mu \mathbf{G}_{\mu+1,\nu+1}\varepsilon^{\frac{m-\nu+1}{m+1}} = I_\mu(\varepsilon)$$

with $I_\mu(\varepsilon)$ defined in Lemma 6.8. Thus, $\|\mathbf{P}\hat{\mathbf{z}}_\mu(t)\| \leq c_\mu \varepsilon^{\frac{m-\mu+1}{m+1}}$ with c_μ as in Lemma 6.8. Henceforth, following a similar reasoning as in the decomposition in (6.10), $\hat{\mathbf{z}}_\mu(t)$ converge towards consensus around $\hat{z}^{(\mu)}(t)$ except for an error of magnitude proportional to $\varepsilon^{\frac{m-\mu+1}{m+1}}$. Finally, the rest of the items are inherited from Lemma 6.6 as well.

6.5 SIMULATION EXAMPLES

In these examples we consider a circular network $\mathcal{G}$ of $\mathsf{N} = 10$ nodes. In addition, for the sake of brevity, we only consider $m = 1$ such that the problem is reduced to computing a first order derivative of the weighted average signal. In all cases, we consider nominal signals of the form $z_i(t) = A_i \cos(\omega_i t + \phi_i)$ where $\{A_i, \omega_i, \phi_i\}_{i=1}^{\mathsf{N}}$ and $\{\chi_{i,\mu}(0)\}_{i=1,\mu=0}^{2\mathsf{N},m}$ were chosen randomly. We used parameters $\theta = 2, k_1 = 1, k_2 = 2, \gamma_1 = \gamma_2 = 1$. Moreover, for the sake of generality we chose $w_1, \ldots, w_{\mathsf{N}}$ randomly on the interval $(0, 1)$. Protocol (6.3) was simulated using an explicit Euler discretization with time step $\Delta t = 10^{-7}$. Noise values $\varepsilon_i(t)$ were sampled from a uniform distribution over $[-\varepsilon, \varepsilon]$ according to a given noise level $\varepsilon > 0$.

First, consider the noiseless case $\varepsilon = 0$. Figure 6.2 shows the behaviour of the derivative estimates $\{\hat{z}_{i,1}(t)\}_{i=1}^{\mathsf{N}}$ for $\dot{\bar{z}}(t)$ as well as the error curves $\{|\hat{z}_{i,1}(t) - \dot{\bar{z}}(t)|\}_{i=1}^{\mathsf{N}}$. It can be noted convergence towards consensus is obtained in finite time and asymptotic convergence is obtained towards $\dot{\bar{z}}(t)$. Moreover, note that there is no steady state error. Now, consider the same experiment with noise level $\varepsilon = 0.1$. Figure 6.2 shows analogous results to the previous experiment in this case. Note that exact convergence is not obtained. Despite this, the protocol manages to estimate $\dot{\bar{z}}(t)$ with a maximum error of up to 0.4, even with noise. Similar results are obtained for the estimates $\{\hat{z}_{i,0}(t)\}_{i=1}^{\mathsf{N}}$ for $\bar{z}(t)$ which are not shown for the sake of brevity.

In order to quantify the effect to the noise in the protocol performance, the experiment was repeated with 100 noise levels $\varepsilon \in [0, 1]$. Figure 6.3 shows the value of the maximum consensus error after convergence for each experiment. Moreover, the curve $c_1\sqrt{\varepsilon}$ is shown for comparison according to item 1 of Theorem 6.3 where the value of $c_1 = 1.35$ was adjusted numerically afterwards. It is observed that the maximum error values grow approximately at the same rate as $c_1\sqrt{\varepsilon}$ as expected.

For comparison, the same experiment was repeated for a linear version of the protocol (6.3) with $F_\mu(\bullet) = (\bullet), \mu \in \{0, \ldots, m\}$. In order to make a fair comparison, the same gains as in the nonlinear experiment were used in this case. This strategy corresponds

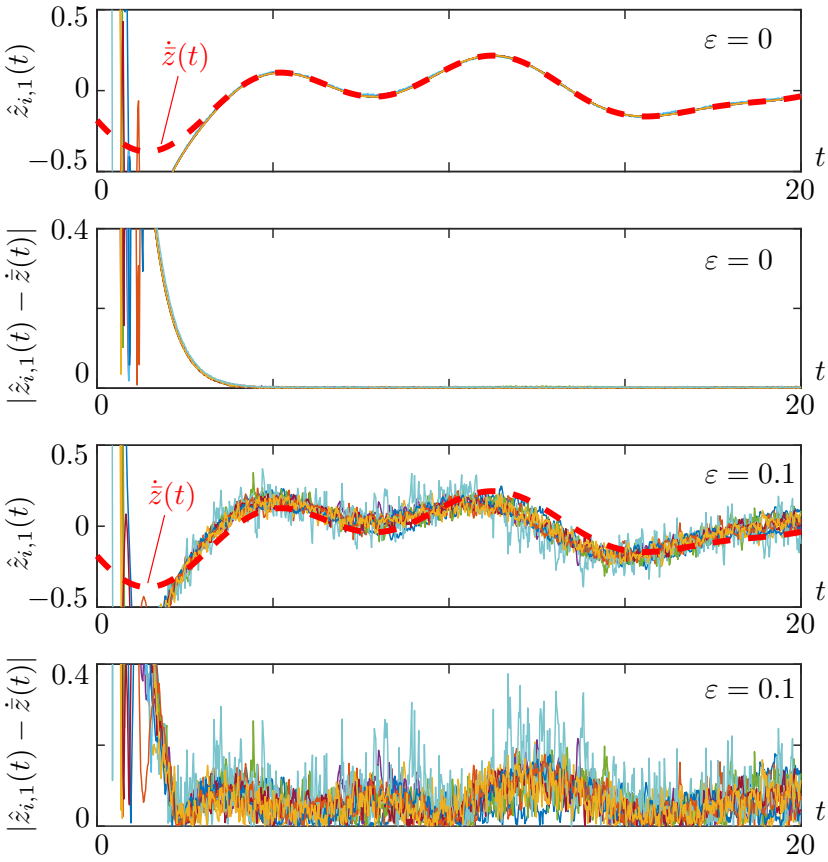

Figure 6.2: Behaviour of $\{\hat{z}_{i,1}(t)\}_{i=1}^{\mathsf{N}}$ and $\{|\hat{z}_{i,1}(t) - \dot{\bar{z}}(t)|\}_{i=1}^{\mathsf{N}}$ for the DDP protocol in the context of the experiment of Section 6.5 with no noise $\varepsilon = 0$ and with noise level of $\varepsilon = 0.1$. Without noise, the protocol manages to obtain exact convergence towards $\dot{\bar{z}}(t)$. With noise, the protocol manages to obtain an estimation of $\dot{\bar{z}}(t)$ with a maximum error of at most 0.4 after the transient.

to an adaptation of well established linear dynamic consensus protocols from [67] to the distributed differentiation problem presented in this chapter. As observed in Figure 6.3, even with $\varepsilon = 0$ the linear protocol cannot obtain exact convergence. Moreover, the differentiation error grows linearly with ε instead of with the optimal order $\sqrt{\varepsilon}$. An upper bound of the form $c_1'\varepsilon + c_2'$ with $c_1' = 4.01, c_2' = 0.25$ was obtained numerically afterwards as well.

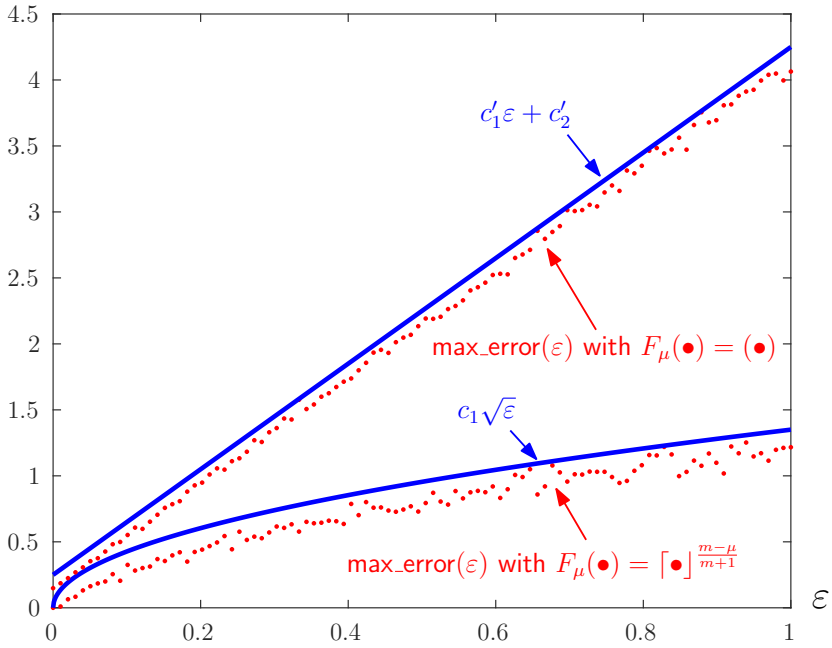

Figure 6.3: Maximum distributed differentiation error as a function of the noise level ε for (6.3) with both nonlinear and linear protocol versions. An upper bound of the form $c_1\sqrt{\varepsilon}$ is shown according to item 1 of Theorem 6.3 for the nonlinear version of (6.3). In addition, an upper bound of the form $c_1'\varepsilon + c_2'$ is shown for comparison for the linear version of (6.3).

6.6 DISCUSSION

This chapter proposed a new protocol to tackle the distributed differentiation problem. Unlike previous works, this protocol can achieve exact convergence to a weighted average signal in a decentralized fashion. Weighted averaging can be used to cope with different noise levels at each agent. For example, we discussed how if Gaussian measurement noise exists at each agent, the weights can be selected using a simple inverse-covariance rule to obtain an optimally fused measurement. Moreover, the structure of the new protocol does not require additional local differentiators. This is highly relevant in the context of perception latency, where the input signals are prone to noise, and no expression or measurement for their derivatives is generally available. Hence, not having to differentiate the inputs first, avoids additional noise amplification.

It was formally shown that the error bounds under noisy measurements are similar to the best possible differentiation accuracy found in single-agent robust exact differentiators. One of the most important takeaways of this chapter is that the advantages of the nonlinear EDC structure used in this thesis are made apparent when compared to standard linear consensus techniques. In particular, we show that the performance bounds for linear consensus grow significantly faster than our algorithms. This makes our EDC-based distributed differentiator a remarkably robust alternative for dynamic consensus

and state estimation problems.

In the subsequent chapters, we maintain a REDCHO-like structure similar to the one presented in Chapter 5 for the sake of simplicity. However, the analysis conducted in this chapter, which considers persistent measurement noise and additional weights to account for an inherent quality assigned to each node, can still be applied to the rest of the consensus algorithms presented in this thesis.

Chapter Seven

Perception Latency Aware Formation Control

In this chapter, we explore a scenario that involves cooperation between agents and perception latency scheduling to achieve formation control around a moving target. We will focus on developing distributed information fusion techniques to enhance target position estimation, effectively combining target detections across the robot network while considering perception latency. Moreover, we explore the requirements need in such target estimation for the formation control strategy. This problem allows to study the joint application of the ideas presented in previous chapters, revealing various challenges that need to be addressed for these algorithms to function effectively when used in conjunction.

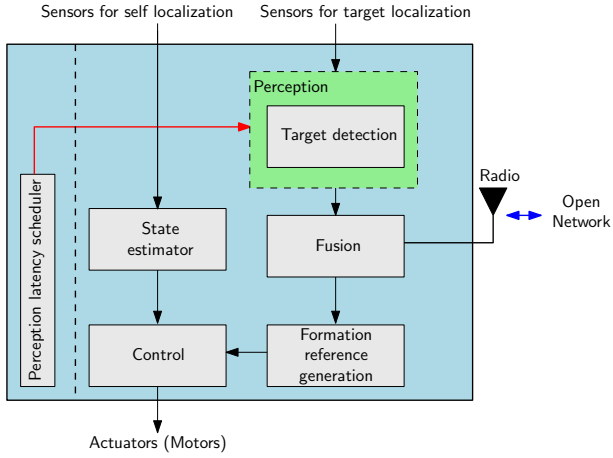

Figure 7.1: Model for the individual robot architecture with perception latency scheduler. Perception latency is only assumed for target detection for the sake of simplicity.

The architecture considered in this chapter is depicted in Figure 7.1. In this setup, we consider continuous-time communication and perception latency scheduling only for the target detection branch. We incorporate an arbitrary perception-latency scheduler, which dictates the quality and latency schedules for the perception stage output. With this setting, our objective is to propose a distributed information fusion module. The aim

of this module is to merge the multiple target detections across the network of robots, resulting in an enhanced target position estimation. This estimation is subsequently employed to construct references for the robot controller, enabling the formation of robots around the time-varying target position.

One of the most challenging aspects of this problem is dealing with perception latency, where only discrete-time measurements for the target position are available, possibly with substantial time gaps between them. Hence, when these measurements are used for trajectory generation in a formation control setting, this sparsity in conjunction with the continuous-time and possibly fast dynamics of the robots may be incompatible in the sense that trajectory tracking requires **smooth trajectories** compatible with the robot dynamics. For example, second integrator dynamics requires the trajectories to have a second order derivative almost everywhere, enabling precise trajectory tracking. In fact, the question on how to generate smooth trajectories using discrete-time irregularly spaced perception measurements is not trivial and if not done correctly may prevent asymptotic stability of the robot system as will be explored later in this chapter. The reason is that due to the hybrid nature of the problem, if standard techniques such as a Kalman filter are used, they may induce persistent transients in the robot behavior due to possible discontinuities in the filter's output. As a result, this can lead to reduced tracking performance.

The goal of this chapter is to propose EDC-based distributed information fusion techniques for combining target position estimations across the network and achieve accurate formation control around the target. Additionally, we introduce smooth estimation techniques that enable compatible formation trajectory generation for each robot. The contributions of this chapter were also published in [4].

7.1 RELATED WORK

Several strategies have been proposed in the literature in order to fuse local Kalman filter estimations either using a fusion center as in [87, 88] or in a decentralized fashion as is of interest in this chapter. In this context, in [89] local Kalman filter estimations are combined using discrete-time static consensus protocols, ignoring cross-correlations between agents. However, these only achieve exact convergence when an infinite number of iterations for the consensus protocol are performed at each sampling step. In contrast, the works [90, 91] use linear dynamic consensus filters [67] which do not have the previously mentioned issue. Despite this, as discussed in [1], linear dynamic consensus protocols cannot exhibit exact convergence with persistently varying references. Other recent fusion strategies have been proposed which improve the fusion quality by using the covariance intersection method [92–94]. The issue with all of the previously mentioned fusion methods is that they mainly work in discrete time. This means that synchronous measurements and updates might be assumed, which is incompatible with the perception-latency setting. On the other hand, the discontinuity issue in the estimations prevails in all these methods, since a continuous time-prediction must be computed in-between filter updates as well.

7.2 PROBLEM STATEMENT

Consider a team of N mobile robots located at positions $\mathbf{p}_i(t) \in \mathbb{R}^n, i \in \{1, \ldots, \mathsf{N}\}$ where n is the dimension of the workspace of the robots. For simplicity, each robot is modeled to be holonomic with m-th order integrator dynamics

$$\dot{\mathbf{x}}_i(t) = \mathbf{A}\mathbf{x}_i(t) + \mathbf{B}\mathbf{u}_i(t), \tag{7.1}$$

where $\mathbf{u}_i(t) \in \mathbb{R}^n$ is a local control input,

$$\mathbf{x}_i(t) = \begin{bmatrix} \mathbf{p}_i^{(0)}(t) \\ \vdots \\ \mathbf{p}_i^{(m-1)}(t) \end{bmatrix},$$

and we consider the following matrix definitions

$$\mathbf{A} = \mathbf{A}_0 \otimes \mathbf{I}_n, \quad \mathbf{B} = \mathbf{B}_0 \otimes \mathbf{I}_n, \quad \mathbf{C} = \mathbf{C}_0 \otimes \mathbf{I}_n$$
$$\mathbf{A}_0 = \begin{bmatrix} \mathbf{0}_m & \mathbf{I}_m \\ 0 & \mathbf{0}_m^\top \end{bmatrix}, \mathbf{B}_0 = [\mathbf{0}_m, 1]^\top, \mathbf{C}_0 = [1, \mathbf{0}_m^\top]$$

We assume that each robot is able to measure its own state $\mathbf{x}_i(t)$ using local sensors, or through a state estimator as in Figure 7.1, which we assume to provide the state directly. The robots are able to share information between them according to a communication network modeled by an undirected graph $\mathcal{G}$. In addition, a target of interest with position $\mathbf{p}(t) \in \mathbb{R}^n$ is assumed to have dynamics with similar integrator dynamics as in (7.1). Moreover, since the input at the target is unknown, we model it in a stochastic fashion as

$$\mathrm{d}\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t)\mathrm{d}t + \mathbf{B}\mathrm{d}\mathbf{u}(t) \tag{7.2}$$

where $\mathbf{p}(t) = \mathbf{C}\mathbf{x}(t)$ and $\mathbf{u}(t) \in \mathbb{R}^n$ is a n-dimensional Wiener processes with covariance $\mathsf{cov}\{\mathbf{u}(s), \mathbf{u}(r)\} = \mathbf{W}\min(s, r)$ [119, Page 63]. As usual, the process $\mathbf{u}(t)$ models disturbances, unknown inputs at the target, and non-modeled dynamics.

Perception mechanism: Consider a similar percetion model as in Chapters 2 and 3 where the robot i uses available sensors such as vision, range, etc, to produce *raw measurements* of the environment. For each raw measurement, a *processed measurement* for the position of the target is obtained through a detection process at, perhaps non-uniform, processing instants $\boldsymbol{\tau}_i = \{\tau_{i,k}\}_{k=0}^{\infty}$, $\tau_{i,0} = 0$. Processed measurements are detections of $\mathbf{p}(\tau_{i,k}) = \mathbf{C}\mathbf{x}(\tau_{i,k})$. According to the current processing and energy budget of robot i, a perception latency $\Delta_{i,k} \in [\Delta_{\min}, \Delta_{\max}]$ with $\Delta_{\min}, \Delta_{\max} > 0$ is chosen for the detection process at $t = \tau_{i,k}$ and the processed measurement $\mathbf{z}_i(\tau_{i,k}) = \mathbf{C}\mathbf{x}(\tau_{i,k}) + \mathbf{v}_i(\tau_{i,k})$ is available at $t = \tau_{i,k} + \Delta_{i,k}$. Here, $\mathbf{v}_i(\tau_{i,k})$ is a Gaussian noise modeling the accuracy of the perception method. In general, $\mathbf{R}(\Delta_{i,k}) = \mathsf{cov}\{\mathbf{v}_i(\tau_{i,k})\}$ decreases in magnitude as more processing time $\Delta_{i,k}$ is employed.

Remark 7.1. *In order to focus in the estimation and information fusion aspects of this chapter, we consider that the process for which the perception latencies $\{\Delta_{i,k}\}_{k=0}^{\infty}$ are chosen at robot i is already given. The schedulers from Chapters 2 and 3 can be applied where only D available perception methods can be used, and the perception-latency schedule $\Delta_{i,k} \in \{\Delta^j\}_{j=1}^{D}$ is chosen to minimize a local performance function.*

The goal is for the robots to achieve a given formation provided known displacements $\mathbf{d}_1, \ldots, \mathbf{d}_{\mathsf{N}} \in \mathbb{R}^n$ around the target. However, given that target localization is imperfect, an approximate scheme must be adopted. The strategy is to obtain local estimates $\hat{\mathbf{x}}_i(t)$ of the target state at each robot given the local perception-latency schedule $\{\Delta_{i,k}\}_{k=0}^{\infty}$, and then combine them into a single global estimate $\bar{\mathbf{p}}_{\mathsf{G}}(t)$ for the trajectory $\mathbf{p}(t)$ in a distributed and decentralized fashion. Then, achieve a formation around $\bar{\mathbf{p}}_{\mathsf{G}}(t)$ instead. This is, given a reference $\mathbf{p}_i^{\mathsf{r}}(t) = \bar{\mathbf{p}}_{\mathsf{G}}(t) + \mathbf{d}_i$, reach

$$\textbf{Goal: } \lim_{t\to\infty} \|\mathbf{p}_i(t) - \mathbf{p}_i^{\mathsf{r}}(t)\| = 0, \forall i \in \{1, \ldots, \mathsf{N}\} \tag{7.3}$$

The reference should be smooth enough for $(\mathbf{p}_i^{\mathsf{r}})^{(m)}(t), \forall t \geq 0$ to exist and achieve trajectory tracking at each robot. Nonetheless, this is incompatible with existing optimal estimation techniques taking into account the hybrid nature of the problem.

7.2.1 Solution outline

To solve the problem, the proposal contains the following ingredients:

- **Smooth-output estimation**: We propose an alternative to classical filtering obtaining a smooth trajectory of the estimate $\hat{\mathbf{x}}_i(t)$ of $\mathbf{x}(t)$ locally at robot i.
- **Estimation fusion**: A distributed and decentralized consensus filter is used to fuse the local information of $\hat{\mathbf{x}}_i(t)$ into the single estimate $\bar{\mathbf{p}}_{\mathsf{G}}(t)$ of $\mathbf{p}(t)$.
- **Formation control:** A local controller $\mathbf{u}_i(t)$ is designed for the robot i to achieve the formation goal in (7.3) using the estimation framework.

Figure 7.2 presents a high-level outline of the processing flow for our proposal. In particular, we assume that the target detection process is already given, taking raw measurements of the environment and producing detections for the target as $\mathbf{z}_i(\tau_{i,k})$. The Smooth-Output Estimation (SOE) block is described in detail in Section 7.3, which takes the output of the detection process, and computes smooth information vector and matrix $\hat{\mathbf{y}}_i(t), \hat{\mathbf{Q}}_i(t)$ associated with the estimation $\hat{\mathbf{x}}_i(t)$ and its covariance $\hat{\mathbf{P}}_i(t)$ for the state of the target. The fusion algorithm is described in detail in Section 7.4, and is composed of two modules. First, a consensus stage computes a fused version of the information vector and matrix $\mathbf{y}_{i,0}(t), \mathbf{Q}_{i,0}(t)$ for the target state as well as its derivatives $\{\mathbf{y}_{i,\mu}(t), \mathbf{Q}_{i,\mu}(t)\}_{\mu=1}^{m}$. Then, the fusion output stage uses them to compute the actual joint estimation $\bar{\mathbf{p}}_{\mathsf{G}}(t)$ for the target position $\mathbf{p}(t)$ and its first m derivatives. These estimations are stored in the local variables $\{\mathbf{p}_{i,\mu}(t)\}_{\mu=0}^{m}$. Finally, these signals are used to compute a local trajectory tracking control for each robot, which is described in Section 7.5.

The main advantage of the architecture depicted in Figure 7.2, is that the smooth estimation block allows the fusion and control blocks to be designed independently from the detection procedure, providing more versatility than in prior literature. In addition, the smooth output estimation and fusion greatly reduces the estimation and tracking errors as shown in the experiments provided in Section 7.6.

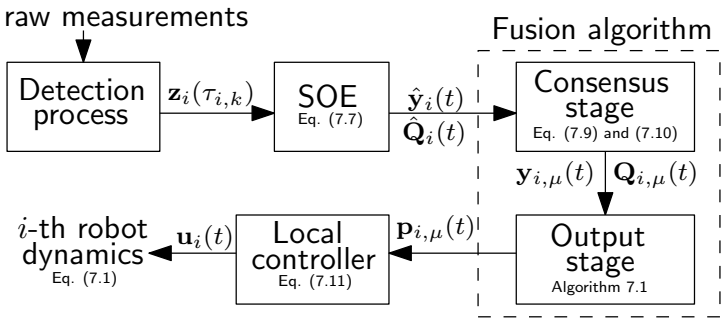

Figure 7.2: High-level of the processing flow for our proposal as described in Section 7.2.1.

7.3 SMOOTH-OUTPUT ESTIMATION

In this section, we focus on the local target estimation at each robot. To simplify the presentation, we drop the index i when there is no ambiguity given that all variables are assumed to be local. First, given a sequence of local sampling instants $\boldsymbol{\tau} = \{\tau_k\}_{k=0}^{\infty}$, adopt the notation $\mathbf{x}[k] := \mathbf{x}(\tau_k)$ and note that a sampled-data version of the target dynamics (7.2) for $\mathbf{x}(t)$ is obtained similarly as in [130, Section 4.5.2]:

$$\mathbf{x}(t) = \mathbf{A}_{\mathsf{d}}(t-\tau_k)\mathbf{x}[k] + \mathbf{u}_{\mathsf{d}}(t), \quad t \in [\tau_k, \tau_{k+1}] \tag{7.4}$$

where $\mathbf{u}_{\mathsf{d}}(t)$ is a normal random variable of zero mean and $\mathsf{cov}\{\mathbf{u}_{\mathsf{d}}(t)\} = \mathbf{W}_{\mathsf{d}}(t-\tau_k) = \int_0^t \mathbf{A}_{\mathsf{d}}(\tau)\mathbf{B}\mathbf{W}\mathbf{B}^\top \mathbf{A}_{\mathsf{d}}(\tau)^\top \mathrm{d}\tau$ with $\mathbf{A}_{\mathsf{d}}(\tau) = \exp(\mathbf{A}\tau)$.

Every robot can use its observations of the target $\mathbf{z}[0], \ldots, \mathbf{z}[k-1]$ and compute a causal optimal filter for (7.4) at the sampling instants $t \in \boldsymbol{\tau}$ of the form $\hat{\mathbf{x}}^*[k] = \mathbb{E}\{\mathbf{x}[k]|\mathbf{z}[0], \ldots, \mathbf{z}[k-1]\}$ which does not take into account $\mathbf{z}[k]$ due to the perception latency. By applying [119, Page 228 - Theorem 4.1] to system (7.4) evaluated at $t = \tau_{k+1}$, the recursive filter structure to obtain $\hat{\mathbf{x}}^*[k]$ and $\hat{\mathbf{P}}^*[k] = \mathsf{cov}\{\mathbf{x}[k] - \hat{\mathbf{x}}^*[k]\}$ has the following form:

$$\begin{aligned}
\mathbf{L}[k] &= \mathbf{A}_{\mathsf{d}}(\Delta_k)\hat{\mathbf{P}}^*[k]\mathbf{C}^\top \left(\mathbf{C}\hat{\mathbf{P}}^*[k]\mathbf{C}^\top + \mathbf{R}(\Delta_k)\right)^{-1} \\
\hat{\mathbf{x}}^*[k+1] &= \mathbf{A}_{\mathsf{d}}(\Delta_k)\hat{\mathbf{x}}^*[k] + \mathbf{L}[k]\left(\mathbf{z}[k] - \mathbf{C}\hat{\mathbf{x}}^*[k]\right) \\
\hat{\mathbf{P}}^*[k+1] &= \left(\mathbf{A}_{\mathsf{d}}(\Delta_k) - \mathbf{L}[k]\mathbf{C}\right)\hat{\mathbf{P}}^*[k]\left(\mathbf{A}_{\mathsf{d}}(\Delta_k) - \mathbf{L}[k]\mathbf{C}\right)^\top + \mathbf{L}[k]\mathbf{R}(\Delta_k)\mathbf{L}[k]^\top + \mathbf{W}_{\mathsf{d}}(\Delta_k)
\end{aligned} \tag{7.5}$$

assuming knowledge of some initial values for $\hat{\mathbf{x}}^*[0], \hat{\mathbf{P}}^*[0]$. On the other hand, an optimal estimation $\hat{\mathbf{x}}^*(t|k) := \mathbb{E}\{\mathbf{x}(t)|\mathbf{z}[0], \ldots, \mathbf{z}[k-1]\}$ for $t \in (\tau_k, \tau_{k+1})$ can be obtained by using a model based prediction of (7.4) as:

$$\begin{aligned}
\hat{\mathbf{x}}^*(t|k) &= \mathbf{A}_{\mathsf{d}}(t-\tau_k)\hat{\mathbf{x}}^*[k] \\
\hat{\mathbf{P}}^*(t|k) &= \mathsf{cov}\{\mathbf{x}(t) - \hat{\mathbf{x}}^*(t|k)\} \\
&= \mathbf{A}_{\mathsf{d}}(t-\tau_k)\hat{\mathbf{P}}^*[k]\mathbf{A}_{\mathsf{d}}(t-\tau_k)^T + \mathbf{W}_{\mathsf{d}}(t-\tau_k)
\end{aligned} \tag{7.6}$$

Note that the expressions in (7.6) can be defined for all $t \geq 0$. However, we use $\hat{\mathbf{x}}^*(t)$ to refer to the whole optimal causal estimated trajectory constructed piecewise as $\hat{\mathbf{x}}^*(t) :=$

$\hat{\mathbf{x}}^*(t|k)$ and $\hat{\mathbf{P}}^*(t) := \hat{\mathbf{P}}^*(t|k)$ for $t \in [\tau_k, \tau_{k+1})$. Note also that (7.5) is a Kalman filter [119, Chapter 7]. Hence, the standard discrete-time Kalman filter coincides with the DOE at $t = \tau_k$.

Remark 7.2. *Note that $\hat{\mathbf{x}}^*(t)$ is a smooth function of time for all $t \notin \boldsymbol{\tau}$. However, $\hat{\mathbf{x}}^*(t)$ may be discontinuous at $t \in \boldsymbol{\tau}$ due to the measurement corrections performed in* (7.5)*. Hence we refer to $\hat{\mathbf{x}}^*(t)$ as a Discontinuous Optimal Estimation (DOE). If $\hat{\mathbf{x}}^*(t)$ is used to construct a reference for the robot, such discontinuities can cause persistent transients in the closed loop behaviour of the robot compromising asymptotic stability. An example of this is shown in Section 7.6.1.*

We propose a near-optimal alternative estimation $\hat{\mathbf{x}}(t)$ which does not have the discontinuity issue at $t \in \boldsymbol{\tau}$. The idea is to combine the current optimal estimate $\hat{\mathbf{x}}^*(t|k)$ for $t \in [\tau_k, \tau_{k+1})$ with the continued prediction from the previous estimate $\hat{\mathbf{x}}^*(t|k-1)$ for $t \geq \tau_k$, using a smooth transition between both.

In order to perform such transition, consider the following auxiliary function:

Definition 7.3 (Transition function)**.** *$\eta(\bullet;\alpha) : [0,1] \to \mathbb{R}$ is a transition function if it complies that $\forall \mu \in \{1,\dots,m\}$:*

1. *$\eta^{(\mu)}(t;\alpha)$ exists $\forall t \in [0,1]$.*
2. *$\eta^{(\mu)}(0;\alpha) = \eta^{(\mu)}(1;\alpha) = 0$.*
3. *$\eta(0;\alpha) = 0$, $\eta(1;\alpha) = 1$.*
4. *$\lim_{\alpha\to 0} \eta(t,\alpha) = 1, \forall t \in (0,1]$*

An example of a transition function is provided in the following which can be assumed to be fixed as such through the rest of the manuscript.

