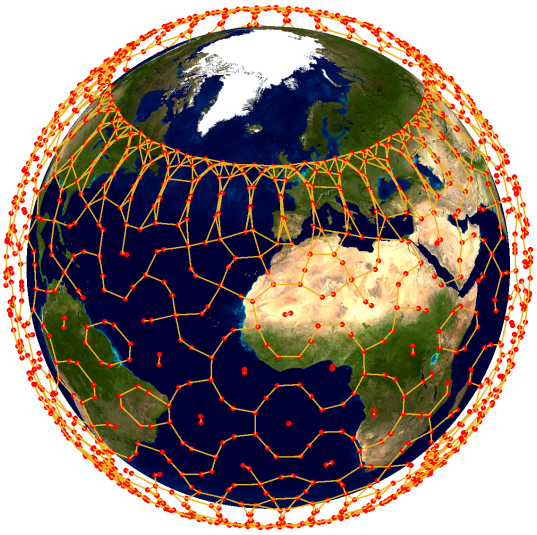
TÉCNICO LISBOA



Distributed decentralized control for very large-scale systems with application to LEO satellite mega-constellations

Leonardo Pedroso Duarte

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisor: Prof. Pedro Tiago Martins Batista

Examination Committee

Chairperson: Prof. José Fernando Alves da Silva

Supervisor: Prof. Pedro Tiago Martins Batista

Member of the Committee: Prof. Paulo Jorge Coelho Ramalho Oliveira

November 2022

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

This thesis marks the end of five years of the roller-coaster that is graduating from IST. Over this time, I had the pleasure of crossing paths with so many interesting people, some of which left their mark and shaped who I am today.

First and foremost, my deepest gratitude goes to Professor Pedro Batista. I could not have asked for a better opportunity to get my hands on academic research. His brilliance does not overshadow his kindness, integrity, and passion for teaching. I will always be grateful for his guidance and advice, for believing in my potential, and for the time and effort he has put into our research endeavors.

Second, I am also very thankful for the financial support of Fundação para a Ciência e a Tecnologia (FCT) through LARSyS - FCT Project UIDB/50009/2020 and through the FCT project DECENTER [LISBOA-01-0145-FEDER-029605], funded by the Programa Operacional Regional de Lisboa 2020 and PIDDAC programs.

Third, the friendship and support of my friends during this journey cannot be overstated. I would like to thank Haidar, Iara, Jafar, Janeiro, Mário, Renata, and Tiago for the good times we spent together. I feel very lucky for having met them. A special word goes to my friends Miguel and Miguel, with whom I had the opportunity to share some great moments. I appreciate their willingness to join me in so many adventures over the past five years. Lastly, my greatest thanks goes to Alice for her companionship, for putting up with my endless jokes, and for cheering me up.

Finally, I have the deepest appreciation for the support of my family. I cherish all the moments with Bea, my sister, who joined me at IST. She made the whole journey much more enjoyable. I appreciate the wise advice of my grandparents and great-grandmother. I am especially thankful to my mother for her love, sacrifice for my well-being, and effort to cultivate my interest in science.

Resumo

As vantagens e o potencial de redes complexas de sistemas interconectados em grande escala são inquestionáveis numa infinidade de áreas de engenharia, não apenas como oportunidades de investimento, mas também como uma transição natural no sentido da melhoria da eficiência, robustez e escalabilidade. Concretamente, as megaconstelações de satélites prometem revolucionar tanto o futuro das comunicações quanto a observação e monitorização da Terra. Ainda que estejam em curso diligências conducentes à operacionalização destas soluções, as atuais tecnologias de comando remoto individual, aplicadas há décadas, comprometem a viabilidade funcional e financeira de tais empreendimentos. Assim, o objetivo da presente tese decorre desta lacuna no estado da arte. Primeiro, o problema de controlo distribuído e descentralizado é formulado numa estrutura de horizonte recuante, a que acrescem restrições de exequibilidade em larga escala. Segundo, é proposto um procedimento de relaxação convexa para aproximar a solução ótima do problema de síntese de um regulador num ambiente descentralizado, validado com recurso a uma simulação numérica de larga escala e a resultados experimentais. Adicionalmente, propõe-se uma solução de seguimento. Terceiro, é desenvolvida uma nova solução de controlo, distribuída e descentralizada, para o caso particular de sistemas dinamicamente desacoplados. A síntese do controlador é distribuída pela rede introduzindo um aproximação e um procedimento de agendamento de operações, além de técnicas de relação convexa, para cumprir as restrições de exequibilidade em grande escala. Quarto, o potencial da solução para o controlo cooperativo de órbita da megaconstelação Starlink é ilustrado, com sucesso, num propagador de órbita de alta fidelidade.