Proposition 7.4. *Let $\alpha > 0$ and $\eta(\tau;\alpha) = \dfrac{\Phi(\tau)}{\Phi(\tau) + \Phi(\alpha(1-\tau))}$ with $\Phi(\tau) = \tau^{m+1}$. Then, $\eta(\tau;\alpha)$ satisfy Definition 7.3.*

Proof: Item 1 follows from smoothness of $\Phi(\tau)$ leading to smoothness of $\eta(\tau,\alpha)$ for any $\tau \in [0,1]$. Note $\Phi^{(\mu)}(0) = 0, \forall \mu \in \{1,\dots,m\}$ and hence $\eta^{(\mu)}(\tau,\alpha) = 0$ as well from the product rule. Moreover, $\eta(\tau;\alpha) = 1 - \eta(\alpha(1-\tau);\alpha^{-1})$ so that $\eta^{(\mu)}(1;\alpha) = -\eta^{(\mu)}(0;\alpha) = 0, \forall \mu \in \{1,\dots,m\}$ showing item 2. Item 3 of Definition 7.3 is complied by evaluating $\eta(\tau;\alpha) = 0, \eta(\tau;\alpha) = 1$. Finally, note that for fixed $\tau \in (0,1)$, it follows that $\lim_{\alpha\to 0} \Phi(\alpha(1-\tau)) = 0$ point-wise, implying item 4. □

Hence, given $\alpha > 0$, a transition function, and by defining the information matrix $\hat{\mathbf{Q}}^*(t|k) = \hat{\mathbf{P}}^*(t|k)^{-1}$ and information vector $\hat{\mathbf{y}}^*(t|k) = \hat{\mathbf{Q}}^*(t|k)\hat{\mathbf{x}}^*(t|k)$, let the Smooth Output Estimation (SOE):

$$
\begin{aligned}
\lambda_1(t|k) &= 1 - \eta((t-\tau_k)/\Delta_k;\alpha) \\
\lambda_2(t|k) &= \eta((t-\tau_k)/\Delta_k;\alpha) \\
\hat{\mathbf{Q}}(t) &= \lambda_1(t|k)\hat{\mathbf{Q}}^*(t|k-1) + \lambda_2(t|k)\hat{\mathbf{Q}}^*(t|k) \\
\hat{\mathbf{y}}(t) &= \lambda_1(t|k)\hat{\mathbf{y}}^*(t|k-1) + \lambda_2(t|k)\hat{\mathbf{y}}^*(t|k) \\
\hat{\mathbf{P}}(t) &= \hat{\mathbf{Q}}(t)^{-1} \\
\hat{\mathbf{x}}(t) &= \hat{\mathbf{Q}}(t)^{-1}\hat{\mathbf{y}}(t)
\end{aligned}
\tag{7.7}
$$

for $t \in [\tau_k, \tau_{k+1})$. Similarly as with $\hat{\mathbf{x}}^*(t)$, we drop the dependence on k in the notation for the SOE estimation $\hat{\mathbf{x}}(t)$ in order to refer to the whole sub-optimal trajectory. Now, we establish some important properties of (7.7).

Theorem 7.5. *Given $\alpha > 0$, the SOE in (7.7) complies with the following:*

1. **(Smoothness)** *The information vector and matrix $\hat{\mathbf{y}}(t), \hat{\mathbf{Q}}(t)$ are m-times differentiable $\forall t \geq 0$. As a consequence, $\hat{\mathbf{x}}(t)$ is m-times differentiable $\forall t \geq 0$.*
2. **(Unbiasedness)** $\mathbb{E}\{\mathbf{x}(t) - \hat{\mathbf{x}}(t)\} = 0, \forall t \geq 0$.
3. **(Tight consistency)** $\hat{\mathbf{P}}^*(t) \preceq \mathit{cov}\{\mathbf{x}(t) - \hat{\mathbf{x}}(t)\} \preceq \hat{\mathbf{P}}(t), \forall t \geq 0$ *with equality for all $t \notin \boldsymbol{\tau}$ as $\alpha \to 0$.*

PROOF: For the sake of readability, the proof can be found at the end of this chapter in Section 7.8.1. □

Remark 7.6. *Different from a discrete-time Kalman filter or the DOE, the SOE is ensured to have a sufficiently smooth output for any $t \geq 0$, even under large perception latency. There are two basic smooth-output alternatives to the SOE proposed in (7.7): interpolation techniques and low-pass filtering. In the case of interpolation, an m-th order spline may be used in a moving horizon fashion to fill in the spaces between samples $\{\hat{\mathbf{y}}^*[k], \hat{\mathbf{Q}}^*[k]\}_{k=0}^{\infty}$ for all $t \notin \boldsymbol{\tau}$. On the other hand, an m-th order low-pass filter can be used on top of (7.6) to obtain smooth versions of $\hat{\mathbf{y}}^*(t), \hat{\mathbf{Q}}^*(t)$ as well. Therefore, an analogous to property **i**) of Theorem 7.5 is obtained for both of these alternatives. However, neither unbiasedness nor consistency can be ensured for such types of estimations. In addition, contrary to the alternatives, the tight near-optimal quality of the estimation of our proposal can be adjusted appropriately by modifying the parameter $\alpha > 0$ as shown in item **iii**) of Theorem 7.5.*

Remark 7.7. *In practice, Visual Target Detectors (VTDs) such as the ones in [49, 51, 52] are prone to data association problems when false positives or miss-detections occur, and under false negatives and occlusions where the procedure cannot detect the target. However, a VTD in conjunction with target tracking using state-based predictions provide a more robust alternative since predicted targets can be matched with detections. For instance, [131] provides some examples in which state-based predictions are used to improve the accuracy of the target detector. In this context, the SOE can be used as a state-based prediction by evaluating (7.7) at any time until a new sample is available, which can then be used for data association.*

7.4 ESTIMATION FUSION

Using the SOE in (7.7), each robot i is able to compute a near-optimal local estimation $\hat{\mathbf{x}}_i(t)$ and a consistent covariance matrix $\hat{\mathbf{P}}_i(t)$ with their corresponding smooth information vector and matrix $\hat{\mathbf{y}}_i(t), \hat{\mathbf{Q}}_i(t)$ as a result of Theorem 7.5. Hence, we turn our attention to the task of fusing this information across all robots in the communication network. Note that even when processed measurements are uncorrelated, the estimations $\hat{\mathbf{x}}_i(t)$ are correlated since both depend on the same noise process $\mathbf{u}(t)$ driving system (7.2). However, keeping track of the cross-correlations between estimates at different robots is not scalable with respect to the network size. Therefore, we ignore cross-correlations and fuse estimation information according to:

$$\begin{aligned} \bar{\mathbf{Q}}_{\mathsf{G}}(t) &= \frac{1}{\mathsf{N}} \sum_{i=1}^{\mathsf{N}} \hat{\mathbf{Q}}_i(t), \qquad \bar{\mathbf{y}}_{\mathsf{G}}(t) = \frac{1}{\mathsf{N}} \sum_{i=1}^{\mathsf{N}} \hat{\mathbf{y}}_i(t) \\ \bar{\mathbf{x}}_{\mathsf{G}}(t) &= \bar{\mathbf{Q}}_{\mathsf{G}}(t)^{-1} \bar{\mathbf{y}}_{\mathsf{G}}(t), \qquad \bar{\mathbf{p}}_{\mathsf{G}}(t) = \mathbf{C} \bar{\mathbf{x}}_{\mathsf{G}}(t) \end{aligned} \tag{7.8}$$

Note that the matrix $\bar{\mathbf{P}}_{\mathsf{G}}(t) = \bar{\mathbf{Q}}_{\mathsf{G}}(t)^{-1}$ is consistent in the sense that $\mathsf{cov}\{\mathbf{x}(t) - \bar{\mathbf{x}}_{\mathsf{G}}(t)\} \preceq \bar{\mathbf{P}}_{\mathsf{G}}(t)$ since $\bar{\mathbf{Q}}_{\mathsf{G}}(t)$ is a convex combination of the information matrices $\{\hat{\mathbf{Q}}_i(t)\}_{i=1}^{\mathsf{N}}$ and by the covariance-intersection principle from [164, Section 2.1]. The same idea of computing a consistent inverse-covariance fusion estimate as in (7.8), ignoring cross-correlations, has shown to be very successfully in different state estimation contexts such as [165].

If every robot had access to the global quantity $\bar{\mathbf{p}}_{\mathsf{G}}(t)$ from (7.8), hence a local trajectory $\mathbf{p}_i^{\mathsf{r}}(t) = \bar{\mathbf{p}}_{\mathsf{G}}(t) + \mathbf{d}_i$ can be constructed for each agent to follow. However, in order to construct a controller $\mathbf{u}_i(t)$ to achieve trajectory tracking of (7.1) towards $\mathbf{p}_i^{\mathsf{r}}(t)$, knowledge of the derivatives $(\mathbf{p}_i^{\mathsf{r}})^{(\mu)}(t) = \bar{\mathbf{p}}_{\mathsf{G}}^{(\mu)}(t), \mu \in \{1, \dots, m\}$ is required as we discuss in Section 7.5. While the expressions in (7.8) constitute a transformation between information and state representations, the following result provides an equivalent transformation between the derivatives of $\bar{\mathbf{p}}_{\mathsf{G}}(t)$ and the ones of $\bar{\mathbf{Q}}_{\mathsf{G}}(t), \bar{\mathbf{y}}_{\mathsf{G}}(t)$.

Lemma 7.8. *Let $\bar{\mathbf{p}}_{\mathsf{G}}(t).\bar{\mathbf{Q}}_{\mathsf{G}}(t), \bar{\mathbf{y}}_{\mathsf{G}}(t)$ defined in (7.8) and $\bar{\mathbf{P}}_{\mathsf{G}}(t) = \bar{\mathbf{Q}}_{\mathsf{G}}(t)^{-1}$. Then, for any $\mu \in \{1, \dots, m\}$:*

1. $\bar{\mathbf{P}}_{\mathsf{G}}^{(\mu)}(t) = -\bar{\mathbf{P}}_{\mathsf{G}}(t) \sum_{\nu=0}^{\mu-1} \binom{\mu}{\nu} \bar{\mathbf{Q}}_{\mathsf{G}}^{(\mu-\nu)}(t) \bar{\mathbf{P}}_{\mathsf{G}}^{(\nu)}(t)$.
2. $\bar{\mathbf{p}}_{\mathsf{G}}^{(\mu)}(t) = \mathbf{C} \sum_{\nu=0}^{\mu} \binom{\mu}{\nu} \bar{\mathbf{P}}_{\mathsf{G}}^{(\nu)}(t) \bar{\mathbf{y}}_{\mathsf{G}}^{(\mu-\nu)}(t)$.

PROOF: For the sake of readability, the proof can be found at the end of this chapter in Section 7.8.2. □

Computing (7.8) as well as the derivatives in Lemma 7.8-2 in a decentralized fashion under time-varying $\hat{\mathbf{y}}_i(t), \hat{\mathbf{Q}}_i(t)$ is not trivial. In the following we provide an algorithm to compute such quantities asymptotically without steady state error using exact dynamic consensus tools, even under persistently varying $\hat{\mathbf{y}}_i(t), \hat{\mathbf{Q}}_i(t)$. The fusion algorithm is composed by two sequential stages. First, we use a *consensus stage* where information fusion is performed to compute $\bar{\mathbf{y}}_{\mathsf{G}}(t), \bar{\mathbf{Q}}_{\mathsf{G}}(t)$ and its first m derivatives using local information $\{\hat{\mathbf{y}}_i(t), \hat{\mathbf{Q}}_i(t)\}_{i=1}^{\mathsf{N}}$ in a decentralized fashion. Second, we use an *output stage* where the outputs of the consensus stage are organized in order to compute $\bar{\mathbf{p}}_{\mathsf{G}}(t)$ and its first m derivatives based on the transformations given in (7.8) and Lemma 7.8.

Based on the REDCHO algorithm from Chapter 5 with parameters $\{k_\mu, \gamma_\mu\}_{\mu=0}^m$ and θ, the consensus stage is composed by consensus protocols for the information vector

$$\begin{aligned}
\mathbf{y}_{i,\mu}(t) &= \hat{\mathbf{y}}_i^{(\mu)}(t) - \sum_{\nu=0}^{m} \mathbf{G}_{\mu+1,\nu+1}\mathbf{v}_{i,\nu}(t) \\
\dot{\mathbf{v}}_{i,\mu}(t) &= k_\mu \theta^{\mu+1} \sum_{j\in\mathcal{N}_i} \lceil \mathbf{y}_{i,0}(t) - \mathbf{y}_{j,0}(t) \rfloor^{\frac{m-\mu}{m+1}} + \ \mathbf{v}_{i,\mu+1}(t) - \gamma_\mu \mathbf{v}_{i,\mu}(t), \\
&\quad \text{for } 0 \leq \mu < m \\
\dot{\mathbf{v}}_{i,m}(t) &= k_m \theta^{m+1} \sum_{j\in\mathcal{N}_i} \lceil \mathbf{y}_{i,0}(t) - \mathbf{y}_{j,0}(t) \rfloor^{0} - \gamma_m \mathbf{v}_{i,m}(t)
\end{aligned} \tag{7.9}$$

and for the information matrix

$$\begin{aligned}
\mathbf{Q}_{i,\mu}(t) &= \hat{\mathbf{Q}}_i^{(\mu)}(t) - \sum_{\nu=0}^{m} \mathbf{G}_{\mu+1,\nu+1}\mathbf{V}_{i,\nu}(t) \\
\dot{\mathbf{V}}_{i,\mu}(t) &= k_\mu \theta^{\mu+1} \sum_{j\in\mathcal{N}_i} \lceil \mathbf{Q}_{i,0}(t) - \mathbf{Q}_{j,0}(t) \rfloor^{\frac{m-\mu}{m+1}} + \ \mathbf{V}_{i,\mu+1}(t) - \gamma_\mu \mathbf{V}_{i,\mu}(t), \\
&\quad \text{for } 0 \leq \mu < m \\
\dot{\mathbf{V}}_{i,m}(t) &= k_m \theta^{m+1} \sum_{j\in\mathcal{N}_i} \lceil \mathbf{Q}_{i,0}(t) - \mathbf{Q}_{j,0}(t) \rfloor^{0} - \gamma_m \mathbf{V}_{i,m}(t)
\end{aligned} \tag{7.10}$$

where a_{ij} are the components of the adjacency matrix of $\mathcal{G}$, $\mathbf{G}_{\mu+1,\nu+1}$ with $\mu, \nu \in \{0, \ldots, m\}$ are the components of $\mathbf{G}$ defined in (5.1).

The protocols in (7.9) and (7.10) take as an input the local information $\hat{\mathbf{y}}_i(t), \hat{\mathbf{Q}}_i(t)$, and have outputs

$$\{\mathbf{y}_{i,\mu}(t), \mathbf{Q}_{i,\mu}(t)\}_{\mu=0}^m$$

computed through the internal variables

$$\{\mathbf{v}_{i,\mu}(t), \mathbf{V}_{i,\mu}(t)\}_{\mu=0}^m.$$

The purpose of (7.9) is that $\mathbf{y}_{i,0}(t), \ldots, \mathbf{y}_{i,m}(t)$ will converge towards $\bar{\mathbf{y}}_{\mathsf{G}}(t)$ and its first m derivatives for each robot $i \in \{1, \ldots, \mathsf{N}\}$. Similarly, $\mathbf{Q}_{i,0}(t), \ldots, \mathbf{Q}_{i,m}(t)$ will converge towards $\bar{\mathbf{Q}}_{\mathsf{G}}(t)$ and its first m derivatives. In addition, the structure of the protocol allows each agent to communicate only $\mathbf{y}_{i,0}(t)$ and $\mathbf{Q}_{i,0}(t)$ to its neighbors. It is clear that each robot requires to compute $\{\hat{\mathbf{y}}_i^{(\mu)}(t), \hat{\mathbf{Q}}_i^{(\mu)}(t)\}_{\mu=0}^m$ locally as observed in equations for $\mathbf{y}_{i,\mu}(t), \mathbf{Q}_{i,\mu}(t)$ in (7.9) and (7.10). These can be computed explicitly from (7.7) since the expression for the transition function $\eta(\bullet; \alpha)$ and the model based predictions in (7.6) are known explicitly as well.

Moreover, note that each component of the equations in (7.9) and (7.10) is an independent instance of the REDCHO protocol in (5.2). As a result, (7.9) and (7.10) can be alternatively implemented using $nm + nm(nm+1)/2$ REDCHO instances, one for each non-repeated component of $\mathbf{y}_{i,m}(t) \in \mathbb{R}^{nm}, \mathbf{Q}_{i,m}(t) \in \mathbb{R}^{nm\times nm}$ provided that $\mathbf{Q}_{i,m}(t)$ is symmetric.

In the output stage, $m+1$ outputs are obtained from $\{\mathbf{y}_{i,\mu}(t), \mathbf{Q}_{i,\mu}(t)\}_{\mu=0}^m$ as given in Algorithm 7.1 which is based on the transformation of Lemma 7.8. In fact, the structure in Algorithm 7.1 is equivalent to the one in Lemma 7.8 assuming that $\{\mathbf{y}_{i,\mu}(t), \mathbf{Q}_{i,\mu}(t)\}_{\mu=0}^m$ have already converged towards $\bar{\mathbf{y}}_{\mathsf{G}}(t), \bar{\mathbf{Q}}_{\mathsf{G}}(t)$ and their first m derivatives. Hence, in order to ensure convergence we adopt the following assumption.

Algorithm 7.1 Estimation fusion output stage

Input: $\{\mathbf{y}_{i,\mu}(t), \mathbf{Q}_{i,\mu}(t)\}_{\mu=0}^{m}$ `computed from` (7.9) `and` (7.10)
Output: $\{\mathbf{p}_{i,\mu}(t)\}_{\mu=0}^{m}$
1: $\mathbf{P}_{i,0}(t) \leftarrow \mathbf{Q}_{i,0}(t)^{-1}$
2: $\mathbf{p}_{i,0}(t) \leftarrow \mathbf{C}\mathbf{P}_{i,0}(t)\mathbf{y}_{i,0}(t)$
3: **for** $\mu \in \{1, \ldots, m\}$ **do**
4: $\quad \mathbf{P}_{i,\mu}(t) \leftarrow -\mathbf{P}_{i,0}(t) \sum_{\nu=0}^{\mu-1} \binom{\mu}{\nu} \mathbf{Q}_{i,\mu-\nu}(t)\mathbf{P}_{i,\nu}(t)$
5: $\quad \mathbf{p}_{i,\mu}(t) \leftarrow \mathbf{C} \sum_{\nu=0}^{\mu} \binom{\mu}{\nu} \mathbf{P}_{i,\nu}(t)\mathbf{y}_{i,\mu-\nu}(t)$
6: **end for**

Assumption 7.9. *Let gains $\gamma_0, \ldots, \gamma_m > 0$ and $z_i(t)$ be an arbitrary component of $\hat{\mathbf{y}}_i(t)$ or $\hat{\mathbf{Q}}_i(t)$. Hence, $\{z_i(t)\}_{i=1}^{\mathsf{N}}$ satisfy Assumption 5.2 in Chapter 5 for some $L > 0$.*

Remark 7.10. *Due to item **i)** of Theorem 7.5, Assumption 7.9 is complied for t on any compact interval and sufficiently big $L > 0$. However, this assumption can be complied for any $t \geq 0$ as well by assuming that the motion of the target has bounded derivatives, which is mild in practical scenarios.*

Theorem 7.11. *Let Assumption 7.9 hold and $\mathcal{G}$ be a connected network. Moreover, let the initial conditions and gains for each REDCHO instance configured to satisfy the conditions of Theorem 5.10 in Chapter 5. Then, there exists an m-times differentiable consensus signal $\tilde{\mathbf{p}}(t) \in \mathbb{R}^n$ and $T > 0$ such that the outputs of Algorithm 7.1 satisfy that for all $\mu \in \{0, \ldots, m\}$:*

$$\tilde{\mathbf{p}}^{(\mu)}(t) = \mathbf{p}_{1,\mu}(t) = \cdots = \mathbf{p}_{\mathsf{N},\mu}(t), \forall t \geq T$$

In addition, $\lim_{t\to\infty} \left| \tilde{\mathbf{p}}^{(\mu)}(t) - \bar{\mathbf{p}}_{\mathsf{G}}^{(\mu)}(t) \right| = 0$

Proof: For the sake of readability, the proof can be found at the end of this chapter in Section 7.8.3. □

Remark 7.12. *The interpretation of the convergence result in Theorem 7.11 is that the outputs of Algorithm 7.1 for all robots will converge to some arbitrary consensus signal $\tilde{\mathbf{p}}(t)$ in finite time, which is not necessarily the average $\bar{\mathbf{p}}_{\mathsf{G}}(t)$. Achieving consensus in finite time is interesting from the point of view of formation control since it allows to the observer and the controller to be designed independently without compromising the stability of the overall system as discussed in Section 7.5. Moreover, all robots will place themselves around the same formation center $\tilde{\mathbf{p}}(t)$ in finite time, which will converge towards $\bar{\mathbf{p}}_{\mathsf{G}}(t)$ according to Theorem 7.11.*

7.5 FORMATION CONTROL

Equipping all agents with the previously presented tools for distributed estimation we construct a local controller of the form:

$$\mathbf{u}_i(t) = \mathbf{p}_{i,m}(t) - \kappa_0(\mathbf{p}_i(t) - \mathbf{p}_{i,0}(t) - \mathbf{d}_i) - \sum_{\mu=1}^{m-1} \kappa_\mu(\mathbf{p}_i^{(\mu)}(t) - \mathbf{p}_{i,\mu}(t)) \tag{7.11}$$

where the roots of the polynomial $\lambda^m + \sum_{\mu=0}^{m-1} \kappa_\mu \lambda^\mu$ have negative real part.

Proposition 7.13. *Let the conditions of Theorem 7.11 hold. Define,* $\mathbf{p}_i^{\mathsf{r}}(t) := \bar{\mathbf{p}}_{\mathsf{G}}(t) + \mathbf{d}_i$ *with* $\bar{\mathbf{p}}_{\mathsf{G}}(t)$ *in* (7.8)*. Then, the closed loop system* (7.1) *under controller* (7.11) *satisfies the formation goal in* (7.3) *regardless of* $\mathbf{x}_i(0)$*.*

PROOF: First, note that due to Theorem 7.11, then there exists $T > 0$ such that $\mathbf{p}_{i,\mu}(t) \equiv \tilde{\mathbf{p}}^{(\mu)}(t)$. Moreover, the outputs $\mathbf{p}_{i,\mu}(t)$ remain bounded for $t \in [0, T]$. The closed loop system (7.1) under (7.11) is input to state stable with respect to $\mathbf{u}_i(t)$. Thus, there are no finite-time escapes of $\mathbf{x}_i(t)$ for any $t \in [0, T]$. For $t \geq T$, (7.11) takes the form

$$\mathbf{u}_i(t) = \tilde{\mathbf{p}}^{(m)}(t) - \kappa_0(\mathbf{p}_i(t) - \tilde{\mathbf{p}}(t) - \mathbf{d}_i) - \sum_{\mu=1}^{m-1} \kappa_\mu(\mathbf{p}_i^{(\mu)}(t) - \tilde{\mathbf{p}}^{(\mu)}(t))$$

Now, define $\mathbf{e}(t) = \mathbf{p}_i(t) - \tilde{\mathbf{p}}(t) - \mathbf{d}_i$ to obtain $\mathbf{e}^{(m)}(t) = -\sum_{\mu=0}^{m-1} \kappa_\mu \mathbf{e}^{(\mu)}(t)$ which is asymptotically stable towards the origin. Hence, $\mathbf{p}_i(t)$ converge asymptotically towards $\tilde{\mathbf{p}}(t) + \mathbf{d}_i$ which converge towards $\bar{\mathbf{p}}_{\mathsf{G}}(t) + \mathbf{d}_i$ due to Theorem 7.11, implying (7.3). □

Remark 7.14. *Note that a linear controller was proposed in* (7.11)*, whose design is decoupled from the sampling instants* $\boldsymbol{\tau}_i$ *arising from the perception mechanism. As evident from the proof of Proposition 7.13 and since the outputs of the estimation fusion technique from Algorithm 7.1 converge to all* $m+1$ *derivatives of* $\bar{\mathbf{p}}_{\mathsf{G}}(t)$*, then these ideas can be extended directly to other trajectory tracking controllers, without requiring a co-design between the perception mechanism and the controller in order to ensure stability.*

7.6 SIMULATION EXAMPLES

For simplicity in the presentation, we assume that each of the n coordinates is uncorrelated both in the detection and in the target model (7.2). Hence, it suffices to analyse each coordinate by separate. Equivalently, we consider $n = 1$. Now, let $m = 2$ for (7.1) and (7.2) to represent second-order integrators. All REDCHO instances are configured with $m = 2$ and gains $k_0 = 6, k_1 = 11, k_2 = 6, \gamma_0 = \gamma_1 = \gamma_2 = 1, \theta = 40$ obtained using the tuning rules from [1, 2]. Similarly, all robots use the controller (7.11) with $\kappa_0 = 1, \kappa_1 = 2$.

As characterized by [54], it is expected that a standard target detector based on, e.g., convolutional neural networks, improve its performance when more computing power is

employed, effectively increasing its perception latency. Hence, to illustrate the performance of our proposal under this perception latency and detection quality trade-off, the perception mechanism is configured with two possible perception methods with latencies $\Delta^1 = 1, \Delta^2 = 0.5$. These correspond to a computation-intensive detection method and a lighter one, respectively. As a result, for illustration purposes, we choose covariance matrices $\mathbf{R}^1 = 0.01, \mathbf{R}^2 = 0.1$ respectively to simulate that the first method performs better than the second one at the expense of large computing time.

Furthermore, let $\mathbf{W} = 1$ for the noise process in (7.2). The stochastic system (7.2) was simulated using Euler-Maruyama with time step $\Delta t = 10^{-6}$ whereas the REDCHO instances in (5.2) were simulated using explicit Euler method with the same time step.

7.6.1 Single robot

In this section, we consider $\mathsf{N} = 1$ in order to evaluate the performance of the SOE from Section 7.3. Consider a randomly generated sequence of perception latencies $\{\Delta_{1,k}\}_{k=0}^{\infty}$ with $\Delta_{1,k} \in \{1, 0.5\}$ leading to a sequence of sampling instants $\boldsymbol{\tau}_1 = \{\tau_{1,k}\}_{k=1}^{\infty}$. Figure 7.3 shows a realization $\mathbf{x}(t)$ of (7.2), and the SOE $\hat{\mathbf{x}}_1(t)$ from (7.7) for $\alpha = 1$. Similarly, we show the output $\hat{\mathbf{x}}_1^*(t)$ for the DOE in (7.6) for comparison which coincides with the Kalman filter at $t \in \boldsymbol{\tau}_1$. We construct a control input analogous to (7.11) as:

$$\mathbf{u}_i(t) = \ddot{\mathbf{p}}_1^{\mathsf{r}}(t) - \kappa_0(\mathbf{p}_1(t) - \mathbf{p}_1^{\mathsf{r}}(t)) - \kappa_1(\dot{\mathbf{p}}_1(t) - \dot{\mathbf{p}}_1^{\mathsf{r}}(t)) \tag{7.12}$$

for all $t \notin \boldsymbol{\tau}_1$, where $\mathbf{p}_1^{\mathsf{r}}(t) = \mathbf{C}\hat{\mathbf{x}}_1(t) + \mathbf{d}_1$ for the SOE and $\mathbf{p}_1^{\mathsf{r}}(t) = \mathbf{C}\hat{\mathbf{x}}_1^*(t) + \mathbf{d}_1$ for the DOE. The expressions for the derivatives $\ddot{\mathbf{p}}_1^{\mathsf{r}}(t), \dot{\mathbf{p}}_1^{\mathsf{r}}(t)$ can be obtained explicitly from (7.6) and (7.7) and are omitted here for brevity. Moreover, note that $\ddot{\mathbf{p}}_1^{\mathsf{r}}(t), \dot{\mathbf{p}}_1^{\mathsf{r}}(t), \forall t \in \boldsymbol{\tau}_1$ does not exist when using the DOE. Figure 7.4 shows the trajectory tracking performance of the robot (7.1) using the controller (7.12) for each case. It can be observed that the tracking error converges to zero when using the SOE. On the other hand, persistent transients are observed when using the DOE due to the discontinuities in the reference at $t \in \boldsymbol{\tau}_1$.

7.6.2 Multi-robot

Now, consider a communication network of $\mathsf{N} = 10$ robots connected in a ring topology. The estimation fusion protocol was implemented by separate for the X and Y components of a two dimensional target trajectory $\mathbf{p}(t)$. Figure 7.5 shows the convergence of the first components of $\mathbf{y}_{i,0}(t), \mathbf{y}_{i,1}(t), \mathbf{y}_{i,2}(t)$ for the estimation of the X coordinate, where it can be observed that all agents converge to a common signal in finite time, and then converge asymptotically to the first component of the centralized signal $\bar{\mathbf{y}}_{\mathsf{G}}(t)$ and its derivatives. The convergence of the rest of the components of $\mathbf{y}_{i,0}(t), \mathbf{y}_{i,1}(t), \mathbf{y}_{i,2}(t)$ as well as for $\mathbf{Q}_{i,0}(t), \mathbf{Q}_{i,1}(t), \mathbf{Q}_{i,2}(t)$ is similar and is omitted here for the sake of brevity. In addition, we use the Root Mean Squared (RMS) error performance index in this experiment to measure the impact of using the fusion block. For a single experiment of duration T, the RMS value for an arbitrary scalar signal $x(t)$ is computed as

$$\mathsf{RMS}\{x(t)\} := \sqrt{\frac{1}{T}\int_0^T x(t)^2 \mathrm{d}t}$$

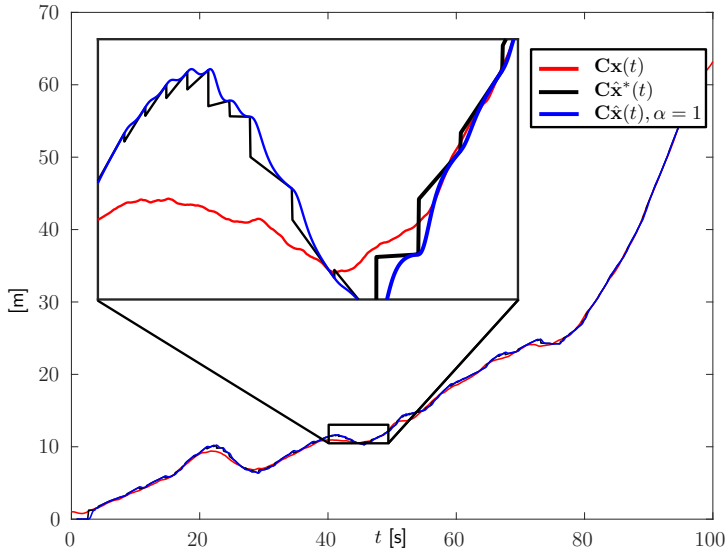

Figure 7.3: A realization of the target position $\mathbf{C}\mathbf{x}(t)$ as well as its corresponding estimations $\mathbf{C}\hat{\mathbf{x}}(t)$ and $\mathbf{C}\hat{\mathbf{x}}^*(t)$ for the SOE and the DOE respectively. Note that the SOE is always a smooth estimation whereas the DOE is discontinuous at the sampling instants.

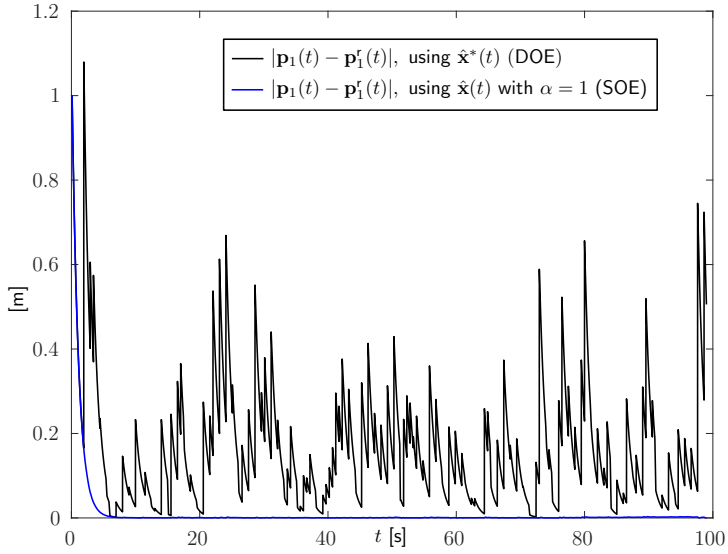

Figure 7.4: Trajectory tracking performance of a single robot (7.1) under the control input (7.12). Note that the reference is always smooth when using the SOE, resulting in asymptotic convergence for the tracking error. However, the reference is discontinuous when using the DOE, which leads to persistent transients in the robot performance, preventing asymptotic convergence of the tracking error.

Furthermore, we compute the average RMS value for arbitrary scalar signals $\{x_i(t)\}_{i=1}^{\mathsf{N}}$ as $\mathsf{RMS}_{\mathsf{avg}}\{x_i(t)\}_{i=0}^{\mathsf{N}} := \frac{1}{\mathsf{N}}\sum_{i=1}^{\mathsf{N}} \mathsf{RMS}\{x_i(t)\}$. The estimation error after the output stage of Algorithm 7.1 is shown in Figure 7.6 where it can be observed that the overall global estimate reduces the RMS error in a factor of 3.5 with respect to the average RMS of the individual local estimations. In addition, Figure 7.7 shows the actual formation behaviour of the robots on the plane. Here, the robots start at random positions and converge to a circular formation around the target estimation $\bar{\mathbf{p}}_{\mathsf{G}}(t)$.

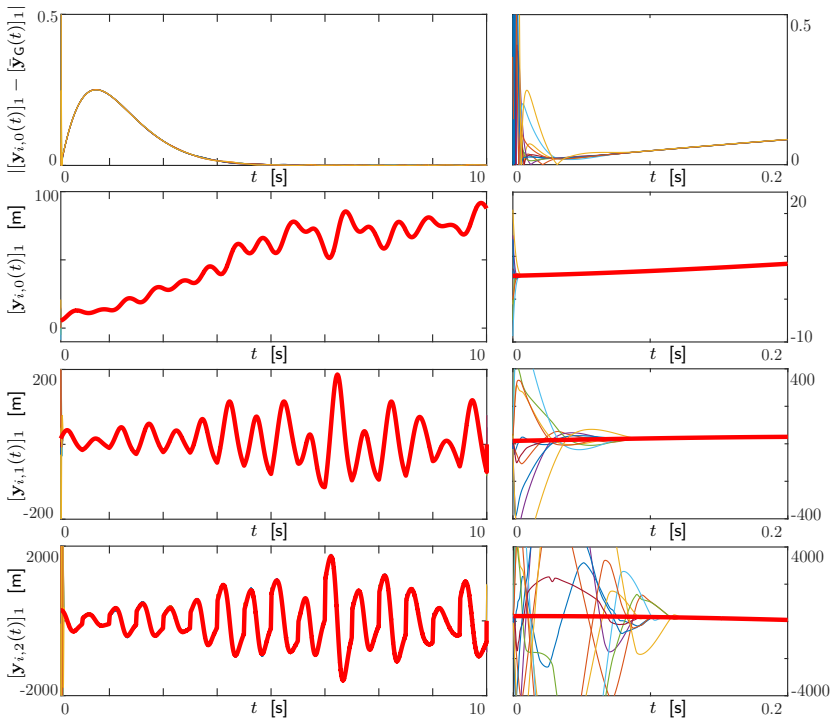

Figure 7.5: Trajectories for the first components of $\mathbf{y}_{i,0}(t), \mathbf{y}_{i,1}(t), \mathbf{y}_{i,2}(t)$ denoted as $[\mathbf{y}_{i,0}(t)]_1, [\mathbf{y}_{i,1}(t)]_1, [\mathbf{y}_{i,2}(t)]_1$, shown to converge to $[\bar{\mathbf{y}}_{\mathsf{G}}(t)]_1, [\dot{\bar{\mathbf{y}}}_{\mathsf{G}}(t)]_1, [\ddot{\bar{\mathbf{y}}}_{\mathsf{G}}(t)]_1$ which appear in solid red color. Figures on the left show trajectories in the interval $t \in [0, 10]$ to depict the asymptotic convergence behavior of the algorithm towards the global signals. On the other hand, figures on the right show convergence towards consensus which occurs in the interval $t \in [0, 0.2]$.

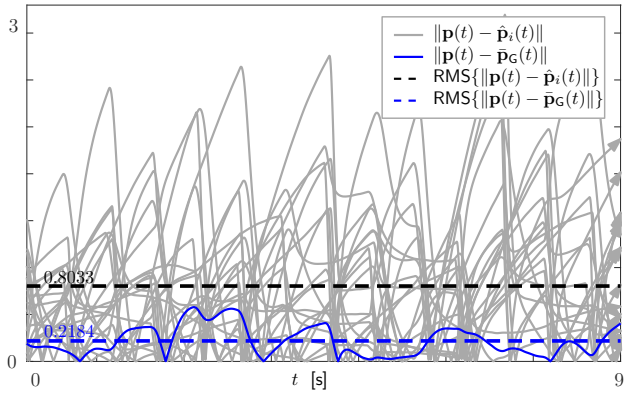

Figure 7.6: Error comparison between the actual realization of the target position $\mathbf{p}(t)$, local SOE estimates $\hat{\mathbf{p}}_i(t)$ (gray) and the collaborative estimation $\bar{\mathbf{p}}_\mathsf{G}(t)$ (blue). Moreover, average Root Mean Squared (RMS) values are shown in each case, where an improvement of a factor of 3.5 is observed when comparing the collaborative estimate with respect to the local ones.

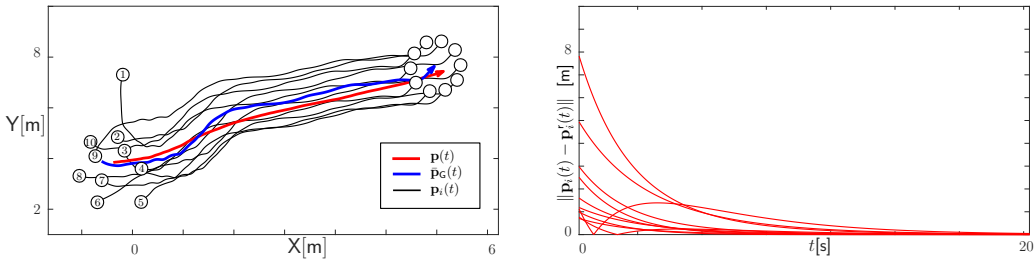

Figure 7.7: (Left) Actual target trajectory $\mathbf{p}(t)$ on the plane as well as the global estimation $\bar{\mathbf{p}}_\mathsf{G}(t)$ and individual robot trajectories $\mathbf{p}_i(t)$ which converge to the circular formation around $\bar{\mathbf{p}}_\mathsf{G}(t)$. (Right) Formation error between the robot position $\mathbf{p}_i(t)$ and the reference $\mathbf{p}_i^\mathsf{r}(t) = \bar{\mathbf{p}}_\mathsf{G}(t) + \mathbf{d}_i$.

7.6.3 Ablation and parameter analysis

In this section, we study the influence of the parameter α in the SOE as well as the fusion block. For this purpose, we perform $\mathsf{N}_\mathsf{MC} = 100$ Monte-Carlo runs, for each of the following configurations. We use a similar setting as in Section 7.6.2 with $\mathsf{N} = 10$ robots connected in a ring topology. We test the SOE with different values of the parameter $\alpha \in \{0.1, 1, 10\}$ as well as the DOE. In addition, for each value of α, we test the performance of the system when using the fusion block or not for computation of the reference signal $\bar{\mathbf{p}}_\mathsf{G}(t)$. In the case of the DOE, we do not include the fusion block, since the output of such estimator does not satisfy Assumption 7.9 due to the discontinuities in the output, regardless of the motion of the target. When the fusion block is not used, we compute $\hat{\mathbf{p}}_i(t) = \mathbf{C}\hat{\mathbf{x}}_i(t)$ from the SOE (7.7) or the DOE (7.6) depending on the configuration. Otherwise, we compute $\hat{\mathbf{p}}_i(t) = \hat{\mathbf{p}}_{i,0}(t)$ as the output of the estimation fusion output stage from Algorithm 7.1.

For each experiment, a different trajectory of the target (7.2) was generated in the interval $t \in [0, T]$ with $T = 100$, as well as different initial conditions for the robots in (7.1). As performance indicators, use the value of the estimation error $\mathsf{RMS}_{\mathsf{avg}}\{\|\hat{\mathbf{p}}_i(t) - \mathbf{p}(t)\|\}_{i=1}^{\mathsf{N}}$ and the tracking error $\mathsf{RMS}_{\mathsf{avg}}\{\|\bar{\mathbf{p}}_{\mathsf{G}}(t) + \mathbf{d}_i - \mathbf{p}_i(t)\|\}_{i=1}^{\mathsf{N}}$ where recall that $\mathbf{p}_i(t)$ is the position of the i-th robot. In addition, we measure the control effort by means of $\mathsf{RMS}_{\mathsf{avg}}\{\|\mathbf{u}_i(t)\|\}_{i=1}^{\mathsf{N}}$ and

$$\mathsf{PEAK}\{\mathbf{u}_i(t)\}_{i=1}^{\mathsf{N}} := \max_{i\in\{1,\ldots,\mathsf{N}\}} \sup_{t\in[0,T]} \|\mathbf{u}_i(t)\|$$

The results of these simulations are summarized in Table 7.1. The performance indicators in the rows of Table 7.1 were averaged over all 100 experiments for each column of the table. It can be observed that the DOE performs the best in terms of the estimation error among the configurations which do not use fusion. However, the SOE in conjunction with the fusion method is able to outperform the DOE in all cases, being the best configuration with $\alpha = 0.1$ due to the tight consistency of the estimations ensured by Theorem 7.5. The disadvantages of the DOE are more evident when looking at the tracking error, where it performs the worst up to 1 order of magnitude. The reason is that the the discontinuities in the DOE cause the persistent transients illustrated in Figure 7.4 whereas the SOE does not have this problem in any configuration.

On the other hand, the bad tracking performance of the DOE is balanced by the resulting small control effort both in RMS and peak values, outperforming the SOE in all cases. The reason is that the the DOE ignores the values of the derivatives of the estimation at the discontinuities since they are undefined for those instants. However, the SOE is able to compute these derivatives for all time. Due to the fact that as $\alpha \to 0$ one recovers a discontinuous behaviour, the derivatives can grow unbounded at the sampling instants as α is decreased. Since these derivatives are used explicitly in the control law (7.11), hence the control effort is impacted by the value of α in a similar fashion. Moreover, an equivalent behaviour happens when α increases, requiring a compromise for this parameter in terms of control effort. For example, the value of $\alpha = 1$ obtains good results for the control effort when compared to $\alpha = 0.1, 10$.

7.7 DISCUSSION

A combination of a smooth-output estimator and an estimation fusion stage was proposed for distributed target estimation. It was shown that, in contrast to the non-smooth optimal alternative, the formation control is able to achieve asymptotic convergence to the formation goal. This allows the control design to be decoupled from the perception-latency decisions. The advantages of the proposal where discussed through simulation examples, when compared to a non-smooth optimal alternative. The proposal is yet to be validated in real-world platforms, which imposes an interesting but highly non-trivial challenge to be explored in a future work.

It's important to note that the architecture considered in this chapter is closely related to the one discussed around Figure 1.4 in Chapter 1. We omitted the perception latency scheduling analysis for the local control branch. However, the ideas presented in Chapter 2 can be applied in such a case if the sensors used for self-localization are independent of those used for target detection. Another element not considered in this chapter is

Estimator	**DOE**	**SOE** $\alpha = 0.1$		**SOE** $\alpha = 1$		**SOE** $\alpha = 10$	
Fusion	✗	✗	✓	✗	✓	✗	✓
$\mathsf{RMS}_{\mathsf{avg}}\{\|\hat{\mathbf{p}}_i(t) - \mathbf{p}(t)\|\}_{i=1}^{\mathsf{N}}$	0.74	0.79	**0.19**	0.85	0.23	0.92	0.34
$\mathsf{RMS}_{\mathsf{avg}}\{\|\bar{\mathbf{p}}_{\mathsf{G}}(t) + \mathbf{d}_i - \mathbf{p}_i(t)\|\}_{i=1}^{\mathsf{N}}$	56.4	2.84	**2.81**	2.82	2.84	2.83	2.84
$\mathsf{RMS}_{\mathsf{avg}}\{\|\mathbf{u}_i(t)\|\}_{i=1}^{\mathsf{N}}$	**6.23**	18.43	22.74	10.75	9.24	15.22	16.81
$\mathsf{PEAK}\{\mathbf{u}_i(t)\}_{i=1}^{\mathsf{N}}$	**8.65**	70.83	75.98	30.21	34.45	65.01	63.78

Table 7.1: Ablation and parameter analysis for the proposal. The DOE is compared with our proposal using different configurations. Moreover, different performance indicators are depicted, where the best value for each row is marked in bold font. In particular the first two rows represent the estimation and tracking RMS errors measured in $[\mathsf{m}]$. The last two rows depict the control effort in terms of RMS and peak values, both measured in $[\mathsf{m}/\mathsf{s}^2]$.

asynchronous discrete-time communication. Since target detections only occur at discrete-time instants, it may be more natural to communicate only when new measurements are available rather than on a continuous-time basis, as explored in this chapter. This problem will be tackled in the next chapter.

7.8 PROOFS

7.8.1 Proof of Theorem 7.5.

For item **i)**, note that $\hat{\mathbf{y}}(t), \hat{\mathbf{Q}}(t)$ are m-times differentiable for any $t \notin \boldsymbol{\tau}$ from the expressions in (7.6) and (7.7) and item 1 of Definition 7.3. For $t \in \boldsymbol{\tau}$, take an arbitrary $\tau_k \in \boldsymbol{\tau}$ and compute the μ-th derivative of $\hat{\mathbf{y}}(t)$ with $\mu \in \{0, \dots, m\}$ as $t \to \tau_k^+$ as:

$$\begin{aligned}\lim_{t\to\tau_k^+} \hat{\mathbf{y}}^{(\mu)}(t) = \lim_{t\to\tau_k^+} \sum_{\nu=0}^{\mu} \binom{\mu}{\nu} \lambda_1^{(\nu)}(t|k)(\hat{\mathbf{y}}^*)^{(\mu-\nu)}(t|k-1) \\ + \binom{\mu}{\nu} \lambda_2^{(\nu)}(t|k)(\hat{\mathbf{y}}^*)^{(\mu-\nu)}(t|k) = (\hat{\mathbf{y}}^*)^{(\mu)}(\tau_k|k-1)\end{aligned}$$

where the μ-th derivative product rule was used as well as by noting that $\binom{\mu}{0} = 1$ and

$$\begin{aligned}\lim_{t\to\tau_k^+} \lambda_1(t|k) &= 1 - \eta(0;\alpha) = 1 \\ \lim_{t\to\tau_k^+} \lambda_2(t|k) &= \eta(0;\alpha) = 0 \\ \lim_{t\to\tau_k^+} \lambda_1^{(\nu)}(t|k) &= -\eta^{(\nu)}(0;\alpha) = 0 \\ \lim_{t\to\tau_k^+} \lambda_2^{(\nu)}(t|k) &= \eta^{(\nu)}(0;\alpha) = 0\end{aligned}$$

for any $\nu \in \{1, \dots, m\}$ by items 2 and 2 of Definition 7.3. Now, for $t \to \tau_k^-$ consider $\hat{\mathbf{y}}(t) = \lambda_1(t|k-1)\hat{\mathbf{y}}^*(t|k-2) + \lambda_2(t|k-1)\hat{\mathbf{y}}^*(t|k-1)$ for $t \in [\tau_{k-1}, \tau_k)$ from (7.7). Similarly as before,

$$\begin{aligned}\lim_{t\to\tau_k^-} \hat{\mathbf{y}}^{(\mu)}(t) = \lim_{t\to\tau_k^-} \sum_{\nu=0}^{\mu} \binom{\mu}{\nu} \lambda_1^{(\nu)}(t|k-1)(\hat{\mathbf{y}}^*)^{(\mu-\nu)}(t|k-2) \\ + \binom{\mu}{\nu} \lambda_2^{(\nu)}(t|k-1)(\hat{\mathbf{y}}^*)^{(\mu-\nu)}(t|k-1) = (\hat{\mathbf{y}}^*)^{(\mu)}(\tau_k|k-1)\end{aligned}$$

where the following identities were used:

$$\begin{aligned}\lim_{t\to\tau_k^-} \lambda_1(t|k-1) &= 1 - \eta((\tau_k - \tau_{k-1})/\Delta_{k-1};\alpha) = 0 \\ \lim_{t\to\tau_k^-} \lambda_2(t|k-1) &= \eta(1;\alpha) = 1 \\ \lim_{t\to\tau_k^-} \lambda_1^{(\nu)}(t|k-1) &= -\eta^{(\nu)}(1;\alpha) = 0 \\ \lim_{t\to\tau_k^-} \lambda_2^{(\nu)}(t|k-1) &= \eta^{(\nu)}(1;\alpha) = 0\end{aligned}$$

due to $(\tau_k - \tau_{k-1}) = \Delta_{k-1}$ and Definition 7.3. Hence, the two-sided limit $\lim_{t\to\tau_k} \hat{\mathbf{y}}^{(\mu)}(t) = (\hat{\mathbf{y}}^*)^{(\mu)}(\tau_k|k-1)$ exists for arbitrary $\tau_k \in \boldsymbol{\tau}$. The proof is the same for $\hat{\mathbf{Q}}(t)$. For item **ii)** note that $\hat{\mathbf{x}}(t|k)$ is unbiased since it comes from the Kalman filter structure of (7.5)

and (7.6). Hence, $\hat{\mathbf{x}}(t)$ is unbiased by linearity of $\mathbb{E}\{\bullet\}$ and by the definition of $\hat{\mathbf{x}}(t)$ from (7.7) as a convex combination of unbiased estimates since $\lambda_1(t|k)+\lambda_2(t|k)=1$. For item **iii)** note that $\hat{\mathbf{P}}^*(t|k), \hat{\mathbf{P}}^*(t|k-1)$ for $t \in [\tau_k, \tau_{k+1})$ are covariances for $\hat{\mathbf{x}}^*(t|k), \hat{\mathbf{x}}^*(t|k-1)$ but that $\mathsf{cov}\{\mathbf{x}(t)-\hat{\mathbf{x}}^*(t|k), \mathbf{x}(t)-\hat{\mathbf{x}}^*(t|k-1)\} \neq 0$ since both estimations are correlated by their dependence of the noise process $\{\mathbf{u}(t')|t' \leq t\}$ in (7.2). However, note that the expression of $\hat{\mathbf{Q}}(t)$ is a convex combination of $\hat{\mathbf{Q}}^*(t|k), \hat{\mathbf{Q}}^*(t|k-1)$. Hence, the covariance-intersection principle from [164, Section 2.1] ensures consistency of $\hat{\mathbf{P}}(t)$ i.e., $\hat{\mathbf{P}}^*(t) \preceq \hat{\mathbf{P}}(t)$. Finally, note that by letting $\alpha \to 0$ we have $\lambda_1(t|k)=0, \lambda_1(t|k)=1$ for all $t \in (\tau_k, \tau_{k+1})$ using item 4 from Definition 7.3. Hence, $\lim_{\alpha\to 0} \hat{\mathbf{P}}(t) = \hat{\mathbf{P}}^*(t|k) = \hat{\mathbf{P}}^*(t), \forall t \notin \boldsymbol{\tau}$.

7.8.2 Proof of Lemma 7.8

For item 1, take the μ-th derivative of the identity $\mathbf{I}_m = \bar{\mathbf{Q}}_{\mathsf{G}}(t)\bar{\mathbf{P}}_{\mathsf{G}}(t)$ yielding:

$$0 = \sum_{\nu=0}^{\mu} \binom{\mu}{\nu} \bar{\mathbf{Q}}_{\mathsf{G}}^{(\mu-\nu)}(t)\bar{\mathbf{P}}_{\mathsf{G}}^{(\nu)}(t) = \bar{\mathbf{Q}}_{\mathsf{G}}(t)\bar{\mathbf{P}}_{\mathsf{G}}^{(\mu)}(t) + \sum_{\nu=0}^{\mu-1} \binom{\mu}{\nu} \bar{\mathbf{Q}}_{\mathsf{G}}^{(\mu-\nu)}(t)\bar{\mathbf{P}}_{\mathsf{G}}^{(\nu)}(t)$$

where $\binom{\mu}{\mu}=1$ was used and from which the expression in item **i)** is obtained by solving for $\bar{\mathbf{P}}_{\mathsf{G}}^{(\mu)}(t)$. Finally, item 2 is obtained by applying the μ-th derivative to the definition $\bar{\mathbf{p}}_{\mathsf{G}}(t) = \mathbf{C}\bar{\mathbf{P}}_{\mathsf{G}}(t)\bar{\mathbf{y}}_{\mathsf{G}}$, completing the proof.

7.8.3 Proof of Theorem 7.11

First, note that each component of the equations in (7.9) and (7.10) is a REDCHO block of the form (5.2). Each individual REDCHO block in (7.9) (resp. (7.10)) has scalar input $z_i(t)$ given by one component of $\hat{\mathbf{y}}_i(t)$ (resp. $\hat{\mathbf{Q}}_i(t)$), internal scalar variables $\{v_{i,\mu}(t)\}_{\mu=0}^m$ given by one component of each $\{\mathbf{v}_{i,\mu}(t)\}_{\mu=0}^m$ (resp. $\{\mathbf{V}_{i,\mu}(t)\}_{\mu=0}^m$) and outputs $\{s_{i,\mu}(t)\}_{\mu=0}^m$ given by one component of each $\{\mathbf{y}_{i,\mu}(t)\}_{\mu=0}^m$ (resp. $\{\mathbf{Q}_{i,\mu}(t)\}_{\mu=0}^m$). Hence Assumption together with item Theorem 5.10 in Chapter 5 imply that there exists $T>0$ such that

$$\tilde{\mathbf{y}}^{(\mu)}(t) = \mathbf{y}_{1,\mu}(t) = \cdots = \mathbf{y}_{\mathsf{N},\mu}(t)$$
$$\tilde{\mathbf{Q}}^{(\mu)}(t) = \mathbf{Q}_{1,\mu}(t) = \cdots = \mathbf{Q}_{\mathsf{N},\mu}(t)$$

for all $\mu \in \{0,\ldots,m\}$ for some consensus signals $\tilde{\mathbf{y}}(t) \in \mathbb{R}^{nm}, \tilde{\mathbf{Q}}(t) \in \mathbb{R}^{nm\times nm}$. Moreover, define $\tilde{\mathbf{P}}(t) = \tilde{\mathbf{Q}}(t)^{-1}$ Hence, for all $t \geq T$ and $i \in \{1,\ldots,N\}$ the expressions in Algorithm 7.1 reads:

$$\begin{aligned}
\mathbf{p}_{i,0}(t) &\equiv \mathbf{C}\tilde{\mathbf{P}}(t)\tilde{\mathbf{y}}_0(t)\\
\mathbf{P}_{i,\mu}(t) &\equiv -\tilde{\mathbf{P}}(t)\sum_{\nu=0}^{\mu-1}\binom{\mu}{\nu}\tilde{\mathbf{Q}}^{(\mu-\nu)}(t)\mathbf{P}_{i,\nu}(t)\\
\mathbf{p}_{i,\mu}(t) &\equiv \mathbf{C}\sum_{\nu=0}^{\mu}\binom{\mu}{\nu}\mathbf{P}_{i,\nu}(t)\tilde{\mathbf{y}}^{(\mu-\nu)}(t)
\end{aligned} \tag{7.13}$$

Now, Theorem 5.10 in Chapter 5 imply that $\tilde{\mathbf{y}}(t), \tilde{\mathbf{Q}}(t)$ converge to $\bar{\mathbf{y}}_{\mathsf{G}}(t), \bar{\mathbf{Q}}_{\mathsf{G}}(t)$ as $t \to \infty$. Hence, the expressions in (7.13) converge to $\bar{\mathbf{p}}_{\mathsf{G}}(t), \bar{\mathbf{P}}_{\mathsf{G}}^{(\mu)}(t), \bar{\mathbf{p}}_{\mathsf{G}}^{(\mu)}(t)$ respectively due to Lemma 7.8, completing the proof.

Chapter Eight

EDC Under Asynchronous Communication

One element of the architecture proposed in Chapter 1, as depicted in Figure 1.4, which has not been discussed thus far is the possibility of having discrete-time asynchronous communication between agents. In this thesis, cooperation between agents for information fusion has been successfully addressed using modular EDC tools, but only under the assumption of continuous-time communication. To complement the presentation of the tools in this thesis, we now analyze new EDC tools under asynchronous communication, where communication instants are determined by an exogenous scheduler, as depicted in Figure 8.1. This scheduler can coincide with the perception latency scheduler, ensuring that agents communicate with their neighbors only when new measurements for the target of interest are available, which aligns well with practical considerations.

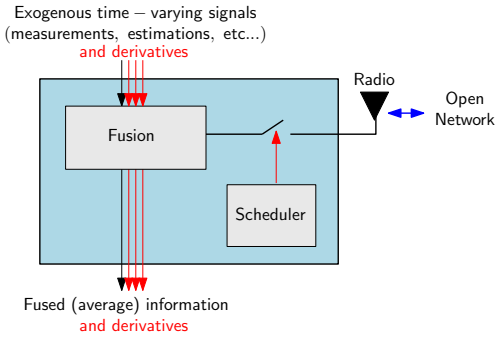

Figure 8.1: Individual agent architecture for distributed information averaging.

In this chapter, we specifically focus on considering an arbitrary sequence of communication instants for each agent. The primary objective remains similar to the standard REDCHO algorithm, which is to compute the time-varying average of a set of signals distributed across an OMAS. We also analyse robustness to symmetric propagation delays. For simplicity, we do not consider weighted averages or self-contained distributed differentiation as discussed in Chapter 6. However, it is important to note that the concepts introduced in this chapter can be extended to include those aspects as well. The

contributions of this chapter were also published in [9].

8.1 RELATED WORK

Dealing with network constraints such as asynchronous sampling and delays is not trivial, leading many authors to simplify the problem by assuming continuous-time communication [85, 154, 162, 166] or periodic discrete-time communication [97–99] without considering communication delays in many cases. Nonetheless, in [167], the authors study the robustness of consensus algorithms concerning delays in communication. In addition, some works such as those based on the concept of event-triggered control, allow asynchronous communication by construction [100, 101].

In general, the study of exogenous asynchronous communication instants and communication delays can be tackled by tools from hybrid systems [168]. Similarly, tools from the literature on analysis of systems under time-varying delays can be used to study both of these problems, as exposed in [72, 169]. The following are some examples of works which have used these tools in the context of consensus. In [170], a static consensus protocol is proposed, which works under asynchronous communication instants. The authors in [102] also deal with a static consensus problem under the presence of non-cooperative agents and asynchronous communication. In [103], a leader-follower protocol is proposed, robust to asynchronous communication in the network. A similar problem is tackled in [106], where the asynchronous communication instants come from an event-triggered rule rather than an arbitrary sequence of time instants. Other works analyze transmission delays as in [104] under an event-triggered setting and in [105] for asynchronous switched networks.

An existing work that is closely related is [171], where multiple agents can synchronize to a time-varying function despite asynchronous communication. However, the protocol is linear, and only achieves bounded steady-state error. Additionally, the protocol converges towards a time-varying signal that is not related to the average of the references at each agent.

8.2 PROBLEM STATEMENT AND PROTOCOL PROPOSAL

Consider a multi-agent system distributed in a network $\mathcal{G}$. In the following, $\mathcal{G}$ is an undirected connected graph of N nodes where each edge in the graph corresponds to an available communication link between two agents. Denote with $\mathcal{N}_i \subseteq \{1, \dots, \mathsf{N}\}$ the index set containing the neighbors of agent i. We consider that each agent i has access to a local $(m+1)$-times differentiable reference signal $z_i(t)$. In addition, in order to reduce the communication burden, messages between agents can only be exchanged at discrete-time instants. Agent i can only initiate an information exchange with its neighbors at some scheduled (perhaps irregular) time instants $t \in \boldsymbol{\tau}_i := \{\tau_{i,k}\}_{k=0}^{\infty}$, referred as *transaction instants*. At these instants, agent i transmits and then receives information from its neighbors. These instants are assumed to comply with $0 < \tau_{i,k+1} - \tau_{i,k} \leq \Delta$ for a fixed $\Delta > 0$ as well as $\lim_{k\to\infty} \tau_{i,k} = \infty$ to avoid Zeno-behaviour.

The goal of the agents is to share information between them in order to collectively estimate the following dynamic average signal in a distributed fashion

$$\bar{z}(t) = \frac{z_1(t) + \cdots + z_{\mathsf{N}}(t)}{\mathsf{N}}$$

We aim to obtain *quasi-exact* performance meaning that $\bar{z}(t)$ is recovered exactly at all agents when $\Delta \to 0$ and $t \to \infty$.

The algorithm proposal is divided in two parts: a communication protocol and an estimation algorithm.