Keywords: Controlo Descentralizado, Sistemas Distribuídos de Robôs, Controlo em Rede, Robótica Espacial, Agentes Autónomos, Megaconstelações

Abstract

The advantages and tremendous potential of very large-scale complex networks of interconnected systems are indisputable in a myriad of engineering fields, not only as business opportunities but also as a natural change towards efficiency, reliability, and scalability. In particular, mega-constellations of satellites promise to revolutionize the future of communications and Earth observation and monitoring. Although efforts towards the deployment of these solutions are underway, decades-old tried and tested individual ground-based tracking telemetry and command technologies condemn these ventures to practical unfeasibility and economic unviability. The goal of this thesis follows from the self-evident void in the state-of-the-art, aiming to bring these endeavors to fruition. First, the distributed and decentralized control problem is formulated in a receding horizon control framework alongside the severe large-scale feasibility constraints. Second, a convex relaxation procedure is proposed to approximate the optimal solution of the regulator synthesis problem in a decentralized setting, which is validated resorting to a large-scale numeric simulation and experimental results. Moreover, a tracking solution is put forward. Third, a novel distributed and decentralized networked control solution is developed for the particular case of dynamically decoupled systems. The controller synthesis computations are distributed across the network leveraging the proposed convex relaxation, an approximation, and a scheduling procedure, to comply with the feasibility constraints on a very large-scale. Fourth, the potential of the proposed solution is successfully illustrated for the cooperative on-board orbit control of the Starlink mega-constellation. The shape-keeping task is formulated in a novel framework with emphasis on efficiency and fuel saving.

Keywords: Decentralized Control, Distributed Robot Systems, Networked Control, Space Robotics, Autonomous Agents, Mega-constellations

Contents

List of Tables

List of Figures

Nomenclature

Greek symbols

$\boldsymbol{\lambda}$	Langrange multipliers vector

Roman symbols

N	Number of systems
H	Length of finite window
d	Number of gains used for the actuation in each finite window
$\mathcal{G}_d$	Dynamic couplings directed graph
$\mathcal{G}_o$	Tracking output couplings directed graph
$\mathcal{G}_c$	State feedback communication directed graph
J	Cost function
V	Cost-to-go

Subscripts

∞	Infinite horizon

Superscripts

T	Transpose

List of Acronyms

CAD	Computer aided design
ECI	Earth centered inertial
EKF	Extended Kalman filter
FCT	Fundação para a Ciência e a Tecnologia
GEO	Geosynchronous equatorial orbit
GNSS	Global navigation satellite system
ISL	Inter-satellite link
LEO	Low Earth orbit
LMI	Linear matrix inequality
LQR	Linear quadratic regulator
LTI	Linear time-varying
LTV	Linear time-invariant
MAE	Mean absolute error
MCC	Mission control center
MEO	Medium Earth orbit
PCB	Printed circuit board
RHC	Receding horizon control
TDB	Dynamical barycentric time (temps dynamique barycentrique)
TT&C	Tracking telemetry and command
TUDAT	TU Delft Astrodynamics Toolbox
UAV	Unmanned aerial vehicle
USB	Universal Serial Bus
iLQR	Iterative linear quadratic regulator

Chapter 1

Introduction

1.1 Motivation

Driven by the ambition to adapt to the environment, overcome limitations, and thrive, innovation is deeply rooted in the nature of the human species. Automatic control is born and keeps evolving from this natural drive. The first documented use of automatic control is a water clock mechanism designed in ancient Greece by Ktesibios (285–222 BC). Markings in a slowly draining water tank were used to quantify the passage of time. Ktesibios employed a conic floating valve that is displaced by the water level in a tank to control and maintain a constant water outflow, which enabled unprecedented timekeeping accuracy. In the 17th century, the fly-ball governor, which uses a mechanical feedback control system, was invented. This groundbreaking technological advance was employed to regulate the speed of steam engines, key to the unprecedented production and population growth witnessed during the industrial revolution. In the 1960s, the advances in digital computing and the seminal work of Bellman and Pontryagin in optimal control culminated in the automatic landing in the Moon in 1969.

Over the past several decades, the fast-paced development of new sensors, robot technology, and control algorithms has enabled the automation of numerous tasks. Some were formerly carried out by humans, while others were previously unimaginable. The productivity, precision, consistency, robustness, and ability to operate in harsh environments have been the main driving factors of this trend. For instance, mapping the magnetic field of the Earth, analyzing the soil composition of Mars, and manufacturing mechanical parts with micrometre precision would be very challenging without the use of robotic systems equipped with cutting-edge sensors, actuators, and control algorithms. Up until very recently, to automate a particular task, few, oftentimes large, multi-purpose, very complex systems were designed. Consider, as an example, the task of seabed mapping. It is carried out making use of either a complex sonar system mounted on a vessel or by towing a single vehicle equipped with a cutting-edge underwater sonar system. Nevertheless, the inefficiency in covering extensive areas, the lack of robustness to the failure of one of the many components that make up these complex systems, and the little flexibility to adapt to changes in the environment have been major concerns that hinder the ability to perform tasks on a very large-scale.

On one hand, bearing in mind the aforementioned shortcomings, the field of swarm robotics has arisen with a heavy inspiration in nature [1–3]. This novel framework aims to achieve tasks that are beyond the capabilities of a single agent, by employing a large number of simple autonomous agents collaboratively working towards a common goal, which mimics the behavior of ant colonies, bird flocks, and fish schools. The potential of this framework is driven by three pivotal properties: i) Robustness, since the failure of an agent does not compromise the swarm; ii) Flexibility, since different agent coordination strategies can be employed in response to changes in the environment; and iii) Scalability, since the throughput of the task can be increased by deploying additional agents. On the other hand, there are several tasks that require the automation of large-scale complex and spatially scattered processes. A common example is the task of controlling the distribution of resources from spatially distributed sources to different users under varying demands, for example, the problem of the distribution of electrical power. Both robotic swarms and large-scale spatially distributed processes, which some authors also designate as multi-agent systems, can be modeled as large-scale networks of interconnected systems with intricate dynamics and complex interactions. Therefore, henceforth, robotic swarms and large-scale networks of spatially distributed processes are both analysed as very large-scale systems. Instead of resisting the distributed nature of these frameworks by designing a single, very complex, expensive system, which requires the transmission of data over long distances, one could embrace it and exploit its compelling properties. This approach is designated by decentralized control and it has been a hot topic for several decades, given its widespread applicability in a broad range of engineering fields.

A plethora of applications of decentralized control have been conceptualized for these frameworks. In the field of swarm robotics, examples of applications are unmanned aircraft formation flight [4, 5], unmanned underwater formations [6, 7], satellite formation control [8, 9], precision agriculture [10], firefighting [11], surveillance [12], light shows [13], and exploration and navigation on Mars [14]. For large-scale networks of spatially scattered processes, there are also pressing applications such as irrigation networks [15, 16] and traffic networks [17], on which work that precedes the developments of this thesis has been carried out by the author [18]. Notwithstanding, one of the most promising applications has to be power distribution networks [19, 20]. Indeed, with the increasing proliferation of smaller renewable energy sources, the number of sources and sinks of power grids is booming and spreading geographically [21]. This change calls for network-wise management and efficient distribution in a smart grid. It is in line with an acceleration of the energy efficiency improvement rate and transition towards renewable energy sources for decarbonisation called for by the United Nations. The emergence of these applications has accelerated the demand for efficient decentralized algorithms.

Despite the very compelling robustness, flexibility, and scalability properties of swarm robotics and very large-scale complex networks of spatially distributed processes and the myriad of envisioned high-impact applications that rely on them, most of the given examples are: i) yet to transition from conceptualization to deployment; ii) deployed only in very controlled environments as a proof of concept; or iii) implemented in practice in a very small scale with a small number of agents. The reason is clear: there are inhibiting technical challenges, especially regarding the feasibility and economic viability of the implementation of state-of-the-art algorithms to these very large-scale systems. One remarkable ex-