Communication protocol: First, at any time $t \geq 0$, each agent $i \in \{1, \ldots, \mathsf{N}\}$ has a local estimation $\hat{z}_i(t)$ for the global average signal $\bar{z}(t)$. At each transaction instant $t = \tau_{i,k} \in \boldsymbol{\tau}_i$, agent i sends its own estimate $\hat{z}_i(\tau_{i,k})$ to its neighbors $j \in \mathcal{N}_i$ and asks them for their estimate $\hat{z}_j(\tau_{i,k})$ at that time. Then, agent i stores the difference $\hat{z}_{ij}(\tau_{i,k}) = \hat{z}_i(\tau_{i,k}) - \hat{z}_j(\tau_{i,k})$. This difference is not only updated at $t \in \boldsymbol{\tau}_i$ but also at $t \in \boldsymbol{\tau}_j, j \in \mathcal{N}_j$ when a neighbor j sends agent i its own estimate first. This protocol is depicted in Figure 8.2. Denote with $\tau_{ij}(t)$ the last instant in which the link (i, j) was updated under this setting, this is, either triggered by a transaction instant at agent i or at agent j. Concretely, $\tau_{ij}(t) = \max\{\tau \in (\boldsymbol{\tau}_i \cup \boldsymbol{\tau}_j) : \tau \leq t\}$ which complies $\tau_{ij}(t) = \tau_{ji}(t)$. Hence, since the difference $\hat{z}_{ij}(t)$ is updated symmetrically at both sides of the communication link, then $\hat{z}_{ij}(t) = -\hat{z}_{ji}(t)$ for any $t \in (\boldsymbol{\tau}_i \cup \boldsymbol{\tau}_j)$. Equivalently, the difference $\hat{z}_{ij}(t)$ stored at agent i coincides with the difference $\hat{z}_{ji}(t)$ stored at agent j except for a change of sign.

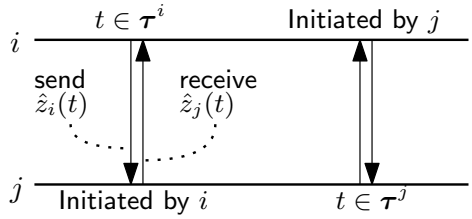

Figure 8.2: The communication protocol from the perspective of agent i exchanging information with agent j is depicted under nominal asynchronous communication where no delays are present.

So far, we have described the scenario referred in this chapter as *nominal asynchronous communication*, where we do not consider communication delays, and each agent can sample from $z_i(t)$ at any time $t \geq 0$. However, in the subsequent sections, we broaden the scope and consider scenarios that include communication delays and situations in which $z_i(t)$ can only be sampled at specific times $t \in \boldsymbol{\tau}_i$.

Estimation algorithm: Now, we present the algorithm that computes the local estimation $\hat{z}_i(t)$ for $\bar{z}(t)$ at agent $i \in \{1, \ldots, \mathsf{N}\}$. To do so, each agent i stores $m + 1$ internal variables denoted as $\{\chi_{i,\mu}(t)\}_{\mu=0}^{m}$. The evolution of these variables as well as the

definition of the estimation output is subject to the following:

Dynamic evolution:

$$\begin{aligned}\dot{\chi}_{i,\mu}(t) &= k_\mu \theta^{\mu+1} \sum_{j\in\mathcal{N}_i} F_\mu(\hat{z}_{ij}(\tau_{ij}(t))) + \chi_{i,\mu+1}(t) \\ &\quad -\gamma_\mu \chi_{i,\mu}(t), \qquad \text{for } \mu \in \{0,\dots,m-1\} \\ \dot{\chi}_{i,m}(t) &= k_m \theta^{m+1} \sum_{j\in\mathcal{N}_i} F_m(\hat{z}_{ij}(\tau_{ij}(t))) - \gamma_m \chi_{i,m}(t)\end{aligned} \tag{8.1}$$

Output:

$$\begin{aligned}\hat{z}_i(t) &= \hat{z}_{i,0}(t) \\ \hat{z}_{i,\mu}(t) &= z_i^{(\mu)}(t) - \textstyle\sum_{\nu=0}^{m} \mathbf{G}_{\mu+1,\nu+1}\chi_{i,\nu}(t)\end{aligned}$$

where $F_\mu(\bullet) = \lceil \bullet \rfloor^{\frac{m-\mu}{m+1}}$, $\{\mathbf{G}_{\mu+1,\nu+1}\}_{\nu=0,\mu=0}^{m,m}$ are the components of the matrix $\mathbf{G}$ and $\theta, \{k_\mu, \gamma_\mu\}_{\mu=0}^{m}$ are all positive design parameters. For conciseness, and to aid the analysis, the evolution of $\{\chi_{i,\mu}(t)\}_{\mu=0}^{m}$ is written in terms of a differential equation in (8.1). However, there is an explicit expression for these trajectories provided in Appendix 8.3, removing the necessity to use approximations to solve for trajectories of (8.1).

To show convergence of (8.1), we rely on the following assumption:

Assumption 8.1. *Let*

$$s_i(t) = z_i^{(m+1)}(t) + \sum_{\mu=0}^{m} l_\mu z_i^{(\mu)}(t)$$

where $\{l_\mu\}_{\mu=0}^{m}$ *are the coefficients of the polynomial*

$$(\lambda+\gamma_0)\cdots(\lambda+\gamma_m) = \lambda^{m+1} + \sum_{\mu=0}^{m} l_\mu \lambda^\mu.$$

Thus, $|s_i(t)| \leq L, \forall t \geq 0$ *for fixed* $\{\gamma_\mu\}_{\mu=0}^{m}$ *and known* $L \geq 0$.

In the following, we present the main result of this chapter, which is the convergence characterization of (8.1) for nominal asynchronous communication.

Theorem 8.2. *Let,* $\mathcal{G}$ *be a connected graph. Consider arbitrary scheduled transaction instants with* $\Delta \in (0, \Delta_{\max}]$, $\Delta_{\max} > 0$. *Then, for any* $\Delta_{\max} > 0$ *there exist* $\gamma_{\max} > 0$ *sufficiently small with* $\gamma_0, \dots, \gamma_m \in (0, \gamma_{\max}]$ *such that if Assumption 8.1 holds, there exists:*

- *convergence time* $T > 0$ *and sufficiently big* $\theta, \{k_\mu\}_{\mu=0}^{m}$,
- *admissible intervals* $\mathcal{R}_0, \dots, \mathcal{R}_m \subset \mathbb{R}$ *for the initial conditions* $\{\chi_{i,\mu}(0)\}_{\mu=0}^{m}$,
- m*-times differentiable consensus signal* $\zeta(t)$,

such that algorithm (8.1) *complies:*

1. $|\zeta^{(\mu)}(t) - \hat{z}_{i,\mu}(t)| \leq c_\mu \Delta^{m-\mu+1}$, $\forall t \geq T, \forall i \in \{1,\dots,N\}$ *with positive constants* $\{c_\mu\}_{\mu=0}^{m}$ *depending only on the algorithm parameters,* $\mathcal{G}$ *and* $\Delta_{\max}$.

2. $\lim_{t\to\infty} |\zeta(t) - \bar{z}(t)| = 0$.

3. *The intervals $\mathcal{R}_0, \ldots, \mathcal{R}_m$ can be made arbitrarily large by increasing θ.*

PROOF: The proof is provided Section 8.4.2. □

Remark 8.3. *Item 1 of Theorem 8.2 implies that $\hat{z}_i(t) = \hat{z}_{i,0}(t)$ converges exactly to $\zeta(t)$ when $\Delta \to 0$. This leads to asymptotic convergence towards $\bar{z}(t)$ without any steady-state error, resulting in quasi-exact performance. This is an advantage with respect to linear techniques such as [67, 98, 171] where exact convergence cannot be obtained for persistently varying signals even in the continuous-communication limit.*

Remark 8.4. *Algorithm 8.1 is based on sliding modes, which results in chattering due to the discontinuous term in the equation for $\dot{\chi}_{i,m}(t)$. However, Theorem 8.2 shows that the chattering has a worst-case effect proportional to Δ^{m+1} on the estimation of $\bar{z}(t)$. This improvement is due to the HOSM in* (8.1)*, which reduces chattering, and is superior to FOSM approaches for dynamic consensus, such as [85].*

Remark 8.5. *Note that the output in* (8.1) *for $\mu = 0$ leads to*

$$\hat{z}_i(t) = \hat{z}_{i,0}(t) = z_i(t) - \chi_{i,0}(t)$$

which can be computed without requiring any derivatives of $z_i(t)$. In addition, the remaining $\{\hat{z}_{i,\mu}(t)\}_{\mu=1}^{m}$ are not utilized in computing the dynamic evolution in (8.1) *or $\hat{z}_i(t)$. However, if the derivatives of $z_i(t)$ are available, Theorem 8.2 implies that the derivatives $\bar{z}^{(\mu)}(t)$ can be estimated via $\hat{z}_{i,\mu}(t)$ at each agent. This property is shared with other high-order EDC protocols such as [1, 2, 10] and is important for applications such as formation tracking for mobile robots [172].*

8.2.1 Non-cooperative sampling

Note that in order to update the dynamic evolution of the algorithm in (8.1), it suffices to sample the signal $z_i(t)$ at $t \in \bar{\boldsymbol{\tau}}_i := \bigcup_{j\in\mathcal{N}_i} \boldsymbol{\tau}_j \cup \boldsymbol{\tau}_i$, which we refer here as *cooperative sampling*. The reason comes from the way the differences $\hat{z}_{ij}(\tau_{ij}(t))$ are constructed, which samples from $\hat{z}_i(t)$ only at transaction instants in $\bar{\boldsymbol{\tau}}_i$. Hence, continuous sampling for $z_i(t)$ is not required for the algorithm to work. However, in the case of *non-cooperative sampling*, the signal $z_i(t)$ is only sampled at the scheduled instants $\boldsymbol{\tau}_i$, instead of $\bar{\boldsymbol{\tau}}_i$. This case requires the following modification of Algorithm 8.1. Consider a new piece-wise constant output $\hat{z}_{i,\mu}(t)$ of the form

$$\hat{z}_{i,\mu}(t) = z_i^{(\mu)}(\tau_{i,k}) - \sum_{\nu=0}^{m} \mathbf{G}_{\mu+1,\nu+1}\chi_{i,\nu}(\tau_{i,k}), \tag{8.2}$$

for $t \in [\tau_{i,k}, \tau_{i,k+1})$ instead of the output in (8.1). Hence, a similar result as the one from Theorem 8.2 is obtained.

Corollary 8.6. *The statement of Theorem 8.2, replacing the output in* (8.1) *with* (8.2)*, is true.*

PROOF: The proof is provided in Section 8.4.3 □

8.2.2 Symmetric communication delays

In the previous exposition, it was assumed that the exchange of messages between agents occurred instantly. However, in practice, there is a finite delay between the time an agent i transmits a message and the time it is received at agent j. While the single-trip delay cannot be directly measured, the round-trip delay can be measured when agent j responds back to the message. This is, when agent i sends a message at a specific time $t = \tau_{ij}^a \in \boldsymbol{\tau}_i$, agent j receives it at a later time $t = \tau_{ij}^b > \tau_{ij}^a$, and then responds to agent i. This response message arrives at agent i at a later time $t = \tau_{ij}^c > \tau_{ij}^b$. Therefore, agent i can determine the round-trip delay as $d_{ij} = \tau_{ij}^c - \tau_{ij}^a$. In this context, it is reasonable to assume that the single-trip delays are symmetric, meaning that the time it takes for a message to travel from agent i to j is equal to the time it takes for the response message to travel from agent j to i. This assumption implies that $\tau_{ij}^b - \tau_{ij}^a = \tau_{ij}^c - \tau_{ij}^b = d_{ij}/2$ for all i and j. This assumption is reasonable in practice, specially when nodes exchange information through one-hop communication [173].

With this information, updates for the difference $\hat{z}_{ij}(\tau_{ij}(t))$ can be synchronized as described in Figure 8.3. First, agent i request a transaction with agent j at time $t = \tau_{ij}^a$. The message is received at agent j at time $t = \tau_{ij}^b$ from which it sends its current estimate $\hat{z}_j(\tau_{ij}^b)$. This message is received at agent i at $t = \tau_{ij}^c$ and it can compute $d_{ij} = \tau_{ij}^c - \tau_{ij}^a$. Then, it transmits its previous estimate $\hat{z}_i(\tau_{ij}^c - d_{ij}/2) = \hat{z}_i(\tau_{ij}^b)$.

The end of the transaction is denoted as τ_{ij}^d, which is the time when the last message is received at agent j, or equivalently, $\tau_{ij}^d = \tau_{ij}^c + d_{ij}/2$ from the perspective of agent i. At $t = \tau_{ij}^d$, both agents have synchronized versions of $\hat{z}_i(\tau_{ij}^b)$ and $\hat{z}_j(\tau_{ij}^b)$, allowing them to compute the difference $\hat{z}_i(\tau_{ij}^b) - \hat{z}_j(\tau_{ij}^b)$. We set $\tau_{ij}(t), d_{ij}(t)$ to the last end of transaction instant before some time t for link (i,j) and its corresponding round-trip delay respectively. In the example in Figure 8.3, if $t \geq \tau_{ij}^d$ and before any other transaction instant occurs for the link (i,j), then $\tau_{ij}(t) = \tau_{ij}^d$ and $\hat{z}_i(\tau_{ij}^b) - \hat{z}_j(\tau_{ij}^b) = \hat{z}_i(\tau_{ij}(t) - d_{ij}(t)) - \hat{z}_j(\tau_{ij}(t) - d_{ij}(t))$ Therefore, using the information available at each agent we can define

$$\hat{z}_{ij}(\tau_{ij}(t)) = \hat{z}_i(\tau_{ij}(t) - d_{ij}(t)) - \hat{z}_j(\tau_{ij}(t) - d_{ij}(t)) \tag{8.3}$$

which is a delayed version of the original $\hat{z}_{ij}(\tau_{ij}(t))$ satisfying $\hat{z}_{ij}(\tau_{ij}(t)) = -\hat{z}_{ji}(\tau_{ji}(t))$ as well. With this new protocol, we obtain an equivalent version of Theorem 8.2 as follows.

Corollary 8.7. *Consider any delay $d_{ij}(t) \in [0, d], \forall t \geq 0$ for some maximum delay $d > 0$. Hence, the statement of Theorem 8.2, replacing $\hat{z}_{ij}(\tau_{ij}(t))$ with (8.3) and Δ replaced by $\Delta + d$, is true.*

PROOF: The proof is provided in Section 8.4.3 □

It is important to note that at $t = \tau_{ij}^c$ in Figure 8.3, agent i must transmit a past estimate $\hat{z}_{i,j}(\tau_{ij}^c - d_{ij}/2)$ and not its current estimate $\hat{z}_{i,j}(\tau_{ij}^c)$. It may seem that it implies agent i must store all previous values of $\hat{z}_{ij}(\tau), \forall \tau \in [t - (d + \Delta), t]$ for this to be possible, since $\tau_{ij}^c - d_{ij}/2$ do not need to belong to $\bar{\boldsymbol{\tau}}_i$. Evidently, storing all these infinite values is impossible. However, if agent i at time t has stored the finite number of values $\hat{z}_i(\tau), \tau \in \bar{\boldsymbol{\tau}}_i \cap [t-(d+\Delta), t]$ in a buffer, then it can recover all other $\hat{z}_i(\tau), \forall \tau \in [t-(d+\Delta), t]$ through the explicit expression of the solutions of (8.1) provided in the next Section.

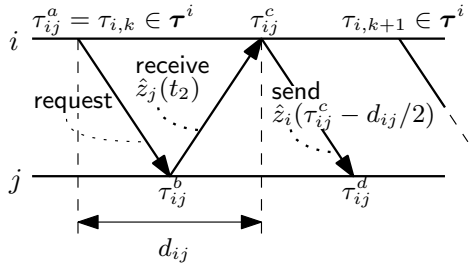

Figure 8.3: Communication protocol with symmetric round-trip delays from the perspective of agent i communicating with agent j.

8.3 EXACT SOLUTION FOR THE DYNAMIC EVOLUTION

In this section, we provide exact solutions for (8.1) without requiring to use any discretization or approximation. We use the following notation for simplicity. Let

$$\begin{aligned} g_{i,\mu}(t) &= k_\mu \theta^{\mu+1} \sum_{j \in \mathcal{N}_i} \lceil \hat{z}_{ij}(\tau_{ij}(t)) \rfloor^{\frac{m-\mu}{m+1}} \\ \mathbf{g}_i(t) &= [g_{i,0}(t), \dots, g_{i.m}(t)]^\top \\ \mathbf{x}_i(t) &= [\chi_{i,0}(t), \dots, \chi_{i,m}(t)]^\top. \end{aligned}$$

Moreover, let $\bar{\boldsymbol{\tau}}_i = \bigcup_{j \in \mathcal{N}_i} \boldsymbol{\tau}_j \cup \boldsymbol{\tau}_i$ be the set of all transaction instants at agent i including the ones started by i and the ones started by its neighbors. Then, consider arbitrary consecutive instants $\tau_k, \tau_{k+1} \in \bar{\boldsymbol{\tau}}_i$ with $\tau_k < \tau_{k+1}$. Hence, an explicit expression of the trajectories of (8.1) for the interval $t \in [\tau_k, \tau_{k+1})$ is provided in the following.

Proposition 8.8. *The unique solution to* (8.1) *for the interval* $t \in [\tau_k, \tau_{k+1})$ *is*

$$\mathbf{x}_i(t) = \exp(\boldsymbol{\Gamma}(t - \tau_k))\mathbf{x}_i(\tau_k) + \boldsymbol{\Gamma}^{-1}(\exp(\boldsymbol{\Gamma}(t - \tau_k)) - \mathbf{I}_m)\mathbf{g}_i(\tau_k) \tag{8.4}$$

PROOF: First, write (8.1) in vector form as

$$\dot{\mathbf{x}}_i(t) = \boldsymbol{\Gamma}\mathbf{x}_i(t) + \mathbf{g}_i(t) \tag{8.5}$$

Then, note that due to how the differences $\hat{z}_{ij}(\tau_{ij}(t))$ are constructed, then $\mathbf{g}_i(t) = \mathbf{g}_i(\tau_k)$ remains constant for all $t \in [\tau_k, \tau_{k+1})$. Then, for such interval, (8.5) is a linear ordinary differential equation with a constant input, whose solution is given by

$$\mathbf{x}_i(t) = \exp(\boldsymbol{\Gamma}(t - \tau_k))\mathbf{x}_i(\tau_k) + \int_{\tau_k}^{t} \exp(\boldsymbol{\Gamma}(t - \tau))\mathrm{d}\tau \mathbf{g}_i(\tau_k)$$

which reduces to (8.4). □

8.4 CONVERGENCE ANALYSIS

8.4.1 Auxiliary results for nominal asynchronous communication

In this section we derive the necessary results in order to show Theorem 8.2 which considers cooperative sampling and non-delayed asynchronous communication. First, we write (8.1) in compact form as follows. Let $\boldsymbol{\chi}_\mu(t) = [\chi_{1,\mu}(t), \dots, \chi_{\mathsf{N},\mu}(t)]^\top$, $\mathbf{z}(t) = [z_1(t), \dots, z_{\mathsf{N}}(t)]^\top$ and $\hat{\mathbf{z}}_\mu(t) = [\hat{z}_{1,\mu}(t), \dots, \hat{z}_{\mathsf{N},\mu}(t)]^\top$. Now, denote with $\mathbf{b}_i \in \mathbb{R}^{\mathsf{N}}$ the i-th standard unit vector. Let $\mathcal{E}$ represent the set of directed edges of $\mathcal{G}$ picking an arbitrary orientation, and $\mathbf{D}$ its incidence matrix. Each column of $\mathbf{D}$ corresponds to a link in $\mathcal{E}$. Hence, for the link $(i,j) \in \mathcal{E}$, the corresponding column of $\mathbf{D}$ can be written as $\mathbf{b}_{ij} = \mathbf{b}_i - \mathbf{b}_j$.

For arbitrary link (i,j), it follows that $\hat{z}_i(t) = \mathbf{b}_i^\top \hat{\mathbf{z}}_0(t)$ and therefore,

$$\hat{z}_{ij}(\tau_{ij}(t)) = \mathbf{b}_i^\top \hat{\mathbf{z}}_0(\tau_{ij}(t)) - \mathbf{b}_j^\top \hat{\mathbf{z}}_0(\tau_{ij}(t)) = \mathbf{b}_{ij}^\top \hat{\mathbf{z}}_0(\tau_{ij}(t))$$

Moreover, let $\delta_{ij}(t) := t - \tau_{ij}(t)$ such that $\hat{\mathbf{z}}_0(\tau_{ij}(t)) = \hat{\mathbf{z}}_0(t - \delta_{ij}(t))$ and thus, $\hat{z}_{ij}(\tau_{ij}(t)) = \mathbf{b}_{ij}^\top \hat{\mathbf{z}}_0(t - \delta_{ij}(t))$. Note that $|\delta_{ij}(t)| \leq \Delta, \forall t \geq 0, \forall (i,j) \in \mathcal{E}$ by construction of the transaction instants. Now, use the fact that $\hat{z}_{ij}(\tau_{ij}(t)) = -\hat{z}_{ji}(\tau_{ij}(t))$ to write (8.1) in compact form as follows:

$$\begin{aligned}
\dot{\boldsymbol{\chi}}_\mu(t) &= k_\mu \theta^{\mu+1} \sum_{(i,j)\in\mathcal{E}} \mathbf{b}_{ij} \left\lceil \mathbf{b}_{ij}^\top \hat{\mathbf{z}}_0(t - \delta_{ij}(t)) \right\rfloor^{\frac{m-\mu}{m+1}} + \boldsymbol{\chi}_{\mu+1}(t) \\
&\quad - \gamma_\mu \boldsymbol{\chi}_\mu(t), \qquad \text{for } \mu \in \{0, \dots, m-1\} \\
\dot{\boldsymbol{\chi}}_m(t) &= k_m \theta^{m+1} \sum_{(i,j)\in\mathcal{E}} \mathbf{b}_{ij} \left\lceil \mathbf{b}_{ij}^\top \hat{\mathbf{z}}_0(t - \delta_{ij}(t)) \right\rfloor^{0} - \gamma_m \boldsymbol{\chi}_m(t)
\end{aligned} \tag{8.6}$$

Consider the recursive sequence of filters

$$\mathbf{s}_0(t) = \mathbf{z}(t), \quad \mathbf{s}_{\mu+1}(t) = \left(\frac{\mathrm{d}}{\mathrm{d}t} + \gamma_\mu\right) \mathbf{s}_\mu(t) \tag{8.7}$$

for $\mu \in \{0, \dots, m\}$ with $\mathbf{z}(t) = [z_1(t), \dots, z_{\mathsf{N}}(t)]^\top$. Note that

$$\begin{aligned}
\mathbf{s}_{m+1}(t) &= \left(\frac{\mathrm{d}}{\mathrm{d}t} + \gamma_m\right) \cdots \left(\frac{\mathrm{d}}{\mathrm{d}t} + \gamma_0\right) \mathbf{z}(t) \\
&= \mathbf{z}^{(m+1)}(t) + \sum_{\mu=0}^{m} l_\mu \mathbf{z}^{(\mu)}(t)
\end{aligned}$$

where $l_0, \dots, l_m$ are defined in Assumption 8.1. Define $\mathbf{y}_\mu(t) = \mathbf{s}_\mu(t) - \boldsymbol{\chi}_\mu(t)$. We make a decomposition of $\mathbf{y}_\mu(t)$ in a consensus component

$$\bar{y}_\mu(t) = (\mathbb{1}^\top / \mathsf{N}) \mathbf{y}_\mu(t)$$

and a consensus error

$$\tilde{\mathbf{y}}_\mu(t) = \mathbf{P} \mathbf{y}_\mu(t)$$

with $\mathbf{P} = (\mathbf{I}_{\mathsf{N}} - \mathbb{1}\mathbb{1}^\top / \mathsf{N})$. Consequently,

$$\mathbf{y}_\mu(t) = \bar{y}_\mu(t) \mathbb{1} + \tilde{\mathbf{y}}_\mu(t).$$

In the following lemma, we provide characterization of the trajectories of $\bar{y}_\mu(t)$.

Lemma 8.9. *Let* $\bar{y}_\mu(t) = (\mathbb{1}^\top/\mathsf{N})\mathbf{y}_\mu(t)$ *with* $\mathbf{y}_\mu(t) = \mathbf{s}_\mu(t) - \boldsymbol{\chi}_\mu(t)$ *and* $\boldsymbol{\chi}_\mu(t)$ *subject to* (8.6). *Finally, define*

$$\zeta_\mu(t) = \sum_{\nu=0}^{m} \mathbf{G}_{\mu+1,\nu+1}\bar{y}_\nu(t) \tag{8.8}$$

Hence, $\lim_{t\to\infty} |\zeta_\mu(t) - \bar{z}^{(\mu)}(t)| = 0, \forall \mu \in \{0,\dots,m\}$.

PROOF: First, note that $\mathbb{1}^\top \mathbf{b}_{ij} = 0$ for any link $(i,j) \in \mathcal{E}$ such that $\mathbb{1}^\top \dot{\boldsymbol{\chi}}_\mu(t)$ contains only the linear terms in (8.6). Hence,

$$\begin{aligned}
\dot{\bar{y}}_\mu(t) &= (\mathbb{1}^\top/\mathsf{N})(\dot{\mathbf{s}}_\mu(t) - \dot{\boldsymbol{\chi}}_\mu(t)) \\
&= (\mathbb{1}^\top/\mathsf{N})(\mathbf{s}_{\mu+1}(t) - \gamma_\mu \mathbf{s}_\mu(t) - \boldsymbol{\chi}_{\mu+1}(t) + \gamma_\mu \boldsymbol{\chi}_\mu(t)) \\
&= \bar{y}_{\mu+1}(t) - \gamma_\mu \bar{y}_\mu(t)
\end{aligned}$$

for $\mu \in \{0,\dots,m-1\}$ and similarly

$$\dot{\bar{y}}_m(t) = (\mathbb{1}^\top/\mathsf{N})\mathbf{s}_{m+1}(t) - \gamma_m \bar{y}_m(t)$$

equivalently $\dot{\bar{\mathbf{y}}}(t) = \boldsymbol{\Gamma}\bar{\mathbf{y}}(t) + \mathbf{b}_{m+1}(\mathbb{1}^\top/\mathsf{N})\mathbf{s}_{m+1}(t)$ where $\bar{\mathbf{y}}(t) = [\bar{y}_0(t),\dots,\bar{y}_m(t)]^\top$. According to [2, Corollary 5] it follows that:

$$\mathbf{G}\boldsymbol{\Gamma}\mathbf{G}^{-1} = \begin{bmatrix} \mathbf{0}_{m\times 1} & \mathbf{I}_m \\ -l_0 & -[l_1,\dots,l_m] \end{bmatrix}, \mathbf{G}\mathbf{b}_{m+1} = \mathbf{b}_{m+1}$$

So that (8.8), transforms the dynamics of $\bar{y}_\mu(t)$ into the canonical form:

$$\begin{aligned}
\dot{\zeta}_\mu(t) &= \zeta_{\mu+1}(t) \\
\dot{\zeta}_m(t) &= (\mathbb{1}^\top/\mathsf{N})\mathbf{s}_{m+1}(t) - \sum_{\mu=0}^{m} l_\mu \zeta_\mu(t) \\
&= \bar{z}^{(m+1)}(t) - \sum_{\mu=0}^{m} l_\mu(\zeta_\mu(t) - \bar{z}^{(\mu)}(t))
\end{aligned}$$

where we used

$$\begin{aligned}
&(\mathbb{1}^\top/\mathsf{N})\mathbf{s}_{m+1}(t) \\
&= (\mathbb{1}^\top/\mathsf{N})\mathbf{z}^{(m+1)}(t)/\mathsf{N} + \sum_{\mu=0}^{m} l_\mu (\mathbb{1}^\top/\mathsf{N})\mathbf{z}^{(\mu)}(t) \\
&= \bar{z}^{(m+1)}(t) + \sum_{\mu=0}^{m} l_\mu \bar{z}^{(\mu)}(t)
\end{aligned}$$

Note that $\zeta_\mu(t) = \zeta_0^{(\mu)}(t)$ and define $\tilde{z}(t) = \zeta_0(t) - \bar{z}(t)$ to obtain

$$\tilde{z}^{(m+1)}(t) = -\sum_{\mu=0}^{m} l_\mu \tilde{z}^{(\mu)}(t)$$

which is a linear system with stable poles at $-\gamma_0,\dots,-\gamma_m$. Thus, $\tilde{z}^{(\mu)}(t) = \zeta_\mu(t) - \bar{z}^{(\mu)}(t)$ converge to the origin as $t \to \infty$. □

Now, we characterize the behaviour of the consensus error $\tilde{\mathbf{y}}_\mu(t)$. First, we obtain its dynamics according to (8.6). To simplify the exposition, denote with

$$\mathbf{f}_\mu(\hat{\mathbf{z}}_0;\boldsymbol{\delta}(t)) = k_\mu\theta^{\mu+1}\sum_{(i,j)\in\mathcal{E}}\mathbf{b}_{ij}\left\lceil \mathbf{b}_{ij}^\top\hat{\mathbf{z}}_0(t-\delta_{ij}(t))\right\rfloor^{\frac{m-\mu}{m+1}} \tag{8.9}$$

where $\boldsymbol{\delta}(t)\in\mathbb{R}^{|\mathcal{E}|}$ is a vector coining all the $\delta_{ij}(t),(i,j)\in\mathcal{E}$.

Lemma 8.10. *Let Assumption 8.1 hold and $\tilde{\mathbf{y}}_\mu(t)=\mathbf{P}\mathbf{y}_\mu(t)$ with $\mathbf{P}=(\mathbf{I}_\mathsf{N}-\mathbb{1}\mathbb{1}^\top/\mathsf{N})$, $\mathbf{y}_\mu(t)=\mathbf{s}_\mu(t)-\boldsymbol{\chi}_\mu(t)$ and $\boldsymbol{\chi}_\mu(t)$ subject to (8.6). Hence, $\tilde{\mathbf{y}}_\mu(t)$ satisfies the following differential inclusion:*

$$\begin{aligned}\dot{\tilde{\mathbf{y}}}_\mu(t)&\in-\mathbf{f}_\mu(\tilde{\mathbf{y}}_0;[0,\Delta]^{|\mathcal{E}|})+\tilde{\mathbf{y}}_{\mu+1}(t)-\gamma_\mu\tilde{\mathbf{y}}_\mu(t)\\ &\qquad\text{for }\mu\in\{0,\dots,m-1\}\\ \dot{\tilde{\mathbf{y}}}_m(t)&\in-\mathbf{f}_m(\tilde{\mathbf{y}}_0;[0,\Delta]^{|\mathcal{E}|})-\gamma_m\tilde{\mathbf{y}}_m(t)+[-L,L]^\mathsf{N}\end{aligned} \tag{8.10}$$

with set valued function $\mathbf{f}_\mu(\tilde{\mathbf{y}}_0;[0,\Delta]^{|\mathcal{E}|}):=\{\mathbf{f}\in\mathbb{R}^\mathsf{N}:\mathbf{f}=\mathbf{f}_\mu(\tilde{\mathbf{y}}_0;\boldsymbol{\delta}),\forall\boldsymbol{\delta}\in[0,\Delta]^{|\mathcal{E}|}\}$.