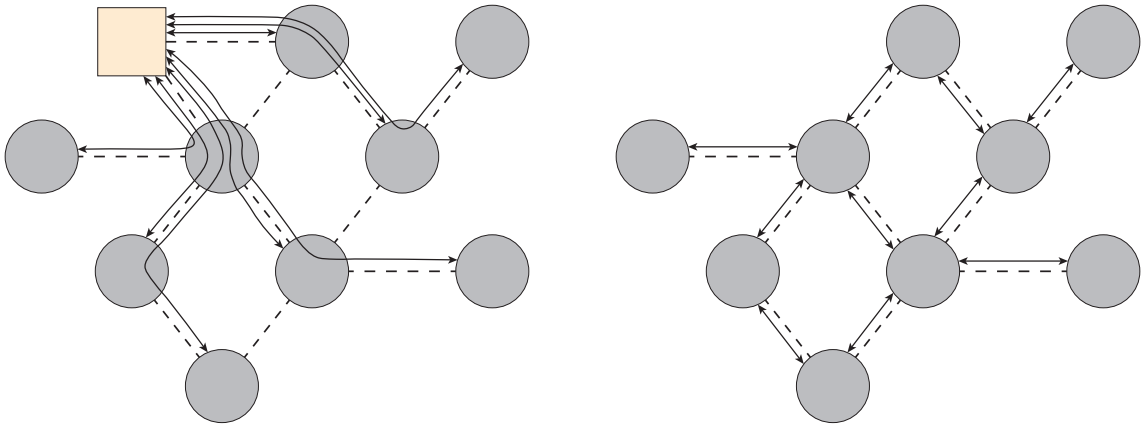
ception is the development of large-scale low Earth orbit (LEO) constellations. However, although some prototypes have already started being deployed on a large-scale, their economical viability is doomed unless control algorithms befitting the challenges of such large-scale systems are developed.

In what follows, the motivation for this work is further addressed from two angles. First, the technological challenges behind the implementation feasibility of control algorithms over very large-scale systems are analyzed. Second, the potential impact and feasibility challenges of LEO mega-constellations are discussed.

1.1.1 Technical challenges

It is well established that as the scale of the systems increases, the classical control solutions, which are developed in a centralized framework, eventually become infeasible to implement in practice. This effect is commonly designated as the curse of dimensionality. The classical centralized control frameworks require: i) infrastructure for the centralized coordination; ii) the transmission of a large amount of information between every agent and the central node; and iii) serious computational power for real-time processing in the central node. In these centralized configurations, each system must transmit local information to a central computational unit. This central node processes the global data, received from each system that makes up the network, to compute a control solution, which is posteriorly transmitted to every system in the network. The communication protocol of a centralized solution can rely on data transmission via a path of several physical communication links. Nevertheless, the information has to be retransmitted via several systems to the central unit, which receives information from all nodes. As the dimension of the network increases, the load on the communication links, the communication delays, the complexity of the protocol, and the computational load of the central computational unit increase. For these reasons, the sheer communication pressure and computational load in the central unit render its implementation infeasible as the dimension of the network increases. Moreover, these strategies offer little robustness to failure of the central processing node or the communication infrastructure. This challenge, well-known for decades, has been extensively addressed, and many alternatives have been proposed in a decentralized control framework. In fact, decentralized solutions rely on local computations and local communication to overcome the curse of dimensionality. No central computing unit is required and, at no moment in time, any entity in the network has knowledge about the global state of the network. Even though local communication relies on the same physical data transmission links that may be used in a centralized configuration, the decisive difference is the amount of information that needs to be handled at the protocol level. In a decentralized paradigm, only local information is transmitted in each physical communication link, which does not scale as the dimension of the network increases. Figure 1.1 depicts a scheme of the comparison between the communication requirements in centralized and decentralized configurations. The sought-after scalability of the latter is achieved, nonetheless, at the expense of increased complexity of the control algorithms and oftentimes sub-optimal performance.

The decentralized control and estimation problems can be formulated as optimization problems subject to constraints that arise from the decentralized nature of the network. Despite the large research



(a) Centralized. (b) Decentralized.

Figure 1.1: Configuration of networked systems: physical links (dashed lines) and information flow (solid lines).

effort in this field, it remains an open problem even for linear time-invariant (LTI) systems due to its intractability [22]. On top of that, a significant portion of the envisioned large-scale applications has underlying nonlinear dynamics, for which befitting solutions are even scarcer. Oftentimes, linearization techniques are employed, approximating the dynamics of the nonlinear system over successive operating points [23]. This approach yields a more tractable linear time-varying (LTV) system for which decentralized control strategies have to be designed. Nevertheless, when it comes to the implementation of time-varying decentralized algorithms over very large-scale networks, more challenges are brought to light in addition to the intricacies of the decentralized problem. As a result, the inevitable change from large to very large-scale networked control systems calls for a consistent paradigm revolution from a control standpoint. It is impossible to benefit from the robustness, flexibility, and scalability properties of this novel architecture, if its implementation to real-life applications is unfeasible. Likewise, its economic viability must not be jeopardized by the lack of befitting algorithms that take into account the challenges imposed by the dimension of the network. For these reasons, heavy constrains at the: i) topological, ii) communication, iii) synthesis, iv) computational, and v) memory level must be enforced to enable a seamless practical implementation.

First, the topology of a network is characterized by the existence of couplings of some nature between the systems that make up the network. For instance, if the local goal of a mobile robot represented by system $\mathcal{S}_i$ is to follow another robot $\mathcal{S}_j$, then it is characterized by a tracking coupling of $\mathcal{S}_i$ with $\mathcal{S}_j$. It is assumed that the number of systems with which each system has couplings does not scale with the number of systems in the whole network. Note that this assumption is embedded in the definition of swarm robotics and is inherent to large-scale spatially distributed processes. Therefore, it should be regarded as a property, which should be cleverly exploited, rather than a constraint that is imposed.

Second, to seek a scalable solution, the number of communication links established with each system must not scale with the number of systems in the network. As it was previously pointed out, these limitations are addressed at the protocol level, i.e., the restrictions are applied to the exchange of data

between systems, not on the physical communication links. Moreover, data transmissions between systems cannot be considered instantaneous.

Third, the design of a control solution usually entails a synthesis stage with the goal of optimizing the controller parameters for a particular task. On one hand, solutions for time-invariant dynamics and topology can be synthesized prior to deployment, oftentimes referred to as offline synthesis. On the other hand, if either the network topology or the network dynamics are time-varying, then the synthesis procedure has to be carried out with a certain periodicity in real-time. Therefore, the synthesis procedure must be carried out cooperatively distributing the load of the overall algorithm evenly among the systems that make up the network.

Fourth, computational and memory resources available on-board each system are very limited. The computational load of the real-time implementation of the cooperative control algorithms must be distributed across all agents in such a way that the replication of computations among agents is reduced to a minimum, which circumvents the curse of dimensionality. Thus, the computational complexity and data storage requirements of the algorithms running on-board each system must not scale with the number of systems in the whole network.

Regarding these aspects, an important distinction is made in this thesis between the terms distributed and decentralized. Although these are, oftentimes, used interchangeably by many authors, in this thesis they characterize control algorithms with different characteristics. Herein, a control solution is said to be decentralized if the implementation of the control law in each system can be deployed resorting to local communication exclusively. In contrast, a control solution is said to be distributed if its control law can be synthesized in real-time in a distributed manner across the systems of the network resorting to local communication exclusively. For instance, if a control solution resorts to a local linear feedback in each system, but the gains of such control law have to be synthesized in real time globally in a single node or replicated in each system of the network, the control solution is decentralized but not distributed. Thus, the inevitable paradigm change towards very large-scale networks calls for distributed and decentralized control algorithms.

1.1.2 LEO mega-constellations

The conceptualization of large LEO constellations began in the 1990s, with Globalstar, Iridium, Odyssey, and Teledesic, as an attempt to provide communication services globally. Nevertheless, none of these projects but Iridium succeeded. They either filled for bankruptcy after deployment, were aborted, or were significantly downscaled. The reason for their collapse was the reduced demand for these services and the high deployments costs, which rendered these projects inviable [24]. Despite that, the technological advances and soaring demand for broadband connectivity that were witnessed in the following two decades led to a reawakening of LEO mega-constellation projects. These come as a complement to the already established geosynchronous equatorial orbit (GEO) satellite communications. On one hand, GEO communications proved, over the past several decades, to be reliable and have high coverage, due to their high orbit. On the other hand, a single LEO satellite has much less coverage,

thus requiring several satellites in a constellation to provide global connectivity. Nevertheless, due to their lower orbit, these constellations: i) are able to reduce the latency by more than one order of magnitude; ii) require significantly less power for effective communication; iii) provide coverage to areas near the poles of the Earth, which are not reliably provided by GEO communications providers; iv) allow for a high channel reusability, which increases capacity by two to three orders of magnitude in relation to GEO systems; and v) allow for active power control, unlike GEO or medium Earth orbit (MEO) communication systems, using a narrow beam antenna on the receiver [25, 26]. For a more detailed and insightful analysis on LEO communication systems and their advantages over MEO and GEO communication systems, see [25]. Large-scale constellations of satellites in LEO are, with the current technology, unarguably a solution to meet the increasing demand for reliable low-latency, high capacity, global broadband connectivity. Indeed, if these services match or beat the current price of existing high-speed options, there are projections that indicate that, at an affordable cost-per-engagement, the consumer market demands for these systems could soar [24]. Nevertheless, there are still substantial hurdles to the success of the new generation of LEO constellation communication systems. These hurdles are addressed in [24, 27]. Not surprisingly, the greatest concern is their economical viability. In fact, for the success of the recently proposed solutions, the costs across the value chain must be reduced, namely: i) satellite manufacturing; ii) launch; iii) ground equipment and infrastructure; iv) user equipment; v) operating costs; and vi) disposal services.