Proof: First, note that by definition, $\hat{\mathbf{z}}_0(t)=\mathbf{y}_0(t)$ due to (8.7) so that $\mathbf{f}_\mu(\hat{\mathbf{z}}_0;\boldsymbol{\delta}(t))=\mathbf{f}_\mu(\mathbf{y}_0;\boldsymbol{\delta}(t))$. Moreover, note that

$$\mathbf{P}\mathbf{b}_{ij}=(\mathbf{I}_\mathsf{N}+(1/\mathsf{N}\mathbb{1}\mathbb{1}^\top))\mathbf{b}_{ij}=\mathbf{b}_{ij}$$

since $\mathbb{1}^\top\mathbf{b}_{ij}=0$ and for the same reason $\mathbf{b}_{ij}^\top\mathbf{z}_0(t)=\mathbf{b}_{ij}^\top\tilde{\mathbf{y}}_0(t)$. Hence, $\mathbf{P}\mathbf{f}_\mu(\mathbf{z}_0;\boldsymbol{\delta}(t))=\mathbf{f}_\mu(\tilde{\mathbf{y}}_0;\boldsymbol{\delta}(t))$. Now, compute the dynamics of $\tilde{\mathbf{y}}_\mu(t)$ as

$$\begin{aligned}\dot{\tilde{\mathbf{y}}}_\mu(t)&=\mathbf{P}\dot{\mathbf{s}}_\mu(t)-\mathbf{P}\mathbf{f}_\mu(\tilde{\mathbf{y}}_0;\boldsymbol{\delta}(t))-\mathbf{P}\boldsymbol{\chi}_{\mu+1}(t)+\gamma_\mu\mathbf{P}\boldsymbol{\chi}_\mu(t)\\ &=-\mathbf{f}_\mu(\tilde{\mathbf{y}}_0;\boldsymbol{\delta}(t))+\mathbf{P}(\mathbf{s}_{\mu+1}(t)-\boldsymbol{\chi}_{\mu+1}(t))\\ &\quad-\gamma_\mu\mathbf{P}(\mathbf{s}_\mu(t)-\boldsymbol{\chi}_\mu(t))\\ &=-\mathbf{f}_\mu(\tilde{\mathbf{y}}_0;\boldsymbol{\delta}(t))+\tilde{\mathbf{y}}_{\mu+1}(t)-\gamma_\mu\tilde{\mathbf{y}}_\mu(t)\\ \dot{\tilde{\mathbf{y}}}_m(t)&=-\mathbf{f}_m(\tilde{\mathbf{y}}_0;\boldsymbol{\delta}(t))-\gamma_m\tilde{\mathbf{y}}_m(t)-\mathbf{P}\mathbf{s}_{m+1}(t)\end{aligned} \tag{8.11}$$

The inclusion (8.10) follows from (8.11) by the definition of $\mathbf{f}_\mu(\tilde{\mathbf{y}}_0;[0,\Delta]^{|\mathcal{E}|})$ and by noting that $\mathbf{P}\mathbf{s}_{m+1}(t)\in[-L,L]^\mathsf{N}$ due to Assumption 8.1. □

The following lemmas derive some important properties of the inclusion (8.10).

Lemma 8.11. *Let $\mathcal{G}$ be connected and set $\Delta=0$ in the inclusion (8.10). Then, there exists $T>0$, sufficiently big θ, $\{k_\mu\}_{\mu=0}^m$, and admissible balls $\tilde{\mathcal{R}}_0,\dots,\tilde{\mathcal{R}}_m\subset\mathbb{R}^\mathsf{N}$ containing the origin such that if $\tilde{\mathbf{y}}_\mu(0)\in\tilde{\mathcal{R}}_\mu$ for all $\mu\in\{0,\dots,m\}$, then $\|\tilde{\mathbf{y}}_\mu(t)\|=0,\forall t\geq T$ for all $\mu\in\{0,\dots,m\}$. Moreover, the balls $\tilde{\mathcal{R}}_0,\dots,\tilde{\mathcal{R}}_m$ can be made arbitrarily big by increasing $\theta>0$.*

Proof: The proof follows by noticing that with $\Delta=0$ then

$$\begin{aligned}\mathbf{f}_\mu(\hat{\mathbf{z}}_0;\mathbf{0}_{\mathsf{N}\times1})&=k_\mu\theta^{\mu+1}\sum_{(i,j)\in\mathcal{E}}\mathbf{b}_{ij}\left\lceil\mathbf{b}_{ij}^\top\hat{\mathbf{z}}_0(t)\right\rfloor^{\frac{m-\mu}{m+1}}\\ &=k_\mu\theta^{\mu+1}\mathbf{D}\left\lceil\mathbf{D}^\top\hat{\mathbf{z}}_0(t)\right\rfloor^{\frac{m-\mu}{m+1}}\end{aligned}$$

where $\mathbf{D}$ is the adjacency matrix of $\mathcal{G}$ with columns given by $\mathbf{b}_{ij}$. Hence, with $\Delta = 0$, (8.10) is equivalent to (6.11) in Chapter 6 without noise and with equal agent weights. The result follows from Lemma 6.7 in Chapter 6 for such special case. □

Lemma 8.12. *Let $\theta, \{k_\mu\}_{\mu=0}^m$ be chosen such that (8.10) is finite time stable with $\Delta = 0$ and any $\gamma_0, \dots, \gamma_m \in [0, 1)$. Then, for every $\Delta_{\max} > 0$ there exists $\gamma_{\max} > 0$ such that if $\gamma_0, \dots, \gamma_m \in [0, \gamma_{\max})$ and $\Delta \in [0, \Delta_{\max})$ then, there exist $T, \tilde{c}_0, \dots, \tilde{c}_m > 0$ such that $\|\tilde{\mathbf{y}}_\mu(t)\| \leq \tilde{c}_\mu \Delta^{m-\mu+1}, \forall t \geq T$.*

Proof: Note that for fixed $\gamma = [\gamma_0, \dots, \gamma_m]^\top \in [0, 1)^{m+1}$, existence of sufficiently big parameters $\theta^\gamma, \{k_\mu^\gamma\}_{\mu=0}^m$ such that (8.10) is finite time stable with $\Delta = 0$ is guaranteed by Lemma 8.11. Hence, the parameters $\sup_\gamma \theta^\gamma, \{\sup_\gamma k_\mu^\gamma\}_{\mu=0}^m$ exist and ensure finite time stability of (8.10) for $\Delta = 0$ and any $\gamma_0, \dots, \gamma_m \in [0, 1)$. Now, pick some $\Delta' > 0$ and let $\eta = \Delta'/\Delta$. Then, set $\gamma_{\max} = \Delta'/\Delta_{\max}$. Consider the change of coordinates

$$t' = \eta t, \quad \tilde{\mathbf{y}}'_\mu(t') = \eta^{m-\mu+1} \tilde{\mathbf{y}}_\mu(t'/\eta)$$

Now, since $\Delta' = \eta\Delta$, and $\tilde{\mathbf{y}}'_0(t') = \tilde{\mathbf{y}}'_0(\eta t) = \eta^{m+1}\tilde{\mathbf{y}}_0(t)$ then

$$\begin{aligned}
\mathbf{f}_\mu(\tilde{\mathbf{y}}_0; [0, \Delta]^{|\mathcal{E}|}) &= k_\mu \theta^{\mu+1} \sum_{(i,j)\in\mathcal{E}} \mathbf{b}_{ij} \left\lceil \mathbf{b}_{ij}^\top \tilde{\mathbf{y}}_0(t - [0, \Delta]) \right\rfloor^{\frac{m-\mu}{m+1}} \\
&= k_\mu \theta^{\mu+1} \sum_{(i,j)\in\mathcal{E}} \mathbf{b}_{ij} \left\lceil \mathbf{b}_{ij}^\top \eta^{-m-1} \tilde{\mathbf{y}}'_0(\eta t - \eta[0, \Delta]) \right\rfloor^{\frac{m-\mu}{m+1}} \\
&= \eta^{\mu-m} k_\mu \theta^{\mu+1} \sum_{(i,j)\in\mathcal{E}} \mathbf{b}_{ij} \left\lceil \mathbf{b}_{ij}^\top \tilde{\mathbf{y}}'_0(t' - [0, \Delta']) \right\rfloor^{\frac{m-\mu}{m+1}} \\
&= \eta^{\mu-m} \mathbf{f}_\mu(\tilde{\mathbf{y}}'_0; [0, \Delta']^{|\mathcal{E}|})
\end{aligned}$$

Henceforth,

$$\begin{aligned}
\frac{\mathrm{d}\tilde{\mathbf{y}}'_\mu}{\mathrm{d}t'} &= \eta^{-1} \dot{\tilde{\mathbf{y}}}'_\mu = \eta^{m-\mu} \dot{\tilde{\mathbf{y}}}_\mu \\
&\in \eta^{m-\mu} \Big(-\mathbf{f}_\mu(\tilde{\mathbf{y}}_0; [0, \Delta]^{|\mathcal{E}|}) + \tilde{\mathbf{y}}_{\mu+1}(t) - \gamma_\mu \tilde{\mathbf{y}}_\mu(t) \Big) \\
&= \eta^{m-\mu} \big(- \eta^{\mu-m} \mathbf{f}_\mu(\tilde{\mathbf{y}}'_0; [0, \Delta']^{|\mathcal{E}|}) \\
&\quad + \eta^{\mu-m} \tilde{\mathbf{y}}'_{\mu+1}(t') - \eta^{\mu-m-1} \gamma_\mu \tilde{\mathbf{y}}'_\mu(t') \big) \\
&= -\mathbf{f}_\mu(\tilde{\mathbf{y}}'_0; [0, \Delta']^{|\mathcal{E}|}) + \tilde{\mathbf{y}}'_{\mu+1}(t') - (\gamma_\mu/\eta) \tilde{\mathbf{y}}'_\mu(t') \\
\frac{\mathrm{d}\tilde{\mathbf{y}}'_m}{\mathrm{d}t'} &\in -\mathbf{f}_m(\tilde{\mathbf{y}}'_0; [0, \Delta']^{|\mathcal{E}|}) - (\gamma_m/\eta) \tilde{\mathbf{y}}'_m(t') + [-L, L]^{\mathsf{N}}
\end{aligned}$$

Therefore, the dynamics of $\tilde{\mathbf{y}}'_\mu(t')$ are equivalent to the ones in (8.10) with the new Δ' and the new gains $\gamma_0/\eta, \dots, \gamma_m/\eta < \gamma_{\max}/\eta = \Delta/\Delta_{\max} \leq 1$. Picking $\Delta' = 0$ for continuous-time communication, then by assumption $\tilde{\mathbf{y}}'_\mu(t')$ is finite time stable towards the origin since all $\gamma_\mu/\eta \in (0, 1)$. Hence, there exist sufficiently small $\Delta' > 0$ and constants $T', c'_0, \dots, c'_\mu$ such that $\|\tilde{\mathbf{y}}'_\mu(t')\| \leq c'_\mu, \forall t' \geq T'$ by continuity of solutions [174, Page 87], similarly as argued in [175]. This means that there exists $T = T'/\eta > 0$ such that by setting $\tilde{c}_\mu = c'_\mu (\Delta')^{\mu-m+1}$ and using $\|\tilde{\mathbf{y}}'_\mu(t')\| \leq c'_\mu$, $\eta = \Delta'/\Delta$ we have that (8.10) complies the following $\forall t = t'/\eta \geq T = T'/\eta$:

$$\|\tilde{\mathbf{y}}_\mu(t)\| = \eta^{\mu-m-1} \|\tilde{\mathbf{y}}'_\mu(t')\| \leq \tilde{c}_\mu \Delta^{m-\mu+1}$$

completing the proof. □

Lemma 8.13. *Let fixed $\tilde{c}_0, \dots, \tilde{c}_m > 0$ and define*

$$I_\mu(\Delta) := \sum_{\nu=0}^{m} \tilde{c}_\mu \mathbf{G}_{\mu+1,\nu+1} \Delta^{m-\nu+1}$$

Then, for any $\Delta_{\max} > 0$ there exists $c_\mu > 0$ such that $\forall \Delta \in [0, \Delta_{\max}]$ it follows that $I_\mu(\Delta) \leq c_\mu \Delta^{m-\mu+1}$.

Proof: First, note that $\mathbf{G}$ is a lower triangular matrix with only ones in the diagonal. Hence,

$$\begin{aligned} I_\mu(\Delta) &= \tilde{c}_\mu \Delta^{m-\mu+1} + \sum_{\nu=0}^{\mu-1} \tilde{c}_\nu \mathbf{G}_{\mu+1,\nu+1} \Delta^{m-\nu+1} \\ &\leq \Delta^{m-\mu+1} \left(\tilde{c}_\mu + \sum_{\nu=0}^{\mu-1} \tilde{c}_\nu \mathbf{G}_{\mu+1,\nu+1} \Delta^{\mu-\nu} \right) \end{aligned}$$

where $\mathbf{G}_{\mu+1,\mu+1} = 1$ and $\mathbf{G}_{\mu+1,\nu+1} = 0, \nu > \mu$ were used. Thus, since $\Delta \in [0, \Delta_{\max}]$ and $\mu > \nu$ then, the result follows with $c_\mu = \left(\tilde{c}_\mu + \sum_{\nu=0}^{\mu-1} \tilde{c}_\nu \mathbf{G}_{\mu+1,\nu+1} \Delta_{\max}^{\mu-\nu} \right) > 0$ □

8.4.2 Convergence under nominal asynchronous communication

In this section, we provide a proof of Theorem 8.2, using results developed in the previous section to show convergence of (8.1) without communication delays or sampled inputs. Recall the decomposition $\mathbf{y}_\mu(t) = \bar{y}_\mu(t)\mathbb{1} + \tilde{\mathbf{y}}_\mu(t)$. Recall from Lemma 8.10 that $\tilde{\mathbf{y}}_\mu(t)$ satisfies the inclusion (8.10). Hence, given $\Delta_{\max} > 0$ pick $\gamma_{\max} > 0$ and gains as in Lemma 8.12 to conclude that $\|\tilde{\mathbf{y}}_\mu(t)\| \leq \tilde{c}_\mu \Delta^{m-\mu+1}, \forall t \geq T$ for some $T, \tilde{c}_0, \dots, \tilde{c}_m > 0$. Recall that

$$\mathbf{z}^{(\mu)}(t) = \sum_{\nu=0}^{m} \mathbf{G}_{\mu+1,\nu+1} \mathbf{s}_\mu(t)$$

is the inverse transformation of (8.7) such that $\hat{\mathbf{z}}_\mu(t)$ defined from (8.1) comply

$$\hat{\mathbf{z}}_\mu(t) = \mathbf{z}^{(\mu)}(t) - \sum_{\nu=0}^{m} \mathbf{G}_{\mu+1,\nu+1} \boldsymbol{\chi}_\nu(t) \quad = \sum_{\nu=0}^{m} \mathbf{G}_{\mu+1,\nu+1} \mathbf{y}_\nu(t)$$

since $\mathbf{y}_\nu(t) = \mathbf{s}_\nu(t) - \boldsymbol{\chi}_\nu(t)$. Hence, $\mathbf{P}\hat{\mathbf{z}}_\mu(t)$ complies

$$\|\mathbf{P}\hat{\mathbf{z}}_\mu(t)\| \leq 2 \sum_{\nu=0}^{m} \mathbf{G}_{\mu+1,\nu+1} \|\tilde{\mathbf{y}}_\nu(t)\| \leq 2 \sum_{\nu=0}^{m} \tilde{c}_\mu \mathbf{G}_{\mu+1,\nu+1} \Delta^{m-\nu+1} = I_\mu(\Delta)$$

with $I_\mu(\Delta)$ defined in Lemma 8.13. Thus, $\|\mathbf{P}\hat{\mathbf{z}}_\mu(t)\| \leq c_\mu \Delta^{m-\mu+1}$ with c_μ as in Lemma 8.13. On the other hand, we have that

$$(\mathbb{1}^\top / \mathsf{N}) \hat{\mathbf{z}}_\mu(t) = \sum_{\nu=0}^{m} \mathbf{G}_{\mu+1,\nu+1} \bar{y}_\nu(t) = \zeta_\mu(t)$$

with $\zeta_\mu(t)$ defined in (8.8). Therefore, $(\mathbb{1}^\top/\mathsf{N})\hat{\mathbf{z}}_\mu(t) = \zeta_\mu(t)$ converge towards $\bar{z}^{(\mu)}(t)$ as $t \to \infty$ according to Lemma 8.9. Consider the decomposition $\hat{\mathbf{z}}_\mu(t) = (\mathbb{1}^\top/\mathsf{N})\hat{\mathbf{z}}_\mu(t)\mathbb{1} + \mathbf{P}\hat{\mathbf{z}}_\mu(t) = \zeta^{(\mu)}(t)\mathbb{1} + \mathbf{P}\hat{\mathbf{z}}_\mu(t)$ with $\zeta(t) := \zeta_0(t)$ and $\zeta^{(\mu)}(t) = \zeta_\mu(t)$ as a result from the proof of Lemma 8.9. Hence,

$$\|\hat{\mathbf{z}}_\mu(t) - \zeta^{(\mu)}(t)\mathbb{1}\| = \|\mathbf{P}\hat{\mathbf{z}}_\mu(t)\| \leq c_\mu \Delta^{m-\mu+1}$$

which completes the proof for item 1 of the Theorem. Items 2 and 3 follow from Lemmas 8.9 and 8.11 respectively.

8.4.3 Non-nominal asynchronous communication

In this Section, we provide the proofs for Corollaries 8.6 and 8.7 for the cases of non-cooperative sampling and symmetric communication delays respectively.

Proof of Corollary 8.6 (non-cooperative sampling): Similarly as in the proof of Theorem 8.2, we write (8.1) in compact form with the new output (8.2). Recall the definition of $\hat{\mathbf{z}}_0(t) = \mathbf{z}(t) - \boldsymbol{\chi}_0(t)$ from Section 8.4.1. Hence, for arbitrary link (i, j), it follows that $\hat{z}_i(t) = \mathbf{b}_i^\top \hat{\mathbf{z}}_0(\tau_{i,k}), \forall t \in [\tau_{i,k}, \tau_{i,k+1})$ and therefore, $\hat{z}_{ij}(\tau_{ij}(t)) = \mathbf{b}_i^\top \hat{\mathbf{z}}_0(\tau_{ii}(t)) - \mathbf{b}_j^\top \hat{\mathbf{z}}_0(\tau_{jj}(t))$ recalling that by construction $\tau_{ii}(t) = \max\{\tau \in \boldsymbol{\tau}_i | \tau \leq t\}$ is the last sampling instant of agent i before t, similarly for agent j. Let $\delta_i(t) := t - \tau_{ii}(t)$ such that $\hat{\mathbf{z}}_0(\tau_{ii}(t)) = \hat{\mathbf{z}}_0(t - \delta_i(t))$ and,

$$\hat{z}_{ij}(\tau_{ij}(t)) = \mathbf{b}_i^\top \hat{\mathbf{z}}_0(t - \delta_i(t)) - \mathbf{b}_j^\top \hat{\mathbf{z}}_0(t - \delta_j(t))$$

Note that $|\delta_i(t)| \leq \Delta, \forall t \geq 0, \forall i \in \{1, \ldots, \mathsf{N}\}$ by construction. Define

$$\boldsymbol{\delta}(t) = [\delta_i(t), \ldots, \delta_i(t)]^\top \in \mathbb{R}^{\mathsf{N}}.$$

Now, use the fact that $\hat{z}_{ij}(\tau_{ij}(t)) = -\hat{z}_{ji}(\tau_{ij}(t))$ to write (8.1) as follows:

$$\begin{aligned}
\dot{\boldsymbol{\chi}}_\mu(t) &= \mathbf{f}_\mu(\mathbf{z}_0; \boldsymbol{\delta}(t)) + \boldsymbol{\chi}_{\mu+1}(t) - \gamma_\mu \boldsymbol{\chi}_\mu(t) \quad \text{for } \mu \in \{0, \ldots, m-1\} \\
\dot{\boldsymbol{\chi}}_m(t) &= \mathbf{f}_m(\mathbf{z}_0; \boldsymbol{\delta}(t)) - \gamma_m \boldsymbol{\chi}_m(t)
\end{aligned}$$

where we redefined $\mathbf{f}_\mu(\mathbf{z}_0; \boldsymbol{\delta}(t))$ in (8.9) as

$$\mathbf{f}_\mu(\hat{\mathbf{z}}_0; \boldsymbol{\delta}(t)) = k_\mu \theta^{\mu+1} \sum_{(i,j)\in\mathcal{E}} \mathbf{b}_{ij} \left\lceil \mathbf{b}_i^\top \hat{\mathbf{z}}_0(t - \delta_i(t)) - \mathbf{b}_j^\top \hat{\mathbf{z}}_0(t - \delta_j(t)) \right\rfloor^{\frac{m-\mu}{m+1}}$$

Note that $\mathbb{1}^\top \mathbf{f}_\mu(\hat{\mathbf{z}}_0; \boldsymbol{\delta}(t)) = 0$ such that Lemma 8.9 is still valid. An equivalent version of Lemma 8.10 follows, with the new $\mathbf{f}_\mu(\hat{\mathbf{z}}_0; \boldsymbol{\delta}(t))$ instead of $\mathbf{f}_\mu(\tilde{\mathbf{y}}_0; \boldsymbol{\delta}(t))$. Moreover, note that with $\Delta = 0$, then $\mathbf{b}_i^\top \hat{\mathbf{z}}_0(t - \delta_i(t)) - \mathbf{b}_j^\top \hat{\mathbf{z}}_0(t - \delta_j(t)) = \mathbf{b}_i^\top \hat{\mathbf{z}}_0(t) - \mathbf{b}_j^\top \hat{\mathbf{z}}_0(t) = \mathbf{b}_{ij}^\top \hat{\mathbf{z}}_0(t)$. Hence, the continuity of solutions argument in Lemma 8.11 applies as well. Therefore, the proof of Theorem 8.2 in Section 8.4.2 follows in the same way, completing the proof of the Corollary.

Proof of Corollary 8.7 (symmetric delays): Similarly as in Section 8.4.1, we can write $\hat{z}_{ij}(\tau_{ij}(t)) = \mathbf{b}_{ij}^\top \hat{\mathbf{z}}_0(\tau_{ij}(t) - d_{ij}(t))$. Now, let $\delta_{ij}(t) := d_{ij}(t) + (t - \tau_{ij}(t))$ such that we can write $\hat{z}_{ij}(\tau_{ij}(t)) = \hat{z}_{ij}(t - \delta_{ij}(t))$. The new delay signal satisfies $|\delta_{ij}(t)| \leq d + \Delta, \forall t \geq 0$. Hence, the rest of the reasoning in Section 8.4.1 follows equivalently by replacing Δ with $\Delta + d$, completing the proof.

8.5 NUMERICAL EXPERIMENTS

In order to test the proposal, we consider the following scenario. First, we set the number of agents in the network as $\mathsf{N} = 10$ and the communication network to be modeled by the graph $\mathcal{G}$ shown in Figure 8.4. In addition, we set the dynamic reference signals to $z_i(t) = A_i \cos(\omega_i t + \phi_i)$ where

$$\{A_i\}_{i=1}^{\mathsf{N}} = \{0.4, 0.96, 0.17, 0.12, 0.13, 0.5, 0.02, 0.9, 0.8, 0.01\}$$
$$\{\omega_i\}_{i=1}^{\mathsf{N}} = \{0.95, 0.55, 0.9, 0.64, 0.39, 0.5, 0.6, 0.54, 0.92, 0.9\}$$
$$\{\phi_i\}_{i=1}^{\mathsf{N}} = \{0.17, 0.3, 0.13, 0.8, 0.34, 0.94, 0.6, 0.87, 0.84, 0.9\}$$

Note that the signals $\{z_i(t)\}_{i=1}^{\mathsf{N}}$ are persistently varying for all $t \geq 0$ and satisfy Assumption 8.1 for any order m provided a suitable choice of L. In this case, for illustrative purposes, we choose $m = 3$ and $\gamma_0 = \cdots = \gamma_3 = 1$ such that Assumption 8.1 is complied with $L = 2.43$. As evident from the proof of Theorem 8.2, the parameters of (8.1) are related to the ones of the REDCHO protocol. Hence, we choose $\theta = 1$ and $\{k_\mu\}_{\mu=0}^{m} = \{16.87, 24.71, 12.16, 2.44\}$ according to the parameter design rules provided in [1, 2]. In addition, for the sake of generality, initial conditions $\{\chi_{i,\mu}(0)\}_{\mu=0}^{m}$ where drawn randomly from a uniform distribution over $[-10, 10]$.

The transaction instant sequences $\boldsymbol{\tau}_i$ where generated by setting $\tau_{i,k+1} = \delta_{i,k} + \tau_{i,k}$ where $\delta_{i,k}$ was drawn randomly from a uniform distribution over $[0, \Delta]$. Hence, $\boldsymbol{\tau}_i$ is an asynchronous sequence of time instants, different between agents. In addition, we enforce communication delays as in Figure 8.3, all fixed at the maximum delay size d. Therefore, we consider three possible degrees of freedom to stress the proposal. First, by varying the maximum time step Δ. Second, by varying the maximum delay d. Third, by using non-cooperative sampling as described in Section 8.2.1.

As a baseline, we first consider $d = 0$, a small time step $\Delta = 10^{-5}$ and cooperative sampling. Figure 8.5 shows the behaviour of estimation outputs $\{\hat{z}_i(t)\}_{i=1}^{\mathsf{N}}$ for the average signal $\bar{z}(t)$. As expected from items **i**) and **ii**) of Theorem 8.2, it can be observed that the outputs reach consensus in finite time towards a neighborhood of a signal $\zeta(t)$, which reaches $\bar{z}(t)$ asymptotically.

In addition, we use this example to introduce the two main approaches for which we compare. To the best of our knowledge, no dynamic consensus protocols exist in the literature designed for the scenario considered in this chapter. However, two modifications of (8.1) can yield algorithms which are closely related to other works in the literature. First, we compare with a linear analog of our proposal, using $F_\mu(\bullet) = (\bullet)$ in (8.1). This is very similar to the protocols in [67, 98, 171], but tailored to the asynchronous setting considered in this chapter. Second, we compare with (8.1) using $m = 0$, which corresponds to a FOSM protocol. This is equivalent to the protocols used in [85, 160]. Figure 8.6 shows the consensus error in each case. Note that our proposal and the FOSM protocol outperform the linear protocol regardless of the persistent dynamics of the reference signals. On the other hand, the linear protocol can only ensure bounded steady-state error, which depends on how the reference signals are changing.

We now test the algorithm with a larger value for the maximum time-step, specifically $\Delta = 3 \times 10^{-3}$. Figure 8.7 shows the total error of the protocol's outputs with respect to the average signal $\bar{z}(t)$ in this case. It can be observed that the performance of our proposal is degraded as Δ is increased when compared to smaller values of Δ. However,

our proposal still outperforms FOSM and linear protocols. Note that, contrary to the FOSM protocol which degrades very quickly by increasing Δ, the linear protocol produces more consistent results with different values of Δ.

In order to test the protocols under non-nominal asynchronous communication, we start by considering non-cooperative sampling and $\Delta = 3 \times 10^{-3}$. The results are shown in Figure 8.8, where it can be observed that the performance degrades when compared to the results in Figure 8.7 with cooperative sampling. In Figure 8.9, we use $\Delta = d = 3{\times}10^{-3}$ to test the performance of the protocols under symmetric communication delays. It can be observed that the performance degrades with respect to the case of $\Delta = 3 \times 10^{-3}$ without delay. In both Figure 8.8 and Figure 8.9 the robustness of our method is emphasized when compared with the FOSM and linear approaches since it outperforms the rest of the protocols. Figure 8.10 showcases the versatility of our proposal, (8.1), under various configurations of Δ, d and cooperative or non-cooperative sampling. All scenarios are shown in the same scale for easy comparison. It's worth noting that all results are obtained using the same parameters in (8.1). However, Theorem 8.2 suggests that the method's accuracy can be adjusted by modifying the gains $\{k_\mu, \gamma_\mu\}_{\mu=0}^{m}$ for a particular application due to their influence in the constants $\{c_\mu\}_{\mu=0}^{m}$.

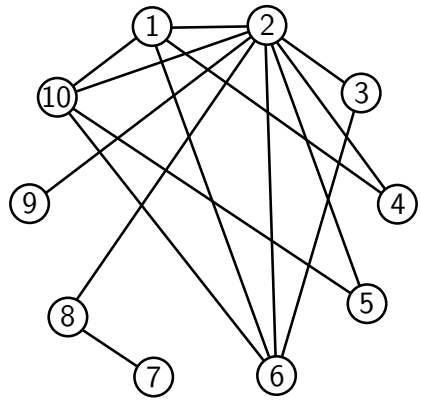

Figure 8.4: Undirected graph $\mathcal{G}$ used in the experiments of Section 8.5.

8.6 DISCUSSION

In this chapter, we have proposed a distributed algorithm for computing the dynamic average of a set of time-varying reference signals distributed across a communication network. The proposed algorithm ensures convergence even under asynchronous communication. Furthermore, we studied the robustness of the algorithm with respect to symmetric communication delays and sampled reference signals. The proposal is based on the HOSM framework, which guarantees EDC when continuous-time communication is used, yielding quasi-exact performance. When discrete-time communication is used, reduced chattering is observed in the proposed algorithm. We provided a formal convergence analysis and several numerical experiments to confirm the advantages of the proposal.

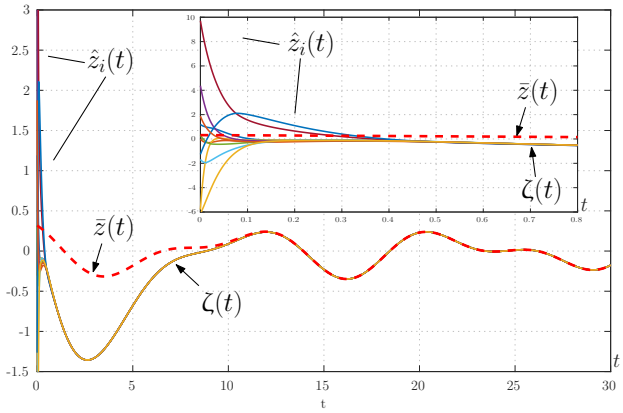

Figure 8.5: Baseline trajectories for the outputs $\hat{z}_i(t)$ of algorithm (8.1) for the scenario described in Section 8.5. The configuration of (8.1) has $\Delta = 10^{-5}, d = 0$ and cooperative sampling. At around $t \approx 0.7$, all outputs agree on the signal $\zeta(t)$, which approaches a small neighborhood around $\bar{z}(t)$ over time. The neighborhood size is roughly 10^{-4}, which is better appreciated in Figure 8.6. The vertical axis is shown in the range $[-1.5, 3]$ to emphasize the behaviour of $\bar{z}(t)$. Additionally, a closer look at the consensus transient is provided for the time interval $t \in [0, 0.8]$ with a vertical axis range of $[-6, 10]$.