In recent years, four promising LEO constellation projects started development and testing, namely: i) Telsat's Telesat Lightspeed; ii) OneWeb; iii) SpaceX's Starlink; and iv) Amazon's Project Kuiper. The projected characteristics of these constellations are detailed in [28, 29]. It is possible to note a paradigm shift from the use of constellations of a small number of highly complex satellites to the employment of a very large number of smaller and simpler satellites that cooperate in large-scale networks. In fact, the term mega-constellation has been coined to designate these very large-scale constellations. For instance, the Starlink constellation is projected to feature 11926 satellites in LEO over 8 shells, of which over 3000 satellites have already been launched. Nevertheless, the aforementioned paradigm change has not yet been accompanied by a paradigm change from an operations standpoint. As pointed out in [27], the tracking telemetry and command (TT&C) system projected for these constellations does not differ from the TT&C system architecture employed for a single satellite. This system consists of a single centralized mission control center (MCC) with several ground terminals scattered across the globe to allow for continuous monitoring of the whole constellation. However, the implementation of such a centralized architecture in a mega-constellation is very challenging and expensive, because of the dimension of the network. First, it requires all-to-all communication via the MCC, which is achieved by several ground stations scattered across the Earth. It consists of a massive amount of data that has to be transmitted, in real-time, between a central node and a very large number of systems, across long distances, which requires complex communication protocols and introduces significant delays. Second, the amount of data that has to be processed in real-time by the MCC requires serious processing power. Third, as mentioned previously, to monitor continuously each satellite of the constellation, i.e. to assure continuous TT&C, there have to be several ground stations scattered across the Earth. Nevertheless,

for several reasons, namely geo-political factors, it may not be possible to ensure a direct link with a ground station over some areas. In that case, it would be necessary to retransmit TT&C data through inter-satellite links (ISL) via a path of satellites to an available ground station. Although a large number of gateways, which are installed in ground stations, and ISL are required for providing communication services, continuous TT&C flow of data puts pressure on these links. Fourth, given its critical nature, TT&C data is usually transmitted redundantly, which wastes much needed bandwidth. Fifth, the continuous flow of redundant TT&C data over long distance wireless links also requires significant power consumption. Sixth, given the sensitivity of TT&C data and the fact that it is transmitted over an open wireless link, complex protection protocols are required to maintain the integrity of the data. Seventh, such a configuration is not robust to the failure of the central node, i.e., the MCC. For these reasons, the current TT&C architecture is not suitable for mega-constellations, as it is very challenging to implement in practice, wastes much needed resources, such as power and bandwidth, and, as a result, is very expensive to maintain. As aforementioned, given the paramount importance of cutting costs for the viability of mega-constellations, there has to be a paradigm shift as far as TT&C architecture is concerned so that it is efficient and cost effective. This necessity has already been recently pointed out in [27].

The emergence, over the past decades, of decentralized solutions in various engineering fields as an alternative to the use of these well-known centralized solutions is noteworthy. The management of satellite mega-constellations could also be carried out in this fashion in a decentralized TT&C architecture. In a decentralized configuration, low level constellation operations, such as orbit determination and constellation control, are carried out cooperatively resorting to local communication between satellites via ISL. The gains in efficiency and cost effectiveness of such a paradigm shift are self-evident. First, only local communication between satellites in close proximity is required, which is assured by ISL. Second, by employing cooperative algorithms that run in a distributed configuration across the whole network, all-to-all communication is no longer required, which dramatically reduces the quantity of data that is transmitted. Third, as the constellation management is carried out in a distributed manner, the computational load is shared across multiple satellites. Fourth, the ISL are more secure and require less power, given the proximity of the communication endpoints. For these reasons, the cost effectiveness of LEO mega-constellations would greatly improve with the adoption of a decentralized architecture.

We are witnessing a tipping point. Given the soaring demand, the advantages of such large-scale swarms of satellites constitute a promising business opportunity. Likewise, the advantages and tremendous potential of the applications of very large-scale robotic swarms and very large-scale networks of spatially distributed processes can no longer be ignored, not only as business opportunities but also as a natural change towards efficiency, reliability, and scalability, with profound impact in our society. The LEO mega-constellation venture is pioneering in this aspect and will certainly act as a catalyst for the emergence of a number of similar projects across different fields. In what follows, it is shown that the state-of-the-art tools are not well-suited for control networks of very large-scale, rendering their implementation to real-life applications very challenging or, more often than not, unfeasible. Once again in history, the time has come for automatic control to push the boundaries of technology to enable groundbreaking tasks with far-reaching societal impacts. The goal of this thesis follows from the self-evident

void in the state-of-the-art, aiming to bring these endeavors to fruition.

1.2 State-of-the-art

Decentralized control has been given much attention over the past decade, resulting in a number of very important contributions. The surveys [30–34] offer brief overviews of this extensive topic. In this section, the presentation of the state-of-the-art is divided into three parts. First, the main approaches found in the literature that address the decentralized control problem are detailed. Second, particular emphasis is given to approaches and strategies that are distributed, i.e., that can be synthesized cooperatively in real-time in a distributed manner across the systems of the network resorting to local communication exclusively. Third, state-of-the-art techniques for the control of formations and constellations of satellites are briefly detailed.

1.2.1 Decentralized control

Although plenty of work has been carried out in decentralized control of LTI systems, the problem of synthesizing such controllers, which consists in solving an optimization problem subject to a constraint that arises from the decentralized nature of the configuration, is extremely difficult [22] and remains an open problem. In fact, the optimal solution for a linear system may be nonlinear [35]. Furthermore, it has been shown that the solution of a decentralized design control problem is the result of a convex optimization problem if and only if quadratic invariance of the controller set is ensured [36, 37]. For these reasons, the overwhelming majority of the approaches found in the literature attempt to find the optimal linear solution, which is also a difficult nonconvex optimization problem that remains unsolved. The research on decentralized control of LTV systems, which is naturally more challenging, has been undergone to a much lesser extent. Even though a considerable fraction of real-life systems can be modeled as LTI, there is a multitude of engineering problems that either require an LTV model [38, 39] or can be approximated by an LTV system employing linearization techniques.

One of the proposed approaches for the design of a decentralized controller for an arbitrary network of interconnected LTI systems is to design an $\mathcal{H}_2$-optimal control policy, which amounts to solving a bilinear matrix inequality [40–42]. Although there are well-known algorithms to solve these problems, the associated computational load renders this solution unfeasible for large-scale systems. This approach is extended for LTV systems in [43] and for time-varying network topologies in [44].

Another promising approach to design a control law for an arbitrary decentralized configuration is to relax the underlying optimization problem so that it becomes convex, allowing for the use of well-known optimization techniques. Albeit optimal for the modified problem, the relaxed solution is only an approximation to the solution of the original problem, thus careful relaxation is necessary to ensure that the separation between both solutions is minimal. This approach is designated convex relaxation [45] and it has been successfully employed in control theory [46, 47]. However, such results seldom have stability or boundedness guarantees for the closed-loop system. In [48] this technique is employed to

devise two algorithms for the decentralized controller synthesis problems of LTI systems, one of which has a closed-form solution and, thus, is computationally efficient.

For systems which verify the aforementioned quadratic invariance condition, the decentralized formulation of the control problem is convex and thus tractable, so they can be solved resorting to well-known optimization techniques [49, 50]. Finally, powerful results can be found for even more specific classes of systems, such as positive systems [51] and systems that rely on symmetric Hurwitz state matrices [52]. These approaches are very interesting from a theoretical standpoint, providing valuable insight into the intricacies of the decentralized control problem. Nevertheless, the limiting assumptions on the control networked system, imposed to achieve tractability, are rarely encountered in real-life applications.

Another approach found in the literature is to decouple the network of agents into clusters of agents and consider the interactions between distinct clusters as disturbances. Applying standard techniques to each of the clusters allows to obtain a control law that can be implemented in a decentralized configuration and that can be computed in parallel cluster-wise. However, each local minimization iteration does not take into account its effect on the performance of the remaining clusters, thus this solution is sub-optimal. Moreover, unless the inter-cluster couplings are very weak, this approach leads to significant performance losses. In [53], algorithms for optimized static and dynamic allocation of agents across clusters are detailed, as an attempt to improve the approximation. It is of the utmost importance to make the following distinction clear. These are decoupled strategies, in the sense that the interactions between clusters are not considered at the synthesis level. On the contrary, all other aforementioned approaches take into account couplings between agents at the synthesis level to find a decentralized controller. Note that these algorithms can be implemented distributively among clusters, which is one of the aforementioned technical challenges that must be tackled for the application in very large networks of systems. However, this property is achieved at the expense of performance, since couplings between clusters are neglected. Furthermore, the computations are carried out in a centralized manner in each cluster, requiring i) all-to-all intra-cluster communication; and ii) an agent in each cluster that is in charge of all the computations of the cluster. This approach was applied, for instance, for the control of large-scale urban road networks, for example in [54], and for large-scale smart grids, for example in [55], given that its implementation is feasible for very large-scale.