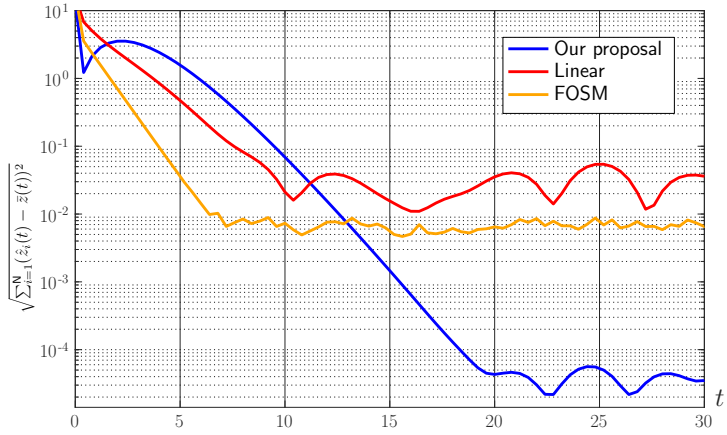

Figure 8.6: Total errors for the outputs $\hat{z}_i(t)$ of algorithm (8.1) when compared to $\bar{z}(t)$ for the scenario described in Section 8.5. The configuration of (8.1) has $\Delta = 10^{-5}, d = 0$ and cooperative sampling. Our proposal performs the best in this case whereas the linear protocol has the worse performance due to its steady-state error.

The results in this chapter, when combined with those in Chapter 7, lead to an architecture that closely resembles the goal presented in Figure 1.4 in Chapter 1. This integration provides a solid foundation for tackling formation control of multiple robots under perception-latency and resource trade-offs.

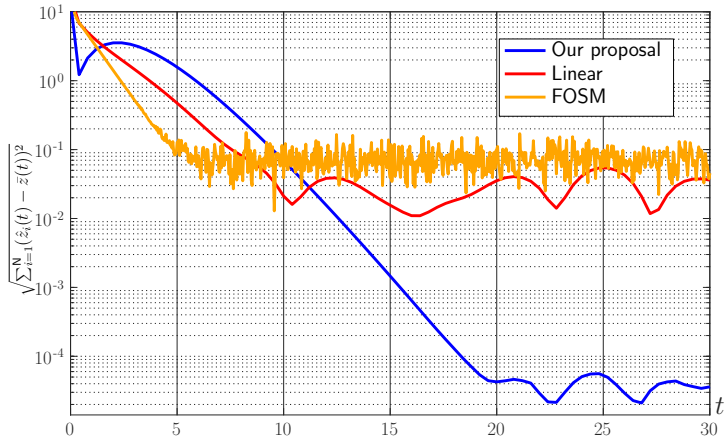

Figure 8.7: Total errors for the outputs $\hat{z}_i(t)$ of algorithm (8.1) when compared to $\bar{z}(t)$ for the scenario described in Section 8.5. The configuration of (8.1) has $\Delta = 3 \times 10^{-3}, d = 0$ and cooperative sampling. Our proposal performs the best in this case whereas the FOSM has the worse performance due to chattering.

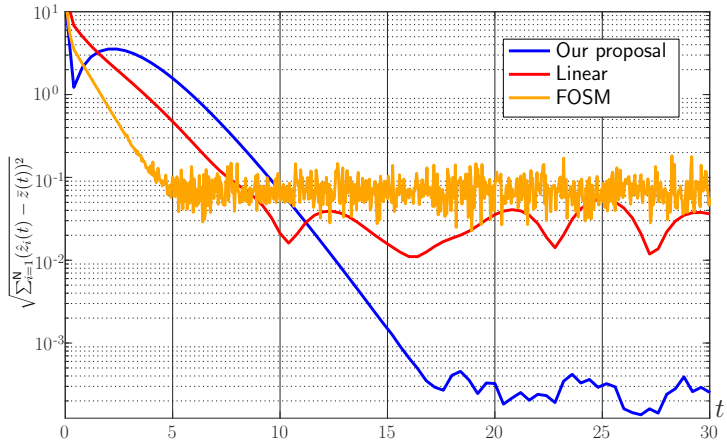

Figure 8.8: Total errors for the outputs $\hat{z}_i(t)$ of algorithm (8.1) when compared to $\bar{z}(t)$ for the scenario described in Section 8.5. The configuration of (8.1) has $\Delta = 3 \times 10^{-3}, d = 0$ and non-cooperative sampling. The performance of our proposal and the FOSM protocol is degraded with respect to the case with cooperating sampling whereas the linear protocol maintains a similar performance.

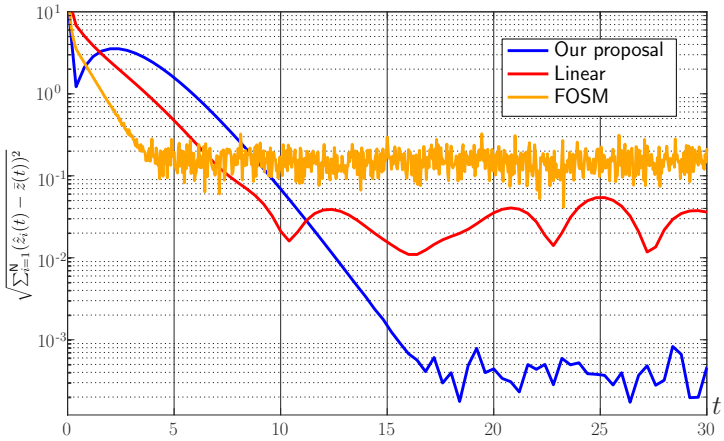

Figure 8.9: Total errors for the outputs $\hat{z}_i(t)$ of algorithm (8.1) when compared to $\bar{z}(t)$ for the scenario described in Section 8.5. The configuration of (8.1) has $\Delta = d = 3 \times 10^{-3}$ and cooperative sampling. The performance of our proposal and the FOSM protocol is degraded with respect to the case with cooperative sampling and no delay, whereas the linear protocol maintains a similar performance.

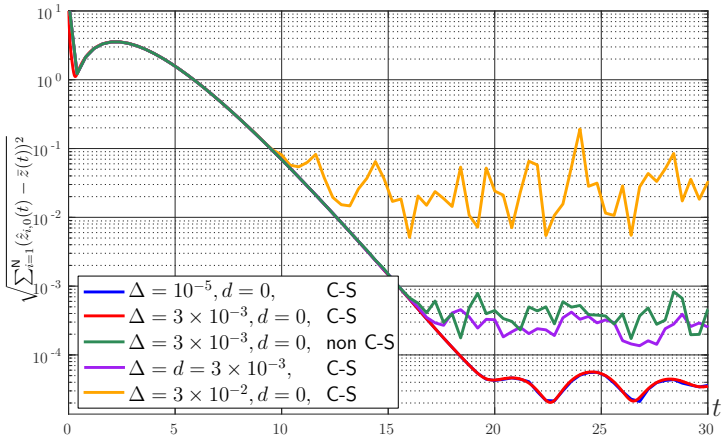

Figure 8.10: Summary of the accuracy results for the outputs $\hat{z}_i(t)$ of algorithm (8.1) when compared to $\bar{z}(t)$ with different values of Δ, d and where C-S stands for cooperative sampling.

Chapter Nine

Conclusions

In this thesis, we have explored various perception latency-aware solutions and consensus-based cooperation strategies for information fusion in multi-agent systems. These approaches are particularly relevant when multiple robots aim to track a target of interest using their onboard sensors while contending with resource constraints such as perception latency and network imperfections. Given the complexity of the overall problem, we adopted a modular strategy throughout this thesis. By studying perception latency scheduling problems independently of distributed consensus, we aimed to gain a deeper understanding of the specific challenges posed by each aspect.

Chapter 2 explored a control problem involving discrete-time measurements with perception latency and accuracy trade-offs. The objective was to develop perception-latency schedulers that optimize control performance and resource usage. However, we encountered significant challenges related to stability and complexity. Our main approach was to identify scheduling policy candidates that preserve stability, which led to an interesting connection between this problem and the stability of switching systems. While we made progress in ensuring stability, we discovered that finding the optimal stability-preserving schedule is an NP-hard problem, making it impractical to obtain optimal solutions. Despite this limitation, our framework demonstrated that stability is maintained for any schedule, opening possibilities for future research to explore more efficient sub-optimal algorithms. While the problem complexity remains a challenge, the insights gained from this chapter lay the groundwork for developing practical and effective perception-latency scheduling algorithms in control systems.

Chapter 3, PLATE, focuses on the complementary problem of state estimation for an unknown input system, such as a target of interest, using sensors with perception latency and accuracy trade-offs. As with Chapter 2, this problem also faces challenges related to the combinatorial nature of the solution space. In response, we proposed an efficient approximate solution to tackle the state estimation problem under perception latency constraints. The approach was validated using real-world data in a frame-skipping context, demonstrating its practical applicability and effectiveness.

In the domain of multi-agent cooperation, we have focused on dynamic average consensus techniques, which form the fundamental building blocks for information fusion in the later parts of this thesis. Our approach has presented results in increasing complexity,

considering various network constraints. Chapter 4 introduced EDCHO in the context of a static network topology, noiseless measurements, and continuous-time communication. This laid the groundwork for understanding average consensus in a simplified setting. Building on EDCHO, Chapter 5 presented REDCHO, an extension that accommodates connectivity changes, enabling the inclusion of new agents into the network dynamically. This extension allowed us to consider certain classes of OMAS systems and demonstrated greater flexibility in handling real-world scenarios. In Chapter 6, REDCHO was extended to account for measurement noise and compute weighted averages. This step towards incorporating more realistic factors represents a significant contribution, resulting in the first distributed differentiator introduced in the literature.

The methods presented in Chapters 4, 5, and 6 offer a notable departure from the linear consensus protocols commonly found in the existing literature. The key advantage of these techniques lies in the usage of nonlinear consensus algorithms, resulting in exact dynamic consensus without any steady-state error. In contrast, linear approaches often suffer from bounded steady-state errors, even when operating without noise and in continuous-time communication scenarios. The precise dynamic consensus features achieved by our methods are of particular significance, especially when considering the presence of noise, as investigated in Chapter 6, outperforming linear and other sliding mode protocols.

Our tools for achieving exact dynamic consensus are built upon high-order sliding modes, which are instrumental in enabling the exactness feature. However, it is important to acknowledge that these techniques are not immune to chattering, particularly when dealing with delays and discrete-time updates. These limitations are thoroughly quantified and analyzed in Chapter 8, demonstrating that our methods can still outperform state-of-the-art techniques in a wide range of scenarios, even under such challenging conditions. By rigorously evaluating the performance of our methods in the presence of delays and discrete-time updates, Chapter 8 sheds light on the robustness and effectiveness of our techniques.

Chapters 7 and 8 addressed the specific challenges of integrating perception latency and consensus in multi-agent systems. In Chapter 7, we explored a formation control scenario where the dynamics of the robots surpass the speed of the perception stage, making formation trajectory generation based on perception measurements non-trivial. To tackle this issue, we introduced a smooth output estimator, ensuring stability of the local controller for each robot. The smoothness property of the estimator also facilitated the use of REDCHO as an information fusion building block, as it requires sufficiently smooth time-varying inputs. This modular approach allowed us to leverage the advantages of both perception latency scheduling and consensus techniques in a unified framework.

A key assumption made in Chapter 7 was that the communication between agents occurs rapidly enough to be modeled in continuous-time. In real-world scenarios, communication delays and asynchronous communication are often unavoidable, requiring the consideration of these factors to achieve practical and robust multi-agent coordination. Chapter 8 addressed this limitation by handling asynchronous discrete-time communication directly. This is the final piece of the architecture presented in Figure 1.4 in Chapter 1, providing a comprehensive framework for cooperative multi-agent systems under perception latency and communication constraints.

Future work:
Although this thesis has made significant contributions to perception latency scheduling and consensus-based multi-robot cooperation, there are several opportunities for further research:

Improved noise, robot and target models: This thesis employed a simplified noise model for robot perception, assuming Gaussian distributions. While this approach may be pragmatic, it could lead to unrealistic outcomes. Hence, exploring more comprehensive noise models is worth considering. Moreover, the usage of linear state space systems for both robot and target models might restrict the applicability of the findings to specific types of robots. Extending the study to encompass more general nonlinear systems poses significant challenges, which might be explored in future research.

Imperfect scheduling: In this thesis, it was assumed that perception latency scheduling leads to a predictable sequence of sampling instants for each robot. However, in real-world scenarios, computing systems often run on complex operating systems, which could introduce additional jitter or imperfections to the commands due to scheduling with other tasks. These disturbances may be inevitable, emphasizing the need for further research on enhancing the robustness of the system to handle such challenges.

General switching topologies and directed graphs: In this thesis, the focus was primarily on static networks for EDCHO and isolated connectivity change events for REDCHO-based algorithms, providing basic robustness in OMAS. However, in low-resource networks with cheap radios, significant connectivity changes may occur during an experiment based on robot positions and proximity. Therefore, it becomes crucial to investigate robustness to general switching network topologies. Additionally, in scenarios where robots broadcast information rather than establishing formal bi-directional communication links with neighbors, analysis under directed communication graphs needs to be taken into account. Addressing these issues requires non-trivial approaches, opening up interesting avenues for future research.

Approximate optimization techniques: In Chapters 2 and 3, optimization algorithms were employed for perception latency scheduling. While these algorithms aimed to achieve performance guarantees such as stability, they could be conservative and not necessarily the most efficient solutions from an implementation point of view. To explore more efficient and less conservative algorithms, machine learning techniques can be employed. For instance, in Chapter 3, it was demonstrated that a sub-optimal solution could be obtained by discretizing the problem space and applying a classifier to determine the perception method for each region. This classifier might be trained using reinforcement learning on simulation examples or learning from a supervisor algorithm with performance guarantees. However, ensuring that these approximations maintain similar performance guarantees in practical applications is a non-trivial challenge. Careful validation and testing would be required to assess the effectiveness and reliability of such machine learning-based approaches in real-world scenarios.

Coordination without global reference frames: In Chapter 7, a scenario with a global reference frame is considered for the sake of simplicity. This assumption is reasonable when using sensors such as GPS, ultra-wide band radio anchors, or visual fiducials in the environment. However, a more robust, cost-effective, and scalable solution is one that does not assume a global reference frame. To address this, our methods could be adapted along with other ideas from the literature to resolve the problem using local reference frames. Nonetheless, this adaptation process may not be trivial and requires

careful study and analysis.

Advanced formation and motion planning techniques: As discussed in Chapter 1, motion planning is a fundamental capability required in autonomous robots. However, in this work, we simplified the problem by assuming a fixed formation around a target of interest. To make our solutions more applicable in real-world scenarios, other aspects of motion planning need to be taken into account. For instance, incorporating affine transformations of a nominal formation based on the trajectory of leader agents could enhance the adaptability of the system. Moreover, collision avoidance between agents is crucial to ensure safe operation. More advanced motion planning and trajectory optimization techniques should be considered to avoid deadlocks and dead-end situations.

Real-world validation and platform integration: While this thesis has extensively evaluated the proposed algorithms through simulations and theoretical results, only few real-world experiments are provided such as the case of PLATE in Chapter 3. Hence, further validation on physical robotic platforms is essential to demonstrate their effectiveness and practicality. Integrating the perception latency-aware formation control and consensus algorithms into real robotic systems would provide valuable insights and validate their performance under real-world constraints.

Distributed optimization: In this thesis, EDC tools where used for information fusion and formation control applications. However, as discussed in [176, 177], dynamic average consensus algorithms can be used to enable distributed optimization with a wide range of applications such as distributed training of deep neural networks. In fact, we have already started exploring this idea in adjacent work of this thesis [107]. There, EDC tools are applied to a particular optimization problem of finding the smallest ellipsoid containing the intersection of a set of ellipsoids, with application to information fusion. Hence, this motivates to use EDC to broader classes of optimization programs in the future.

Conclusiones

En esta tesis, se han explorado diversas soluciones para la calendarización de latencia de percepción y estrategias de cooperación basadas en consenso para la fusión de información en sistemas multiagente. Estos enfoques son especialmente relevantes cuando múltiples robots intentan rastrear un objetivo de interés utilizando sus sensores a bordo, enfrentándose a restricciones de recursos como la latencia de percepción y las imperfecciones de la red.

Dada la complejidad del problema general, se adoptó una estrategia modular a lo largo de esta tesis. Al estudiar los problemas de calendarización de la latencia de percepción independientemente del consenso distribuido, se buscó obtener una comprensión más profunda de los desafíos específicos presentados por cada aspecto.

El Capítulo 2 exploró un problema de control que involucra mediciones en tiempo discreto con compensaciones entre latencia de percepción y precisión. El objetivo era desarrollar calendarizadores de latencia de percepción que optimicen el rendimiento del control y el uso de recursos. Sin embargo, se encontraron desafíos significativos relacionados con la estabilidad y la complejidad. El enfoque principal fue identificar candidatos a políticas de calendarización que preserven la estabilidad, lo que llevó a una conexión interesante entre este problema y la estabilidad de sistemas conmutativos. Aunque se avanzó en asegurar la estabilidad, se descubrió que encontrar el programa óptimo que preserve la estabilidad es un problema NP-difícil, lo que hace impráctico obtener soluciones óptimas. A pesar de esta limitación, nuestro marco demostró que la estabilidad se mantiene para cualquier programa, abriendo posibilidades para futuras investigaciones que exploren algoritmos subóptimos más eficientes. Aunque la complejidad del problema sigue siendo un desafío, las percepciones obtenidas en este capítulo sientan las bases para desarrollar algoritmos prácticos y efectivos de calendarización de latencia de percepción en sistemas de control.

El Capítulo 3, PLATE, se centra en el problema complementario de la estimación del estado de un sistema de entrada desconocida, como un objetivo de interés, utilizando sensores con compensaciones entre latencia de percepción y precisión. Al igual que en el Capítulo 2, este problema también enfrenta desafíos relacionados con la naturaleza combinatoria del espacio de soluciones. En respuesta, se propuso una solución aproximada eficiente para abordar el problema de estimación del estado bajo restricciones de latencia de percepción. El enfoque fue validado utilizando datos del mundo real en un contexto de omisión de cuadros, demostrando su aplicabilidad práctica y efectividad.

En el dominio de la cooperación multiagente, nos hemos centrado en técnicas de

consenso promedio dinámico, que forman los bloques fundamentales para la fusión de información en las partes posteriores de esta tesis. Nuestro enfoque ha presentado resultados en creciente complejidad, considerando diversas restricciones de red. El Capítulo 4 introdujo EDCHO en el contexto de una topología de red estática, mediciones sin ruido y comunicación en tiempo continuo. Esto sentó las bases para entender el consenso promedio en un entorno simplificado. Basándose en EDCHO, el Capítulo 5 presentó REDCHO, una extensión que acomoda cambios de conectividad, permitiendo la inclusión de nuevos agentes en la red de manera dinámica. Esta extensión nos permitió considerar ciertas clases de sistemas OMAS y demostró una mayor flexibilidad en el manejo de escenarios del mundo real. En el Capítulo 6, REDCHO se amplió para tener en cuenta el ruido en las mediciones y calcular promedios ponderados. Este paso hacia la incorporación de factores más realistas representa una contribución significativa, resultando en el primer diferenciador distribuido introducido en la literatura.

Los métodos presentados en los Capítulos 4, 5 y 6 ofrecen una notable salida de los protocolos de consenso lineal comúnmente encontrados en la literatura existente. La ventaja clave de estas técnicas radica en el uso de algoritmos de consenso no lineales, resultando en un consenso dinámico exacto sin ningún error en estado estacionario. En contraste, los enfoques lineales a menudo sufren de errores en estado estacionario limitados, incluso cuando operan sin ruido y en escenarios de comunicación en tiempo continuo. Las características de consenso dinámico preciso logradas por nuestros métodos son de particular importancia, especialmente al considerar la presencia de ruido, como se investigó en el Capítulo 6, superando a los protocolos lineales y de modo deslizante.

Nuestras herramientas para lograr un consenso dinámico exacto se basan en modos deslizantes de alto orden, que son fundamentales para habilitar la característica de exactitud. Sin embargo, es importante reconocer que estas técnicas no son inmunes a la conmutación, especialmente al lidiar con retrasos y actualizaciones en tiempo discreto. Estas limitaciones se cuantifican y analizan exhaustivamente en el Capítulo 8, demostrando que nuestros métodos aún pueden superar a las técnicas de vanguardia en una amplia gama de escenarios, incluso bajo condiciones tan desafiantes. Al evaluar rigurosamente el rendimiento de nuestros métodos en presencia de retrasos y actualizaciones en tiempo discreto, el Capítulo 8 arroja luz sobre la robustez y efectividad de nuestras técnicas.

Los Capítulos 7 y 8 abordaron los desafíos específicos de integrar la latencia de percepción y el consenso en sistemas multiagente. En el Capítulo 7, se exploró un escenario de control de formación donde la dinámica de los robots supera la velocidad de la etapa de percepción, haciendo que la generación de trayectorias de formación basada en mediciones de percepción no sea trivial. Para abordar este problema, se introdujo un estimador de salida suave, asegurando la estabilidad del controlador local para cada robot. La propiedad de suavidad del estimador también facilitó el uso de REDCHO como un bloque de construcción de fusión de información, ya que requiere entradas variables en el tiempo suficientemente suaves. Este enfoque modular nos permitió aprovechar las ventajas de la programación de latencia de percepción y las técnicas de consenso en un marco unificado.

Una suposición clave hecha en el Capítulo 7 fue que la comunicación entre agentes ocurre lo suficientemente rápido como para ser modelada en tiempo continuo. En escenarios del mundo real, los retrasos en la comunicación y la comunicación asincrónica son a menudo inevitables, requiriendo la consideración de estos factores para lograr una coordinación multiagente práctica y robusta. El Capítulo 8 abordó esta limitación manejando

directamente la comunicación asincrónica en tiempo discreto. Esta es la pieza final de la arquitectura presentada en la Figura 1.4 en el Capítulo 1, proporcionando un marco integral para sistemas multiagente cooperativos bajo restricciones de latencia de percepción y comunicación.

Como trabajo futuro, se sugiere explorar modelos de ruido, robot y objetivo más completos, considerando sistemas de estado espacial no lineales y modelos de ruido más allá de las distribuciones gaussianas. Otro aspecto es la programación imperfecta, reconociendo que en escenarios reales, los sistemas de cómputo pueden introducir imperfecciones y variaciones en la secuencia de muestreo. Además, se propone investigar topologías de red cambiantes y gráficos dirigidos, especialmente relevantes en redes de bajo recurso con cambios significativos en la conectividad. También se destacan las técnicas de optimización aproximada, donde el aprendizaje automático podría ofrecer enfoques más eficientes y menos conservadores, aunque validar su efectividad y fiabilidad en aplicaciones prácticas será un desafío. Otra área de interés es la coordinación sin marcos de referencia globales, adaptando los métodos actuales para trabajar con marcos de referencia locales. Además, se resalta la importancia de técnicas avanzadas de planificación de formación y movimiento, esenciales para la operación segura y eficaz de robots autónomos. Finalmente, se enfatiza la necesidad de validación en el mundo real e integración de plataformas para demostrar la efectividad de los algoritmos propuestos y, por último, se sugiere la aplicación de herramientas de consenso dinámico promedio (EDC) en optimización distribuida, lo que podría tener aplicaciones en áreas como el entrenamiento distribuido de redes neuronales profundas.

Appendix A

Sampled-data stochastic linear systems

In this section, we obtain expressions for $\hat{\mathbf{x}}[k|k-1] := \mathbb{E}\{\mathbf{x}[k]|\mathbf{z}[0],\ldots,\mathbf{z}[k-1]\}$, the distribution of $\mathbf{x}(t)$ as the closed-loop solution of (2.1) and an expression for (2.2).

Proposition A.1. *Consider Gaussian assumptions over the initial conditions of* (2.1)*, the disturbance* $\mathbf{w}(t)$ *as well as the measurement noise. Then, given measurements*

$$\mathbf{z}[0],\ldots,\mathbf{z}[k-1]$$

and a perception schedule p*, the value of*

$$\hat{\mathbf{x}}[k|k-1] := \mathbb{E}\left\{\mathbf{x}[k] \;\middle|\; \mathbf{z}[0],\ldots,\mathbf{z}[k-1]\right\}$$

is computed according to:

$$\begin{aligned}
\mathbf{H}[k] &= \mathbf{A}_{\mathsf{d}}(\Delta^{p_k})\hat{\mathbf{P}}[k]\mathbf{C}^\top\left(\mathbf{C}\hat{\mathbf{P}}[k]\mathbf{C}^\top + \mathbf{R}^{p_k}\right)^{-1}\\
\hat{\mathbf{x}}[k+1|k] &= \mathbf{A}_{\mathsf{d}}(\Delta^{p_k})\hat{\mathbf{x}}[k|k-1] + \mathbf{B}_{\mathsf{d}}(\Delta^{p_k})\mathbf{u}[k] + \mathbf{H}[k](\mathbf{z}[k] - \mathbf{C}\hat{\mathbf{x}}[k|k-1])\\
\hat{\mathbf{P}}[k+1] &= (\mathbf{A}_{\mathsf{d}}(\Delta^{p_k}) - \mathbf{H}[k]\mathbf{C})\hat{\mathbf{P}}[k]\mathbf{A}_{\mathsf{d}}(\Delta^{p_k})^\top + \mathbf{W}_{\mathsf{d}}(\Delta^{p_k})
\end{aligned}$$

with $\hat{\mathbf{x}}[0|-1] = \bar{\mathbf{x}}_0, \hat{\mathbf{P}}[0] = \mathbf{P}_0$*,*

$$\mathbf{A}_{\mathsf{d}}(\tau) := \exp(\mathbf{A}\tau), \mathbf{B}_{\mathsf{d}}(\tau) := \int_0^\tau \mathbf{A}_{\mathsf{d}}(s)\mathrm{d}s\mathbf{B}, \mathbf{W}_{\mathsf{d}}(\tau) := \int_0^\tau \mathbf{A}_{\mathsf{d}}(s)\mathbf{W}\mathbf{A}_{\mathsf{d}}(s)^\top \mathrm{d}s.$$

Moreover, $\mathsf{cov}\{\hat{\mathbf{x}}[k|k-1] - \mathbf{x}[k]\} = \hat{\mathbf{P}}[k]$*.*

Proof: First, note that the explicit solution to (2.1) in the interval $t - \tau_k \in [0, \Delta^{p_k})$ is given by

$$\mathbf{x}(t) = \mathbf{A}_{\mathsf{d}}(t-\tau_k)\mathbf{x}[k] + \mathbf{B}_{\mathsf{d}}(t-\tau_k)\mathbf{u}[k] + \mathbf{w}_d(t) \tag{A.1}$$

with $\mathbf{w}_d(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{W}_{\mathsf{d}}(t-\tau_k)$ [130, Section 4.5.2]. Note that evaluating (A.1) at $t = \tau_{k+1}$ leads to the discrete-time system which transitions from $\mathbf{x}[k]$ to $\mathbf{x}[k+1]$. Thus, the result follows by applying the estimator in [119, Theorem 4.1, Page 228]. □

Proposition A.2. *Let p a perception schedule and*

$$\mathbf{\Lambda}(t-\tau_k) := \mathbf{A}_\mathsf{d}(t-\tau_k) + \mathbf{B}_\mathsf{d}(t-\tau_k)\mathbf{L}^{p_k}, t-\tau_k \in [0, \Delta^{p_k})$$

The solution $\mathbf{x}(t)$ of the SDE in (2.1) under the controller $\mathbf{u}[k] = \mathbf{L}^{p_k}\hat{\mathbf{x}}[k|k-1]$ given samples $\{\mathbf{z}[0], \ldots, \mathbf{z}[k-1]\}$ is

$$\mathbf{x}(t) = \Lambda(t-\tau_k)\mathbf{x}[k] + \mathbf{w}_\Lambda(t) \tag{A.2}$$

for $t-\tau_k \in [0, \Delta^{p_k})$ and $\mathbf{w}_\Lambda(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{W}_\Lambda(t-\tau_k))$ with:

$$\mathbf{W}_\Lambda(t-\tau_k) = \mathbf{B}_\mathsf{d}(t-\tau_k)\mathbf{L}^{p_k}\hat{\mathbf{P}}[k]{\mathbf{L}^{p_k}}^\top\mathbf{B}_\mathsf{d}(t-\tau_k)^\top + \mathbf{W}_\mathsf{d}(t-\tau_k) \tag{A.3}$$

Moreover, $\mathbf{x}(t) \sim \mathcal{N}(\bar{\mathbf{x}}(t), \mathbf{P}(t))$ for $t-\tau_k \in [0, \Delta^{p_k})$ with

$$\begin{aligned} \bar{\mathbf{x}}(t) &= \mathbf{\Lambda}(t-t_k)\mathbf{x}[k], \quad \bar{\mathbf{x}}[0] = \bar{\mathbf{x}}_0 \\ \mathbf{P}(t) &= \mathbf{\Lambda}(t-\tau_k)\mathbf{P}[k]\mathbf{\Lambda}(t-\tau_k)^\top + \mathbf{W}_\Lambda(t-\tau_k) \end{aligned} \tag{A.4}$$

with $\mathbf{P}[0] = \mathbf{P}_0$. In addition, with $\alpha := \mathit{att}(p; [0, T_f])$, cost (2.2) is

$$\mathcal{J}(p) = \frac{\lambda_\mathbf{x}}{T_f}\int_0^{T_f} \bar{\mathbf{x}}(t)^\top\mathbf{Q}\bar{\mathbf{x}}(t) \; + \mathit{tr}(\mathbf{Q}\mathbf{P}(t))\mathrm{d}t + \lambda_\mathbf{x}\left(\bar{\mathbf{x}}(T_f)^\top\mathbf{Q}_f\bar{\mathbf{x}}(T_f) + \mathit{tr}(\mathbf{Q}_f\mathbf{P}(T_f))\right) + \frac{\lambda_r}{T_f}\sum_{k=0}^{\alpha-1} r^{p_k} \tag{A.5}$$

Proof: Proposition A.1 implies that

$$\hat{\mathbf{x}}[k|k-1] = \mathbf{x}[k] + \tilde{\mathbf{x}}[k]$$

where $\tilde{\mathbf{x}}[k] \sim \mathcal{N}(\mathbf{0}, \hat{\mathbf{P}}[k])$ given samples $\{\mathbf{z}[0], \ldots, \mathbf{z}[k-1]\}$. Hence, using

$$\mathbf{u}[k] = \mathbf{L}^{p_k}\mathbf{x}[k] + \boldsymbol{L}^{p_k}\tilde{\mathbf{x}}[k]$$

in (A.1) leads directly to (A.2) with

$$\mathbf{w}_\Lambda = \mathbf{B}_\mathsf{d}(t-\tau_k)\mathbf{L}^{p_k}\tilde{\mathbf{x}}[k] + \mathbf{w}_d(t).$$

Moreover, according to [130, Section 4.5.2] the process $\{\mathbf{w}_d[0], \ldots, \mathbf{w}_d[k], \mathbf{w}_d(t)\}$ is a white noise process and thus, its individual random variables are uncorrelated. Therefore, $\tilde{\mathbf{x}}[k]$ and $\mathbf{w}_d(t)$ are uncorrelated. Henceforth, computing the covariance of $\mathbf{w}_\Lambda(t)$ leads to (A.3). Furthermore, direct computation of the expectation and covariance over (A.2) leads to (A.4). Finally, (A.5) is computed as in [8, Theorem 1]. □

Appendix B

$C^\star$ sets and gauge functions

Definition B.1. *[115, Definition 1] A $C^\star$ set $\mathbb{S} \subset \mathbb{R}^n$ is a compact, star-convex set with the origin as a center i.e., for any $\mathbf{x} \in \mathbb{S}$ the whole line*

$$\{\mathbf{x}_\lambda \in \mathbb{R}^n : \mathbf{x}_\lambda = \lambda \mathbf{x},\ \lambda \in [0,1]\}$$

is contained in $\mathbb{S}$ and $0 \in \mathit{int}(\mathbb{S})$. Moreover, the gauge function of a $C^\star$ set $\mathbb{S}$ is defined as

$$\Psi(\mathbf{x};\mathbb{S}) := \inf\{\alpha \in \mathbb{R} : \alpha \geq 0,\ \mathbf{x} \in \alpha\mathbb{S}\}$$

This is, $\Psi(\mathbf{x};\mathbb{S})$ is the smallest scale α such that $\mathbf{x}$ is still contained in $\alpha\mathbb{S}$. In the following, we enumerate some properties regarding $C^\star$ sets and their gauge functions:

Lemma B.2. *Let $\mathbb{S}$ be a $C^\star$ set and $\Psi(\bullet;\mathbb{S}) : \mathbb{R}^n \to [0,+\infty)$ be its gauge function. Then, the following statements are true:*

1) Let $0 < \alpha < 1$ and $\mathbf{x} \in \mathbb{S}$, then $\alpha\mathbf{x} \in \mathbb{S}$.

2) Let $0 < \alpha < \beta$, then $\alpha\mathbb{S} \subset \beta\mathbb{S}$.

3) Let $\alpha := \Psi(\mathbf{x};\mathbb{S})$, then $\mathbf{x} \in \partial(\alpha\mathbb{S})$.

4) $\Psi(\bullet;\mathbb{S})$ is homogeneous of degree one, i.e. $\Psi(\beta\mathbf{x};\mathbb{S}) = \beta\Psi(\mathbf{x};\mathbb{S}),\ \forall\beta \geq 0$ and $\forall\mathbf{x} \in \mathbb{R}^n$.

5) Let $\mathbb{S}'$ be a $C^\star$ set with $\mathbb{S}' \subset \mathbb{S}$. Thus $\Psi(\mathbf{x};\mathbb{S}') > \Psi(\mathbf{x};\mathbb{S}), \forall\mathbf{x} \in \mathbb{R}^n$.

6) Let $\mathbb{S}_0 = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^T M_0 \mathbf{x} \leq 1\}$ for some positive definite matrix M_0. Then, $\Psi(\mathbf{x};\mathbb{S}_0) = \sqrt{\mathbf{x}^T M_0 \mathbf{x}}$ for any $\mathbf{x} \in \mathbb{R}^n$.

PROOF: These properties can be verified equivalently as the properties of gauge functions of standard convex sets [178, Page 28]. Item 1) follows from the definition of a $C^\star$ set, since the whole line $\mathbf{x}_\alpha = \alpha\mathbf{x}$ for $\alpha \in [0,1)$ and $\mathbf{x} \in \mathbb{S}$, is contained in $\mathbb{S}$. Item 2) follows from the fact that $\alpha/\beta < 1$ and thus, any element $\alpha/\beta\mathbf{x}$ with $\mathbf{x} \in \mathbb{S}$, is contained in $\mathbb{S}$ due to 1). Therefore, $\alpha/\beta\mathbb{S} \subset \mathbb{S}$, or equivalently $\alpha\mathbb{S} \subset \beta\mathbb{S}$. For 3) we proceed by contradiction.

Assume that $\mathbf{x} \in \mathsf{int}(\alpha\mathbb{S})$, then for exists $\varepsilon(\mathbf{x}) > 0$ such that the ball $\{\mathbf{x}_b : \|\mathbf{x}_b - \mathbf{x}\| \leq \varepsilon(\mathbf{x})\}$ is contained in $\mathsf{int}(\alpha\mathbb{S})$ [179, Definition 2.18-(e)]. Choose

$$\mathbf{x}_b^* = \left(1 + \frac{\varepsilon(\mathbf{x})}{\|\mathbf{x}\|}\right)\mathbf{x}$$

which satisfies

$$\|\mathbf{x}_b^* - \mathbf{x}\| = \varepsilon(\mathbf{x}),$$

and

$$\left(1 + \frac{\varepsilon(\mathbf{x})}{\|\mathbf{x}\|}\right)\mathbf{x} \in \mathsf{int}(\alpha\mathbb{S}) \subset \alpha\mathbb{S}$$

or equivalently

$$\mathbf{x} \in \alpha\left(1 + \frac{\varepsilon(\mathbf{x})}{\|\mathbf{x}\|}\right)^{-1}\mathbb{S}.$$

However

$$\alpha\left(1 + \frac{\varepsilon(\mathbf{x})}{\|\mathbf{x}\|}\right)^{-1} < \alpha$$

which is a contradiction for the definition of α, resulting in $\mathbf{x} \notin \mathsf{int}(\alpha\mathbb{S})$ or equivalently $\mathbf{x} \in \partial(\alpha\mathbb{S})$. Item 4) is enlisted in [115, Property 1] without proof. However, it can be verified by direct computation:

$$\begin{aligned}\Psi(\beta\mathbf{x};\mathbb{S}) &= \inf\{\alpha \in \mathbb{R} : \alpha \geq 0,\ \beta\mathbf{x} \in \alpha\mathbb{S}\} \\ &= \inf\{\alpha \in \mathbb{R} : \alpha = \beta\alpha',\ \alpha' \geq 0,\ \mathbf{x} \in \alpha'\mathbb{S}\} \\ &= \beta\inf\{\alpha' \in \mathbb{R} : \alpha' \geq 0,\ \mathbf{x} \in \alpha'\mathbb{S}\} = \beta\Psi(\mathbf{x};\mathbb{S})\end{aligned}$$

For item 5) let $\mathbf{y} = \mathbf{x}/\Psi(\mathbf{x};\mathbb{S})$. Thus,

$$\Psi(\mathbf{y};\mathbb{S}) = \Psi\left(\frac{\mathbf{x}}{\Psi(\mathbf{x};\mathbb{S})};\mathbb{S}\right) = \frac{\Psi(\mathbf{x};\mathbb{S})}{\Psi(\mathbf{x};\mathbb{S})} = 1$$

by item 4). Hence, $\mathbf{y} \in \partial\mathbb{S}$ by item 3). However, since $\mathbb{S}' \subset \mathbb{S}$, therefore $\mathbf{y} \notin \mathbb{S}'$. Hence,

$$\Psi(\mathbf{y};\mathbb{S}') = \Psi\left(\frac{\mathbf{x}}{\Psi(\mathbf{x};\mathbb{S})};\mathbb{S}'\right) > 1$$

or equivalently $\Psi(\mathbf{x};\mathbb{S}') > \Psi(\mathbf{x};\mathbb{S})$ by item 3).

Finally, for 6), note that $\alpha\mathbb{S}_0 = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^T M_0 \mathbf{x} \leq \alpha^2\}$. Thus,

$$\begin{aligned}&\Psi(\mathbf{x};\mathbb{S}_0) = \inf\{\alpha \in \mathbb{R} : \alpha \geq 0,\ \mathbf{x} \in \alpha\mathbb{S}_0\} \\ &= \inf\{\alpha \in \mathbb{R} : \alpha \geq 0,\ \mathbf{x}^T M_0 \mathbf{x} \leq \alpha^2\} = \inf\left[\sqrt{\mathbf{x}^T M_0 \mathbf{x}}, \infty\right)\end{aligned}$$

□

Appendix C

Auxiliary results in vector and matrix analysis

Proposition C.1. *Let* $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_n]^\top \in \mathbb{R}^n$ *and define*

$$\|\boldsymbol{\lambda}\|_p = \left(\sum_{i=1}^{n} |\lambda_i|^p \right)^{1/p}.$$

Then, with $0 < r < s$*, the following inequalities are satisfied:*

a) [180, Theorem 16, Page 26] $\|\boldsymbol{\lambda}\|_r \leq n^{\frac{1}{r}-\frac{1}{s}} \|\boldsymbol{\lambda}\|_s$.

b) [180, Theorem 19, Page 28] $\|\boldsymbol{\lambda}\|_s \leq \|\boldsymbol{\lambda}\|_r$.

Lemma C.2. *Let* $\mathbf{M}_1, \mathbf{M}_2 \in \mathbb{R}^{n \times n}$ *be positive definite matrices such that* $\mathbf{M}_1 \preceq \mathbf{M}_2$*. Then, the following inequalities are satisfied:*

a) $\mathsf{tr}(\mathbf{M}_1) \leq \mathsf{tr}(\mathbf{M}_2)$.

b) $\|\mathbf{M}_1\|_F \leq \sqrt{n}\|\mathbf{M}_2\|_F$

Proof: First, note that $\mathbf{M}_2 - \mathbf{M}_1$ is positive semi-definite. Then, use the spectral Theorem [123, Theorem 2.5.6] to conclude that the eigenvalues $\{\lambda_i\}_{i=1}^n$ of $\mathbf{M}_2 - \mathbf{M}_1$ are all different non-negative real numbers. Thus,

$$\mathsf{tr}(\mathbf{M}_2) - \mathsf{tr}(\mathbf{M}_1) = \mathsf{tr}(\mathbf{M}_2 - \mathbf{M}_1) = \sum_{i=1}^{n} \lambda_i \geq 0,$$

where the relation between the trace and the sum of the eigenvalues was used in the last step [123, Page 50] completing the proof for a). Similarly, $\mathbf{M}_1, \mathbf{M}_2$ have non-negative eigenvalues too. Let $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2 \in \mathbb{R}^n$ be vectors containing the eigenvalues of $\mathbf{M}_1, \mathbf{M}_2$ respectively. Thus, item a) reads $\|\boldsymbol{\lambda}_1\|_1 \leq \|\boldsymbol{\lambda}_2\|_1$ using the notation of Proposition C.1 in C. Proposition C.1-a) implies $\|\boldsymbol{\lambda}_2\|_1 \leq \sqrt{n}\|\boldsymbol{\lambda}_2\|_2$. Proposition C.1-b) implies $\|\boldsymbol{\lambda}_1\|_2 \leq \|\boldsymbol{\lambda}_1\|_1$. Thus, $\|\boldsymbol{\lambda}_1\|_2 \leq \sqrt{n}\|\boldsymbol{\lambda}_2\|_2$. Now, recall that

$$\|\boldsymbol{\lambda}_1\|_2 = \sqrt{\sum_{i=1}^{n} \lambda_i(\mathbf{M}_1)^2} \equiv \|\mathbf{M}_1\|_F$$

where $\lambda_i(\mathbf{M}_1)$ are the eigenvalues of $\mathbf{M}_1$ and similarly for $\mathbf{M}_2$ [123, Page 342]. Thus, item b) follows. □

Lemma C.3. *Let $\mathbf{M}_1, \mathbf{M}_2 \in \mathbb{R}^{n\times n}$ be symmetric matrices satisfying $\mathbf{M}_1 \preceq \mathbf{M}_2$. Then, $(\mathbf{M}_1 \otimes \mathbf{M}_1) \preceq (\mathbf{M}_2 \otimes \mathbf{M}_2)$.*

PROOF: First, recall from [123, Theorem 7.7.3-(a)] that $\mathbf{M}_1 \preceq \mathbf{M}_2$ if and only if

$$\rho(\mathbf{M}_2^{-1}\mathbf{M}_1) \leq 1$$

where $\rho(\bullet)$ denotes the spectral radius [123, Definition 1.2.9]. Then,

$$1 \geq \rho(\mathbf{M}_2^{-1}\mathbf{M}_1)^2 = \rho((\mathbf{M}_2^{-1}\mathbf{M}_1) \otimes (\mathbf{M}_2^{-1}\mathbf{M}_1)) = \rho((\mathbf{M}_2 \otimes \mathbf{M}_2)^{-1}(\mathbf{M}_1 \otimes \mathbf{M}_1)),$$

which implies $(\mathbf{M}_1 \otimes \mathbf{M}_1) \preceq (\mathbf{M}_2 \otimes \mathbf{M}_2)$ using [123, Theorem 7.7.3-(a)]. □

Corollary C.4. *Let $\mathbf{x} \in \mathbb{R}^n$ and $0 < \alpha < 1$. Then,*

1. $\| \lceil \mathbf{x} \rfloor^{\alpha} \| \leq n^{\frac{1-\alpha}{2}} \|\mathbf{x}\|^{\alpha}$
2. $\mathbf{x}^{\top} \lceil \mathbf{x} \rfloor^{\alpha} \geq \|\mathbf{x}\|^{\alpha+1}$

PROOF: For the first item, note that

$$\| \lceil \mathbf{x} \rfloor^{\alpha} \|^{\frac{1}{\alpha}} = \left(\sum_{i=1}^{n} |x_i|^{2\alpha} \right)^{\frac{1}{2\alpha}} = \|\mathbf{x}\|_{2\alpha}.$$

Moreover, Proposition C.1-(a) with $2\alpha < 2$ leads to

$$\|\mathbf{x}\|_{2\alpha} \leq n^{\frac{1}{2\alpha}-\frac{1}{2}} \|\mathbf{x}\|_2 = n^{\frac{1-\alpha}{2\alpha}} \|\mathbf{x}\|.$$

For the second item note that

$$\mathbf{x}^{\top} \lceil \mathbf{x} \rfloor^{\alpha} = \sum_{i=1}^{n} x_i \lceil x_i \rfloor^{\alpha} = \sum_{i=1}^{n} |x_i|^{\alpha+1} = \left(\|\mathbf{x}\|_{\alpha+1} \right)^{\alpha+1}.$$

Moreover, Proposition C.1-(b) with $\alpha + 1 < 2$ leads to $\|\mathbf{x}\|_{\alpha+1} \geq \|\mathbf{x}\|_2$. Hence,

$$\mathbf{x}^{\top} \lceil \mathbf{x} \rfloor^{\alpha} = \left(\|\mathbf{x}\|_{\alpha+1} \right)^{\alpha+1} \geq \|\mathbf{x}\|^{\alpha+1}.$$

□

Proposition C.5. *Let $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{A} \in \mathbb{R}^{m\times n}$. Then,*

$$s_{\min}(\mathbf{A})\|\mathbf{x}\| \leq \|\mathbf{A}\mathbf{x}\| \leq s_{\max}(\mathbf{A})\|\mathbf{x}\|.$$

PROOF: This proposition is a direct consequence of the Rayleigh inequality [123, Theorem 4.2.2] and the definition of the singular values of $\mathbf{A}$ in [123, Page 151]. □

Appendix D

Auxiliary results in algebraic graph theory

An undirected graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ consists of a node set $\mathcal{I}$ of n nodes and an edge set $\mathcal{E}$ of ℓ edges [158, Page 1]. An edge from node i to node j is denoted as (i, j), which means that node i can communicate to node j in a bidirectional way. $\mathcal{G}$ is said to be connected if there is a path between any two nodes. A subgraph is a cycle if every node in it has exactly two neighbors. $\mathcal{G}$ is said to be a tree, if it is connected and it has no cycles. A spanning tree is a subgraph of $\mathcal{G}$ if it contains all its nodes and is a tree. If $\mathcal{G}$ is connected, there is always a spanning tree [158, Page 4]. Moreover, a tree has exactly $\ell = \mathsf{N} - 1$ edges [158, Page 53]. Furthermore, we define the union of two undirected graphs $\mathcal{G}_A = (\mathcal{I}_A, \mathcal{E}_A)$ and $\mathcal{G}_B = (\mathcal{I}_B, \mathcal{E}_B)$ as $\mathcal{G}_{AB} = (\mathcal{I}_A \cup \mathcal{I}_B, \mathcal{E}_A \cup \mathcal{E}_B)$.

Definition D.1 (Matrices of interest for $\mathcal{G}$)**.** *The following matrices are defined for $\mathcal{G}$:*

1. *[158, Page 163] The adjacency matrix $\mathbf{A} \in \mathbb{R}^{\mathsf{N}\times\mathsf{N}}$ of an undirected graph is defined by its components $[\mathbf{A}]_{ij}$ which comply $a_{ij} = [\mathbf{A}]_{ij} := 1$ if $(i, j) \in \mathcal{E}$ and $a_{ij} := 0$ otherwise.*
2. *[158, Page 167] An incidence matrix $\mathbf{D} \in \mathbb{R}^{\mathsf{N}\times\ell}$ for $\mathcal{G}$ has a column per edge, where all elements of the column corresponding to edge (i, j) are 0 except for the i-th element which is 1 and the j-th which is -1.*
3. *[158, Page 279] The Laplacian matrix of $\mathcal{G}$ is defined as $\mathbf{Q} := \mathbf{D}\mathbf{D}^\top \in \mathbb{R}^{\mathsf{N}\times\mathsf{N}}$.*
4. *[158, Page 305] For any connected graph $\mathcal{G}$, the algebraic connectivity $c(\mathcal{G}) > 0$ is defined as the second smallest eigenvalue of $\mathbf{Q}$.*

Proposition D.2 (Some algebraic properties of $\mathcal{G}$)**.** *Let $\mathcal{G}$ be a connected undirected graph, $x \in \mathbb{R}^{\mathsf{N}}$ be any vector orthogonal to $\mathbb{1}$ and M be the dimension of the null space of $\mathbf{D}$ (flow space). Then,*

1. *[158, Lemma 13.1.1] rank$(\mathbf{Q}) = \mathsf{N} - 1$.*
2. *[158, Corollary 13.4.2] $\mathbf{x}^\top\mathbf{Q}\mathbf{x} \geq c(\mathcal{G})\mathbf{x}^\top\mathbf{x}$.*
3. *[158, Page 280] $\mathbf{D}^\top\mathbb{1} = 0$.*

4. [158, Theorem 14.2.1] $M = \ell - n + 1$.

Proposition D.3. *Let* $\mathcal{G}_{AB} = (\mathcal{I}, \mathcal{E})$ *be an undirected connected graph with flow space of dimension* $M > 0$*. Then, there exists undirected connected graphs* $\mathcal{G}_A, \mathcal{G}_B$ *over the same nodes* $\mathcal{I}$*, with flow space of dimension* $M - 1$ *whose union is* $\mathcal{G}_{AB}$*.*

PROOF: Choose any spanning tree $\mathcal{G}_{AB}^{\mathsf{tree}}$ of $\mathcal{G}_{AB}$ with $\ell_{\mathsf{tree}} = \mathsf{N} - 1$ edges. Then, from Proposition D.2-(4), $\ell - \ell_{\mathsf{tree}} = M > 0$. Consider first $M = 1$. Hence, there is exactly one edge which isn't part of the spanning tree. Moreover, denote this edge as $e = (i, j)$ where $i, j \in \mathcal{I}$. Then, since e is not in the spanning tree, it is in a cycle and i, j have at least two neighbors each. Therefore, there are at least two ways to reach i and j other nodes. Consequently there are at least two different spanning trees $\mathcal{G}_A, \mathcal{G}_B$ (with flow space of dimension $M - 1 = 0$) which contain e. Now, for $M \geq 2$, there exists at least two edges e, e' which are not in the spanning tree. Let $\mathcal{G}_A$ and $\mathcal{G}_B$ be $\mathcal{G}_{AB}$ without e and e' respectively. These graphs are connected over the same node set $\mathcal{I}$ since they contain the same spanning tree of $\mathcal{G}_{AB}$. Moreover, they have flow space of dimension $M - 1$ by Proposition D.2-(4) since they have one edge less than $\mathcal{G}_{AB}$. □

Proposition D.4. *Let* $\mathcal{G}$ *be connected. Then any* $\mathbf{x} \in \mathbb{R}^{\mathsf{N}}$ *can be written as* $\mathbf{x} = \alpha \mathbb{1} + \mathbf{D}\tilde{\mathbf{x}}$ *with* $\alpha \in \mathbb{R}$ *and* $\tilde{\mathbf{x}} \in \mathbb{R}^{\ell}$*.*

PROOF: Let $\lambda_1, \ldots, \lambda_n$ and $\mathbf{v}_1, \ldots, \mathbf{v}_2$ be the eigenvalues and eigenvectors of $\mathbf{Q}$ respectively with $\|\mathbf{v}_i\| = 1, i = 1, \ldots, n$. First, from Propositions D.2-(1) and D.2-(3) we know that $\mathbf{Q}$ has $\lambda_1 = 0$ and that $(1/\sqrt{\mathsf{N}})\mathbb{1}$ is its only eigenvector. Hence, $\{(1/\sqrt{\mathsf{N}})\mathbb{1}, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ is an ortho-normal basis of $\mathbb{R}^{\mathsf{N}}$ by the spectral Theorem [123, Theorem 2.5.6]. Consequently, any vector $\mathbf{x} \in \mathbb{R}^{\mathsf{N}}$ can be decomposed as a vector in the image of $\mathbf{Q}$ and a component parallel to $\mathbb{1}$, equivalently $\mathbf{x} = \alpha\mathbb{1} + \mathbf{Q}\mathbf{y}$ for $\alpha \in \mathbb{R}$ and $\mathbf{y} = \mathbb{R}^{\mathsf{N}}$. Additionally, let $\tilde{\mathbf{x}} = \mathbf{D}^{\top}\mathbf{y}$ obtaining $\mathbf{x} = \alpha\mathbb{1} + \mathbf{D}\mathbf{D}^{\top}\mathbf{y} = \alpha\mathbb{1} + \mathbf{D}\tilde{\mathbf{x}}$. □

Appendix E

Exact differentiation

In this section we provide some results that were used in [96] to show the stability of the Levant's arbitrary order exact differentiator. In particular, we are interested in the properties of the recursive system

$$\begin{aligned}
\dot{\sigma}_1(t) &= \sigma_2(t) - \lambda_1 \lceil \sigma_1(t) + \theta(t) \rfloor^{\frac{m-1}{m}} \\
\dot{\sigma}_\mu(t) &= \sigma_{\mu+1}(t) - \lambda_\mu \lceil \sigma_\mu(t) - \dot{\sigma}_{\mu-1}(t) \rfloor^{\frac{m-\mu}{m-(\mu-1)}} \\
&\quad \text{for } 1 < \mu < m \\
\dot{\sigma}_m(t) &\in -\lambda_m \lceil \sigma_m(t) - \dot{\sigma}_{m-1}(t) \rfloor^0 + [-L, L]
\end{aligned} \tag{E.1}$$

with $\sigma_\mu \in \mathbb{R}, 1 \leq \mu \leq m$, and the measurable map $\theta : \mathbb{R}_+ \to [-\bar{\theta}, \bar{\theta}]$ with $\bar{\theta} > 0$. Two important results regarding the contraction property of (E.1) are given.

Proposition E.1 (Arbitrary boundedness of (E.1))**.** *[96, Lemma 7] Let $\theta : \mathbb{R}_+ \to [-\bar{\theta}, \bar{\theta}]$ satisfy the condition that $\int_{t_0}^{t_0+\delta} |\theta(\tau)| \mathrm{d}\tau < K$ for some $K > 0$. Then, for any $0 < \Omega_\mu < \Omega'_\mu, 0 \leq \mu \leq m$ there exists $\delta > 0$ (sufficiently small) such that any trajectory of* (E.1) *satisfying $|\sigma_\mu(t_0)| \leq \Omega_\mu$ will satisfy $|\sigma_\mu(t)| \leq \Omega'_\mu, \forall t \in [t_0, t_0 + \delta]$.*

Proposition E.2 (Contraction property of (E.1))**.** *[96, Lemma 8] For any $0 < \omega_\mu < \Omega_\mu, 1 \leq \mu \leq m$ there exists $\Omega_\mu < \Psi_\mu$, $T > 0$, some gains $\lambda_1, \ldots, \lambda_m > 0$ (sufficiently big) and $\bar{\theta} > 0$ (sufficiently small) such that any trajectory of* (E.1) *satisfying $|\sigma_\mu(t_0)| \leq \Omega_\mu$ will satisfy $|\sigma_\mu(t)| \leq \Psi_\mu, \forall t \in [t_0, t_0 + T]$ and $|\sigma_\mu(t)| \leq \omega_\mu, \forall t \in [T, +\infty)$.*

Appendix F

Homogeneous differential inclusions

In this section, we consider dynamical systems characterized by set-valued maps $\mathbf{f} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ instead of typical vector fields. Moreover, we assume some regularity conditions on such maps called the basic conditions. We say that a set valued map $\mathbf{f} : \mathcal{S} \rightrightarrows \mathbb{R}^n$ satisfies the basic conditions if for all $\mathbf{x} \in \mathcal{S}$ the set $\mathbf{f}(\mathbf{x})$ is non-empty, bounded, closed, convex and the map $\mathbf{f}$ is upper semi-continuous in $\mathbf{x}$ [181, Chapter 2.7]. The Filippov regularization for vector fields [155], commonly used to study discontinuous dynamical systems, satisfy the basic conditions by construction. In the following, let $\mathbf{\Delta}_\mathbf{r}(\lambda) = \text{diag}([\lambda^{r_1}, \ldots, \lambda^{r_n}])$ where $\mathbf{r} = [r_1, \ldots, r_n]$ are called the weights and $\lambda > 0$. For any $\mathbf{x} \in \mathbb{R}^n$, the vector $\mathbf{\Delta}_\mathbf{r}(\lambda)\mathbf{x} = [\lambda^{r_1} x_1, \ldots, \lambda^{r_n} x_n]^T$ is called its standard dilation (weighted by $\mathbf{r}$). The following are some definitions and results of interest regarding the so called $\mathbf{r}$-homogeneity with respect to the standard dilation.

Definition F.1 (Homogeneous scalar functions)**.** *[159, Definition 4.7] A scalar function $V : \mathbb{R}^n \to \mathbb{R}$ is said to be $\mathbf{r}$-homogeneous of degree d if $V(\mathbf{\Delta}_\mathbf{r}(\lambda)\mathbf{x}) = \lambda^d V(\mathbf{x})$ for any $\mathbf{x} \in \mathbb{R}^n$.*

Definition F.2 (Homogeneous set-valued fields)**.** *[159, Definition 4.20] A set-valued vector field $\mathbf{f} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is said to be $\mathbf{r}$-homogeneous of degree d if $\mathbf{f}(\mathbf{\Delta}_\mathbf{r}(\lambda)\mathbf{x}) = \lambda^d \mathbf{\Delta}_\mathbf{r}(\lambda)\mathbf{f}(\mathbf{x})$ for any $\mathbf{x} \in \mathbb{R}^n$.*

Proposition F.3. *[163, Lemma 4.2] Let $V_1, V_2 : \mathbb{R}^n \to \mathbb{R}$ be continuous functions, $\mathbf{r}$-homogeneous of degrees $d_1 > 0, d_2 > 0$ (respectively). Moreover, let $V_1(\mathbf{x})$ to be positive definite. Then, $\forall \mathbf{x} \in \mathbb{R}^n$,*

$$\beta_m V_1(\mathbf{x})^{d_2/d_1} \leq V_2(\mathbf{x}) \leq \beta_M V_1(\mathbf{x})^{d_2/d_1}$$

with $\beta_m = \inf\{V_2(\mathbf{x}), \forall x : V_1(\mathbf{x}) = 1\}$ and $\beta_M = \sup\{V_2(\mathbf{x}), \forall x : V_1(\mathbf{x}) = 1\}$.

Proposition F.4. *[163, Section 5] Let $V : \mathbb{R}^n \to \mathbb{R}$ and $\mathbf{f} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ be a scalar field and set valued vector field, $\mathbf{r}$-homogeneous of degrees l and m respectively. Then, $L_\mathbf{f} V(\mathbf{x})$ is $\mathbf{r}$-homogeneous of degree $l + m$.*

Proposition F.5. *[159, Theorem 4.24] Let $\mathbf{f} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ be $\mathbf{r}$-homogeneous of degree m satisfying the standard assumptions. Moreover, assume that the differential inclusion*

$\dot{x} \in \mathbf{f}(\mathbf{x})$ is strongly, globally asymptotically stable. Then, for any $k > \max(-m, 0)$, there exists $V, W : \mathbb{R}^n \to \mathbb{R}$ continuously differentiable in all $\mathbb{R}^n$ and $\mathbb{R}^n \setminus \{0\}$ respectively. Moreover, V is positive definite and $\mathbf{r}$-homogeneous of degree k and W is strictly positive outside the origin and $\mathbf{r}$-homogeneous of degree $k + m$. Finally, $\dot{V} \leq -W(\mathbf{x}), \forall \mathbf{x} \neq 0$.

Bibliography

[1] R. Aldana-Lopez, R. Aragues, and C. Sagues, "EDCHO: High order exact dynamic consensus," *Automatica*, vol. 131, p. 109750, 2021.

[2] R. Aldana-Lopez, R. Aragues, and C. Sagues, "REDCHO: Robust exact dynamic consensus of high order," *Automatica*, vol. 141, p. 110320, 2022.

[3] R. Aldana-Lopez, R. Aragues, and C. Sagues, "Latency vs precision: Stability preserving perception scheduling," *Automatica*, vol. 155, p. 111123, 2023.

[4] R. Aldana-Lopez, R. Aragues, and C. Sagues, "Perception-latency aware distributed target tracking," *Information Fusion*, p. 101857, 2023.

[5] R. Aldana-Lopez, R. Aragues, and C. Sagues, "PLATE: A perception-latency aware estimator," *ISA Transactions*, vol. 142, pp. 716–730, 2023.

[6] R. Aldana-Lopez, R. Aragues, and C. Sagues, "Distributed differentiation with noisy measurements for exact dynamic consensus," *IFAC-PapersOnLine*, 2023. 22th IFAC World Congress.

[7] R. Aldana-Lopez, R. Aragues, and C. Sagues, "EDC: Exact dynamic consensus," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 2921–2926, 2020. 21th IFAC World Congress.

[8] R. Aldana-Lopez, R. Aragues, and C. Sagues, "Attention vs. precision: latency scheduling for uncertainty resilient control systems," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 5697–5702, 2020.

[9] R. Aldana-Lopez, R. Aragues, and C. Sagues, "Quasi-exact dynamic consensus under asynchronous communication and symmetric delays," *Submitted to IEEE Transactions on Automatic Control*.

[10] I. Perez-Salesa, R. Aldana-Lopez, and C. Sagues, "Precise dynamic consensus under event-triggered communication," *Machines*, vol. 11, no. 2, 2023.

[11] D. Gomez-Gutierrez, R. Aldana-Lopez, R. Seeber, M. T. Angulo, and L. Fridman, "An arbitrary-order exact differentiator with predefined convergence time bound for signals with exponential growth bound," *Automatica*, vol. 153, p. 110995, 2023.

[12] R. Aldana-Lopez, R. Seeber, D. Gomez-Gutierrez, M. T. Angulo, and M. Defoort, "A redesign methodology generating predefined-time differentiators with bounded time-varying gains," *International Journal of Robust and Nonlinear Control*, pp. 1–16, 2022.

[13] R. Aldana-Lopez, D. Gomez-Gutierrez, R. Aragues, and C. Sagues, "Dynamic consensus with prescribed convergence time for multileader formation tracking," *IEEE Control Systems Letters*, vol. 6, pp. 3014–3019, 2022.

[14] A. Ramirez-Perez, R. Aldana-Lopez, O. Longoria-Gandara, J. Valencia-Velasco, L. Pizano-Escalante, and R. Parra-Michel, "Modular arithmetic cpm for sdr platforms," *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1–1, 2022.

[15] H. Haimovich, R. Seeber, R. Aldana-Lopez, and D. Gomez-Gutierrez, "Differentiator for noisy sampled signals with best worst-case accuracy," *IEEE Control Systems Letters*, pp. 1–1, 2021.

[16] R. Aldana-Lopez, D. Gomez-Gutierrez, M. A. Trujillo, M. Navarro-Gutierrez, J. Ruiz-Leon, and H. M. Becerra, "A predefined-time first-order exact differentiator based on time-varying gains," *International Journal of Robust and Nonlinear Control*, vol. 31, pp. 5510–5522, 2021.

[17] R. Aldana-Lopez, D. Gomez-Gutierrez, E. Jimenez-Rodriguez, J. D. Sanchez-Torres, and M. Defoort, "Generating new classes of fixed-time stable systems with predefined upper bound for the settling time," *International Journal of Control*, vol. 0, no. 0, pp. 1–13, 2021.

[18] L. Campos-Macias, R. Aldana-Lopez, R. de la Guardia, J. I. Parra-Vilchis, and D. Gomez-Gutierrez, "Autonomous navigation of mavs in unknown cluttered environments," *J. Field Rob.*, vol. 38, no. 2, pp. 307–326, 2021.

[19] M. Trujillo, R. Aldana-Lopez, D. Gomez-Gutierrez, M. Defoort, J. Ruiz-Leon, and H. M. Becerra, "Autonomous and non-autonomous fixed-time leader-follower consensus for second-order multi-agent systems," *Nonlinear Dyn.*, vol. 102, pp. 1–18, 12 2020.

[20] R. Aldana-Lopez, D. Gomez-Gutierrez, E. Jimenez-Rodriguez, J. Sanchez-Torres, and A. Loukianov, "On predefined-time consensus protocols for dynamic networks," *Journal of the Franklin Institute*, vol. 357, no. 16, pp. 11880–11899, 2020. Finite-Time Stability Analysis and Synthesis of Complex Dynamic Systems.

[21] J. D. Sanchez-Torres, A. J. Munoz-Vazquez, M. Defoort, R. Aldana-Lopez, and D. Gomez-Gutierrez, "Predefined-time integral sliding mode control of second-order systems," *International Journal of Systems Science*, vol. 51, no. 16, pp. 3425–3435, 2020.

[22] J. Valencia-Velasco, O. Longoria-Gandara, R. Aldana-Lopez, and L. Pizano-Escalante, "Low-complexity maximum-likelihood detector for IoT BLE devices," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4737–4745, 2020.

[23] E. Jimenez-Rodriguez, R. Aldana-Lopez, J. D. Sanchez-Torres, D. Gomez-Gutierrez, and A. G. Loukianov, "Consistent discretization of a class of predefined-time stable systems," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 628–633, 2020. 21st IFAC World Congress.

[24] I. Perez-Salesa, R. Aldana-Lopez, and C. Sagues, "Event-based visual tracking in dynamic environments," in *ROBOT2022: Fifth Iberian Robotics Conference* (D. Tardioli, V. Matellan, G. Heredia, M. F. Silva, and L. Marques, eds.), (Cham), pp. 175–186, Springer International Publishing, 2023.

[25] I. Perez-Salesa, R. Aldana-Lopez, and C. Sagues, "Event-triggered consensus for continuous-time distributed estimation," *IFAC-PapersOnLine*, 2023. 22th IFAC World Congress.

[26] I. Perez-Salesa, R. Aldana-Lopez, and C. Sagues, "Remote estimation with bounded uncertainty under dynamic event-triggered communication," *Submitted to IEEE Transactions on Systems, Man, and Cybernetics*.

[27] R. Aldana-Lopez, M. Aranda, R. Aragues, and C. Sagues, "Robust affine formation tracking," *Submitted to IEEE Transactions on Automatic Control*.

[28] P. Chen, Y. Dang, R. Liang, W. Zhu, and X. He, "Real-time object tracking on a drone with multi-inertial sensing data," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 131–139, 2018.

[29] A. Khan, B. Rinner, and A. Cavallaro, "Cooperative robots to observe moving targets: Review," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 187–198, 2018.

[30] Y. Lan, Z. Lin, M. Cao, and G. Yan, "A distributed reconfigurable control law for escorting and patrolling missions using teams of unicycles," in *49th IEEE Conference on Decision and Control (CDC)*, pp. 5456–5461, 2010.

[31] F. Castanedo, J. Garcia, M. A. Patricio, and J. M. Molina, "Data fusion to improve trajectory tracking in a cooperative surveillance multi-agent architecture," *Information Fusion*, vol. 11, no. 3, pp. 243–255, 2010. Agent-Based Information Fusion.

[32] L.-E. Caraballo, A. Montes-Romero, J.-M. Diaz-Banez, J. Capitan, A. Torres-Gonzalez, and A. Ollero, "Autonomous planning for multiple aerial cinematographers," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1509–1515, 2020.

[33] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Auton. Robots*, vol. 30, no. 1, pp. 73–86, 2011.

[34] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*. The MIT Press, 2nd ed., 2011.

[35] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE Robotics Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[36] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[37] R. Mur-Artal and J. D. Tardos, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[38] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, pp. 13–es, 2006.

[39] B. Bescos, C. Campos, J. D. Tardos, and J. Neira, "Dynaslam ii: Tightly-coupled multi-object tracking and slam," *IEEE Rob. Autom. Lett.*, vol. 6, no. 3, pp. 5191–5198, 2021.

[40] F. Madrigal and J.-B. Hayet, "Motion priors based on goals hierarchies in pedestrian tracking applications," *Machine Vision and Applications*, vol. 28, pp. 341–359, 2017.

[41] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 37, pp. 311–324, 2007.

[42] S. R. E. Datondji, Y. Dupuis, P. Subirats, and P. Vasseur, "A survey of vision-based traffic monitoring of road intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, pp. 2681–2698, 2016.

[43] K. Hashimoto and D. V. Dimarogonas, "Resource-aware networked control systems under temporal logic specifications," *Discrete Event Dynamic Systems*, vol. 29, no. 4, pp. 473–499, 2019.

[44] J. Araújo, "Design, implementation and validation of resource-aware and resilient wireless networked control systems," 2014.

[45] D. Falanga, S. Kim, and D. Scaramuzza, "How fast is too fast? the role of perception latency in high-speed sense and avoid," *IEEE Rob. Autom. Lett.*, vol. 4, no. 2, pp. 1884–1891, 2019.

[46] Y. V. Pant, H. Abbas, K. Mohta, R. A. Quaye, T. X. Nghiem, J. Devietti, and R. Mangharam, "Anytime computation and control for autonomous systems," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 2, pp. 768–779, 2021.

[47] H. M. Strasdat, J. M. M. Montiel, and A. J. Davison, "Visual slam: Why filter?," *Image Vis. Comput.*, vol. 30, pp. 65–77, 2012.

[48] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems," *IEEE Trans. Rob.*, vol. 33, no. 2, pp. 249–265, 2017.

[49] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *PROC CVPR IEEE*, pp. 779–788, 2016.

[50] S. Beery, G. Wu, V. Rathod, R. Votel, and J. Huang, "Context r-cnn: Long term temporal context for per-camera object detection," *PROC CVPR IEEE*, pp. 13072–13082, 2020.

[51] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[52] J. Amirian, B. Zhang, F. V. Castro, J. J. Baldelomar, J.-B. Hayet, and J. Pettre, "Opentraj: Assessing prediction complexity in human trajectories datasets," *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pp. 379–387, 2020.

[53] L. Wang, L. Zhang, and Z. Yi, "Trajectory predictor by using recurrent neural networks in visual tracking," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3172–3183, 2017.

[54] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," *PROC CVPR IEEE*, pp. 3296–3297, 2017.

[55] F. Fraundorfer, L. Heng, D. Honegger, G. Lee, L. Meier, P. Tanskanen, and M. Pollefeys, "Vision-based autonomous mapping and exploration using a quadrotor mav," in *IEEE/RSJ IROS*, pp. 4557–4564, 2012.

[56] H. Oleynikova, Z. Taylor, R. Siegwart, and J. Nieto, "Safe local exploration for replanning in cluttered unknown environments for micro-aerial vehicles," *IEEE Rob. Autom. Lett.*, 2018.

[57] H. Hu, D. Dey, M. Hebert, and J. Bagnell, "Learning anytime predictions in neural networks via adaptive loss balancing," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 3812–3821, 2019.

[58] S. Yao, Y. Hao, Y. Zhao, H. Shao, D. Liu, S. Liu, T. Wang, J. Li, and T. Abdelzaher, "Scheduling real-time deep learning services as imprecise computations," *IEEE 26th International Conference on Embedded and Real-Time Computing Systems and Applications*, 2020.

[59] T. Pfeil, "ItNet: iterative neural networks with tiny graphs for accurate and efficient anytime prediction," *CoRR*, vol. abs/2101.08685, 2021.

[60] M. Guan, C. Wen, M. Shan, C.-L. Ng, and Y. Zou, "Real-time event-triggered object tracking in the presence of model drift and occlusion," *IEEE Trans. Ind. Electron.*, vol. 66, no. 3, pp. 2054–2065, 2019.

[61] "IEEE standard for information technology– local and metropolitan area networks– specific requirements– part 15.1a: Wireless medium access control (mac) and physical layer (phy) specifications for wireless personal area networks (wpan)," *IEEE Std 802.15.1-2005 (Revision of IEEE Std 802.15.1-2002)*, pp. 1–700, 2005.

[62] "IEEE standard for low-rate wireless networks," *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, pp. 1–709, 2016.

[63] B. Crow, I. Widjaja, J. Kim, and P. Sakai, "Ieee 802.11 wireless local area networks," *IEEE Communications Magazine*, vol. 35, no. 9, pp. 116–126, 1997.

[64] L. Zhang, H. Gao, and O. Kaynak, "Network-induced constraints in networked control systems-a survey," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 403–416, 2013.

[65] J. M. Hendrickx and S. Martin, "Open multi-agent systems: Gossiping with random arrivals and departures," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 763–768, 2017.

[66] S. Kar and J. M. F. Moura, "Sensor networks with random links: Topology design for distributed consensus," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3315–3326, 2008.

[67] S. S. Kia, B. Van Scoy, J. Cortes, R. A. Freeman, K. M. Lynch, and S. Martinez, "Tutorial on dynamic average consensus: The problem, its applications, and the algorithms," *IEEE Control Systems Magazine*, vol. 39, no. 3, pp. 40–72, 2019.

[68] L. Ballotta, L. Schenato, and L. Carlone, "From sensor to processing networks: Optimal estimation with computation and communication latency," in *IFAC World Congress*, 2020.

[69] K. Gatsis, H. Hassani, and G. J. Pappas, "Latency-reliability tradeoffs for state estimation," *IEEE Trans. Autom. Control*, vol. 66, no. 3, pp. 1009–1023, 2021.

[70] A. Anta and P. Tabuada, "To sample or not to sample: Self-triggered control for nonlinear systems," *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2030–2042, 2010.

[71] S. Ristevski, T. Yucelen, and J. A. Muse, "An event-triggered distributed control architecture for scheduling information exchange in networked multiagent systems," *IEEE Trans. Control Syst. Technol.*, pp. 1–12, 2021.

[72] L. Hetel, C. Fiter, H. Omran, A. Seuret, E. Fridman, J. Richard, and S. I. Niculescu, "Recent developments on the stability of systems with aperiodic sampling: An overview," *Automatica*, vol. 76, pp. 309–335, 2017.

[73] M. Casares and S. Velipasalar, "Adaptive methodologies for energy-efficient object detection and tracking with battery-powered embedded smart cameras," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 10, pp. 1438–1452, 2011.

[74] H. Luo, W. Xie, X. Wang, and W. Zeng, "Detect or track: Towards cost-effective video object detection/tracking," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 8803–8810, 2019.

[75] X. Wu, K. Zhang, and C. Sun, "Optimal scheduling of multiple sensors in continuous time," *ISA Transactions*, vol. 53, no. 3, pp. 793–801, 2014.

[76] M. Sid, "Sensor scheduling strategies for fault isolation in networked control system," *ISA Transactions*, vol. 54, pp. 92–100, 2015.

[77] L. Zhao, W. Zhang, J. Hu, A. Abate, and C. J. Tomlin, "On the optimal solutions of the infinite-horizon linear sensor scheduling problem," *IEEE Trans. Automat. Contr.*, vol. 59, no. 10, pp. 2825–2830, 2014.

[78] L. Orihuela, A. Barreiro, F. Gomez-Estern, and F. R. Rubio, "Periodicity of kalman-based scheduled filters," *Automatica*, vol. 50, no. 10, pp. 2672–2676, 2014.

[79] Y. Li, C. Chen, S. Zhu, and X. Guan, "Sensor scheduling for relay-assisted wireless control systems with limited power resources," *ISA Transactions*, vol. 88, pp. 246–257, 2019.

[80] M. F. Huber, "Optimal pruning for multi-step sensor scheduling," *IEEE Trans. Automat. Contr.*, vol. 57, no. 5, pp. 1338–1343, 2012.

[81] A. B. Asghar, S. T. Jawaid, and S. L. Smith, "A complete greedy algorithm for infinite-horizon sensor scheduling," *Automatica*, vol. 81, pp. 335–341, 2017.

[82] S. Arai, Y. Iwatani, and K. Hashimoto, "Fast sensor scheduling for spatially distributed sensors," *IEEE Trans. Automat. Contr.*, vol. 56, no. 8, pp. 1900–1905, 2011.

[83] Y. Qi, P. Cheng, and J. Chen, "Optimal sensor data scheduling for remote estimation over a time-varying channel," *IEEE Trans. Automat. Contr.*, vol. 62, no. 9, pp. 4611–4617, 2017.

[84] A. Sen, S. R. Sahoo, and M. Kothari, "Distributed algorithm for higher-order integrators to track average of unbounded signals," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 2903–2908, 2020. 21st IFAC World Congress.

[85] J. George and R. Freeman, "Robust dynamic average consensus algorithms," *IEEE Transactions on Automatic Control*, vol. 64, no. 11, pp. 4615–4622, 2019.

[86] W. Perruquetti and J. P. Barbot, *Sliding Mode Control in Engineering*. USA: Marcel Dekker, Inc., 2002.

[87] H. Jin and S. Sun, "Distributed filtering for multi-sensor systems with missing data," *Information Fusion*, 2022.

[88] S.-L. Sun, "Multi-sensor optimal fusion fixed-interval kalman smoothers," *Information Fusion*, vol. 9, no. 2, pp. 293–299, 2008.

[89] A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, "Information weighted consensus filters and their application in distributed camera networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 12, pp. 3112–3125, 2013.

[90] R. Olfati-Saber, "Distributed kalman filter with embedded consensus filters," in *IEEE Conference on Decision and Control*, pp. 8179–8184, 2005.

[91] R. Olfati-Saber, "Distributed kalman filtering for sensor networks," in *IEEE Conference on Decision and Control*, pp. 5492–5498, 2007.

[92] S. Wang and W. Ren, "On the convergence conditions of distributed dynamic state estimation using sensor networks: A unified framework," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 4, pp. 1300–1316, 2018.

[93] X. He, W. Xue, and H. Fang, "Consistent distributed state estimation with global observability over sensor network," *Automatica*, vol. 92, pp. 162–172, 2018.

[94] E. Sebastian, E. Montijano, and C. Sagues, "All-in-one: Certifiable optimal distributed kalman filter under unknown correlations," in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 6578–6583, 2021.

[95] H. Niederwieser, M. Tranninger, R. Seeber, and M. Reichhartinger, "Unknown input observer design for linear time-invariant multivariable systems based on a new observer normal form," *International Journal of Systems Science*, vol. 53, no. 10, pp. 2180–2206, 2022.

[96] A. Levant, "Higher-order sliding modes, differentiation and output-feedback control," *International Journal of Control*, vol. 76, pp. 924–941, 2003.

[97] M. Zhu and S. Martinez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.

[98] E. Montijano, J. I. Montijano, C. Sagues, and S. Martinez, "Robust discrete time dynamic average consensus," *Automatica*, vol. 50, no. 12, pp. 3131–3138, 2014.

[99] S. S. Kia, J. Cortes, and S. Martinez, "Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication," *Automatica*, vol. 55, pp. 254–264, 2015.

[100] Y. Zhao, C. Xian, G. Wen, P. Huang, and W. Ren, "Design of distributed event-triggered average tracking algorithms for homogeneous and heterogeneous multiagent systems," *IEEE Transactions on Automatic Control*, vol. 67, no. 3, pp. 1269–1284, 2022.

[101] S. S. Kia, J. Cortés, and S. Martínez, "Distributed event-triggered communication for dynamic average consensus in networked systems," *Automatica*, vol. 59, no. C, pp. 112–119, 2015.

[102] L. Shi, L. Chen, and Y. Cheng, "High-order bipartite consensus for multiagent systems over signed networks subject to asynchronous communications," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 4, pp. 3325–3334, 2021.

[103] H. Xia and Q. Dong, "Dynamic leader-following consensus for asynchronous sampled-data multi-agent systems under switching topology," *Information Sciences*, vol. 514, pp. 499–511, 2020.

[104] C. Deng, W.-W. Che, and Z.-G. Wu, "A dynamic periodic event-triggered approach to consensus of heterogeneous linear multiagent systems with time-varying communication delays," *IEEE Transactions on Cybernetics*, vol. 51, no. 4, pp. 1812–1821, 2021.

[105] X. Wu, Y. Tang, J. Cao, and W. Zhang, "Distributed consensus of stochastic delayed multi-agent systems under asynchronous switching," *IEEE Transactions on Cybernetics*, vol. 46, no. 8, pp. 1817–1827, 2016.

[106] H. Zhang, J. Zhang, Y. Cai, S. Sun, and J. Sun, "Leader-following consensus for a class of nonlinear multiagent systems under event-triggered and edge-event triggered mechanisms," *IEEE Transactions on Cybernetics*, vol. 52, no. 8, pp. 7643–7654, 2022.

[107] R. Aldana-Lopez, E. Sebastian, R. Aragues, E. Montijano, and C. Sagues, "Distributed outer approximation of the intersection of ellipsoids," *IEEE Control Systems Letters*, pp. 1–1, 2023.

[108] D. Liberzon, *Switching in Systems and Control*. Systems & control, Birkhauser, 2003.

[109] R. Shorten, F. Wirth, O. Mason, K. Wulff, and C. King, "Stability criteria for switched and hybrid systems," *SIAM Rev.*, vol. 49, pp. 545–592, 2007.

[110] J. Ding, Y. Peres, G. Ranade, and A. Zhai, "When multiplicative noise stymies control," *ArXiv*, vol. abs/1612.03239, 2016.

[111] A. A. Ahmadi, R. Jungers, P. Parrilo, and M. Roozbehani, "Joint Spectral Radius and Path-Complete Graph Lyapunov Functions," *SIAM J. Control. Optim.*, vol. 52, pp. 687–717, 2014.

[112] W. Griggs, C. King, R. Shorten, O. Mason, and K. Wulff, "Quadratic Lyapunov functions for systems with state-dependent switching," *Linear Algebra Appl.*, vol. 433, pp. 52–63, 2010.

[113] F. Blanchini and C. Savorgnan, "Stabilizability of switched linear systems does not imply the existence of convex Lyapunov functions," *Automatica*, vol. 44, no. 4, pp. 1166–1170, 2008.

[114] D. Antunes and W. P. M Heemels, "Linear quadratic regulation of switched systems using informed policies," *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 2675–2688, 2017.

[115] M. Fiacchini and M. Jungers, "Necessary and sufficient condition for stabilizability of discrete-time linear switched systems: A set-theory approach," *Automatica*, vol. 50, pp. 75–83, 2014.

[116] M. Fiacchini, R. Jungers, and A. Girard, "Stabilization and control Lyapunov functions for language constrained discrete-time switched linear systems," *Automatica*, vol. 93, pp. 64–74, 2018.

[117] J. C. Geromel and P. Colaneri, "Stability and stabilization of discrete time switched systems," *Int. J. Control*, vol. 79, no. 7, pp. 719–728, 2006.

[118] Z. Wu and Q. He, "Optimal switching sequence for switched linear systems," *SIAM J. Control. Optim.*, vol. 58, pp. 1183–1206, 2020.

[119] K. Åström, *Introduction to Stochastic Control Theory*. Mathematics in science and engineering, Academic Press, 1970.

[120] F. Kubler, P. Renner, and K. Schmedders, "Chapter 11 - computing all solutions to polynomial equations in economics," in *Handbook of Computational Economics* (K. Schmedders and K. L. Judd, eds.), vol. 3 of *Handbook of Computational Economics*, pp. 599–652, Elsevier, 2014.

[121] J. Verschelde, "Algorithm 795: Phcpack: A general-purpose solver for polynomial systems by homotopy continuation," *ACM Trans. Math. Softw.*, vol. 25, no. 2, pp. 251–276, 1999.

[122] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[123] R. A. Horn and C. R. Johnson, *Matrix Analysis*. USA: Cambridge University Press, 2nd ed., 2012.

[124] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 2nd ed., 2000.

[125] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE ICRA*, pp. 2520–2525, 2011.

[126] M. Faessler, A. Franchi, and D. Scaramuzza, "Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 620–626, 2018.

[127] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[128] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*. Springer Publishing Company, Incorporated, 5th ed., 2012.

[129] H. K. Khalil, *Nonlinear systems*. Upper Saddle River, NJ: Prentice-Hall, 3rd ed., 2002.

[130] T. Soderstrom, *Discrete-Time Stochastic Systems: Estimation and Control*. Berlin, Heidelberg: Springer-Verlag, 2nd ed., 2002.

[131] Y.-M. Song, K. Yoon, Y.-C. Yoon, K. C. Yow, and M. Jeon, "Online multi-object tracking with gmphd filter and occlusion group management," *IEEE Access*, vol. 7, pp. 165103–165121, 2019.

[132] X. Dong, J. Shen, D. Yu, W. Wang, J. Liu, and H. Huang, "Occlusion-aware real-time object tracking," *IEEE Trans. Multimedia*, vol. 19, no. 4, pp. 763–771, 2017.

[133] A. Loquercio, M. Segu, and D. Scaramuzza, "A general framework for uncertainty estimation in deep learning," *IEEE Rob. Autom. Lett.*, vol. 5, no. 2, pp. 3153–3160, 2020.

[134] L. A. Fokin and A. G. Shchipitsyn, "Innovation-based adaptive kalman filter derivation," in *2009 International Siberian Conference on Control and Communications*, pp. 318–323, 2009.

[135] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artif. Intell.*, vol. 293, p. 103448, 2021.

[136] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," *ArXiv*, 2016. arXiv: 1603.00831.

[137] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2." `https://github.com/facebookresearch/detectron2`, 2019.

[138] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Omni-scale feature learning for person re-identification," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3701–3711, 2019.

[139] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *Computer Vision – ECCV 2016 Workshops* (G. Hua and H. Jégou, eds.), (Cham), pp. 850–865, Springer International Publishing, 2016.

[140] J. B. Moore and B. D. O. Anderson, "Coping with singular transition matrices in estimation and control stability theory," *Int. J. Control*, vol. 31, no. 3, pp. 571–586, 1980.

[141] A. Alessandri and P. Coletta, "Switching observers for continuous-time and discrete-time linear systems," *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, vol. 3, pp. 2516–2521 vol.3, 2001.

[142] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[143] D. Gomez-Gutierrez, C. R. Vazquez, S. Celikovsky, J. D. Sanchez-Torres, and J. Ruiz-Leon, "On finite-time and fixed-time consensus algorithms for dynamic networks switching among disconnected digraphs," *International Journal of Control*, vol. 93, no. 9, pp. 2120–2134, 2020.

[144] J. Alonso-Mora, E. Montijano, T. Nägeli, O. Hilliges, M. Schwager, and D. Rus, "Distributed multi-robot formation control in dynamic environments," *Autonomous Robots*, vol. 43, no. 5, pp. 1079–1100, 2019.

[145] M. Zhu and S. Martinez, "On distributed convex optimization under inequality and equality constraints," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 151–164, 2012.

[146] R. Aragues, C. Sagues, and Y. Mezouar, "Feature-based map merging with dynamic consensus on information increments," *Autonomous Robots*, vol. 38, no. 3, pp. 243–259, 2015.

[147] A. Cherukuri and J. Cortes, "Distributed generator coordination for initialization and anytime optimization in economic dispatch," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 3, pp. 226–237, 2015.

[148] Y. Zhao, Y. Liu, G. Wen, and T. Huang, "Finite-time distributed average tracking for second-order nonlinear systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 6, pp. 1780–1789, 2019.

[149] R. A. Freeman, P. Yang, and K. M. Lynch, "Stability and convergence properties of dynamic average consensus estimators," in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 338–343, 2006.

[150] S. Nosrati, M. Shafiee, and M. B. Menhaj, "Dynamic average consensus via nonlinear protocols," *Automatica*, vol. 48, no. 9, pp. 2262–2270, 2012.

[151] S. Y. Chen, "Kalman filter for robot vision: A survey," *IEEE Trans. Ind. Electron.*, vol. 59, no. 12, pp. 4409–4420, 2012.

[152] S. Ghapani, S. Rahili, and W. Ren, "Distributed average tracking of physical second-order agents with heterogeneous unknown nonlinear dynamics without constraint on input signals," *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 1178–1184, 2019.

[153] S. Rahili and W. Ren, "Heterogeneous distributed average tracking using nonsmooth algorithms," in *2017 American Control Conference (ACC)*, pp. 691–696, 2017.

[154] Y. Zhao, Y. Liu, Z. Li, and Z. Duan, "Distributed average tracking for multiple signals generated by linear dynamical systems: An edge-based framework," *Automatica*, vol. 75, pp. 158–166, 2017.

[155] J. Cortes, "Discontinuous dynamical systems," *IEEE Control Systems Magazine*, vol. 28, no. 3, pp. 36–73, 2008.

[156] R. Gama and G. Smirnov, "Stability and optimality of solutions to differential inclusions via averaging method," *Set-Valued and Variational Analysis*, vol. 22, 2014.

[157] E. Cruz-Zavala and J. A. Moreno, "Levant's arbitrary-order exact differentiator: A Lyapunov approach," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 3034–3039, 2019.

[158] C. Godsil and G. Royle, *Algebraic Graph Theory*, vol. 207 of *Graduate Texts in Mathematics*. Springer, 2001.

[159] E. Bernuau, D. Efimov, W. Perruquetti, and A. Polyakov, "On homogeneity and its application in sliding mode control," *Journal of the Franklin Institute*, vol. 351, no. 4, pp. 1866–1901, 2014. Special Issue on 2010-2012 Advances in Variable Structure Systems and Sliding Mode Algorithms.

[160] W. Ren and U. M. Al-Saggaf, "Distributed kalman-bucy filter with embedded dynamic averaging algorithm," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1722–1730, 2018.

[161] L. K. Vasiljevic and H. K. Khalil, "Error bounds in differentiation of noisy signals by high-gain observers," *Syst. Control. Lett.*, vol. 57, pp. 856–862, 2008.

[162] A. Sen, S. R. Sahoo, and M. Kothari, "Distributed average tracking with incomplete measurement under a weight-unbalanced digraph," *IEEE Transactions on Automatic Control*, vol. 67, no. 11, pp. 6025–6037, 2022.

[163] D. Bernstein and S. Bhat, "Geometric homogeneity with application to finite-time stability," *Mathematics of Control, Signals, and Systems*, vol. 17, 2005.

[164] W. Niehsen, "Information fusion based on fast covariance intersection filtering," in *International Conference on Information Fusion.*, vol. 2, pp. 901–904 vol.2, 2002.

[165] R. Aragues, C. Sagues, and Y. Mezouar, "Feature-based map merging with dynamic consensus on information increments," in *IEEE International Conference on Robotics and Automation*, pp. 2725–2730, 2013.

[166] M. Iqbal, Z. Qu, and A. Gusrialdi, "Resilient dynamic average-consensus of multi-agent systems," *IEEE Control Systems Letters*, vol. 6, pp. 3487–3492, 2022.

[167] H. Moradian and S. S. Kia, "On robustness analysis of a dynamic average consensus algorithm to communication delay," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 2, pp. 633–641, 2019.

[168] R. Goebel, R. G. Sanfelice, and A. R. Teel, *Hybrid Dynamical Systems: Modeling, Stability, and Robustness.* Princeton University Press, 2012.

[169] K. Liu, A. Selivanov, and E. Fridman, "Survey on time-delay approach to networked control," *Annual Reviews in Control*, vol. 48, pp. 57–79, 2019.

[170] L. Shi, J. Shao, M. Cao, and H. Xia, "Asynchronous group consensus for discrete-time heterogeneous multi-agent systems under dynamically changing interaction topologies," *Information Sciences*, vol. 463-464, pp. 282–293, 2018.

[171] C. Pilotto, K. M. Chandy, and J. White, "Consensus on asynchronous communication networks in presence of external input," in *49th IEEE Conference on Decision and Control (CDC)*, pp. 3838–3844, 2010.

[172] F. Caccavale and F. Pierri, "An approach to distributed estimation of time-varying signals by multi-agent systems," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 3556–3561, 2022.

[173] O. Gurewitz, I. Cidon, and M. Sidi, "One-way delay estimation using network-wide measurements," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2710–2724, 2006.

[174] A. F. Filippov, *Differential equations with discontinuous righthand sides*, vol. 18 of *Mathematics and its Applications (Soviet Series)*. Dordrecht: Kluwer Academic Publishers Group, 1988. Translated from the Russian.

[175] M. Livne and A. Levant, "Proper discretization of homogeneous differentiators," *Automatica*, vol. 50, no. 8, pp. 2007–2014, 2014.

[176] G. Carnevale, F. Farina, I. Notarnicola, and G. Notarstefano, "Gtadam: Gradient tracking with adaptive momentum for distributed online optimization," *IEEE Transactions on Control of Network Systems*, pp. 1–12, 2022.

[177] G. Carnevale, I. Notarnicola, L. Marconi, and G. Notarstefano, "Triggered gradient tracking for asynchronous distributed optimization," *Automatica*, vol. 147, p. 110726, 2023.

[178] R. T. Rockafellar, *Convex analysis.* Princeton Mathematical Series, Princeton, N. J.: Princeton University Press, 1970.

[179] W. Rudin, *Principles of mathematical analysis / Walter Rudin.* McGraw-Hill New York, 3d ed. ed., 1976.

[180] G. Hardy, J. Littlewood, and G. Pólya, *Inequalities.* Cambridge Mathematical Library, Cambridge University Press, 1988.

[181] F. Arscott and A. Filippov, *Differential Equations with Discontinuous Righthand Sides: Control Systems.* Mathematics and its Applications, Springer Netherlands, 1988.