1.2.2 Distributed control

Farhood et al. [43] reduce the finite-horizon regulator problem of a network of interconnected LTV systems into a sequence of linear matrix inequalities (LMIs). However, even though the synthesized controller can be implemented in a decentralized framework with a topology inherited from the plant, the synthesis of the controller must be performed in real-time in a single computational unit. Since the computational and memory requirements of this solution cannot be distributed across the systems in the network, the computational, memory, and communication burden render such solution unfeasible in practice for very large-scale networks.

This challenge has been noted and addressed to a great extent for the cooperative localization prob-

lem employing the extended Kalman filter (EKF) over very large-scale networks of mobile robots. In this context, partially distributed solutions have been proposed relying on centralized-equivalent frameworks [56], bookkeeping [57], and covariance intersection and split covariance intersection methods [58, 59]. Nevertheless, for each computational unit associated with a system, the communication, memory, and computational requirements, respectively, of these solutions scale with the dimension of the network. Thus, these are not suited to the envisioned very large-scale applications. A promising step towards efficient distributed solutions has been made by [60, 61]. They propose a decentralized method that relies on an approximation of the covariance between the estimation error of each pair of systems, for a general network, which can be computed distributively and supports asynchronous communication and measurements. Although the computational and communication burden of each system does not scale with the dimension of the network, the memory requirements of each computational unit scale linearly with the number of systems in the whole network.

Unlike the estimation problem, research into distributed control schemes even for decoupled LTV systems with a common control objective is rather limited and focuses mainly on particular control problems. For instance, in [62] a solution is presented for the particular case of a formation of unmanned aerial vehicles. These problems are most of the time tackled in a receding horizon control (RHC) scheme, also designated as model predictive control by some authors. For a comprehensive overview of the approaches to distributed RHC, see [63]. Nevertheless, some results for decoupled nonlinear systems have already matured. Although these are designed in a distributed scheme, because of the nonlinear dynamics they, generally, rely on local communication followed by the numerical solution of local optimization problems in real-time. A distributed RHC solution in continuous-time with stability guarantees is proposed in [64]. A decentralized RHC scheme suitable for leader-follower topologies is presented in [65, 66]. Another very interesting distributed and decentralized approach to the RHC problem over networks of decoupled nonlinear systems is proposed in [67]. Therein, unlike in this work, a priori knowledge of the overall system equilibrium is assumed. Their approach is to divide the global optimization problem in several smaller problems that concern each system and its neighborhood. At each time instant, each system solves a local RHC problem to find optimal inputs for itself and the systems in its neighborhood. Then, in each system, the optimal input of the first instant of the finite window concerned with that system is used. Note that, in this framework, the optimal input that a system i predicts for another system j in its neighborhood is, generally, different from the optimal input that system j computes for itself. Sufficient stability conditions are derived as a function of this mismatch between optimal solutions. However, this work does not provide a bound on the mismatch between solutions as a function of the network topology and dynamics. Moreover, in this framework, to compute the control input at each time instant, each system ought to receive the state of every neighbor and only then proceed with the solution of the local receding horizon optimization problem. Thus, it is challenging to implement it in practice without introducing significant delays.

1.2.3 Formations and constellations of satellites

Orbit determination and control has already been given attention for small formations and constellations of satellites, for example in [56, 68–72] for cooperative navigation and in [73–76] for cooperative orbit control. However, these methods do not meet the aforementioned strict computational, memory, and communication requirements that enable the application to very large-scale constellations of satellites. One of the few works to address the decentralized constellation navigation problem is [56], at the expense of unbearable computational load, since a centralized-equivalent approach is proposed. In [77] a stable decentralized control solution is proposed leveraging local communication exclusively for the particular case of a constellation of satellites over a single orbital plane.

A common approach to the cooperative orbit control problem is the bounding-box method, employed, for instance, in [73] and [76]. In this scheme, a reference position is generated for each satellite around which an error box is defined. Whenever each satellite is inside the error box no control input is used, but when it leaves said box the feedback control is enabled to drive it inside of the error box. The main advantage of this scheme is that the low-level control feedback loop of each satellite is decoupled from the others. However, this decoupled scheme tries to correct common secular and periodic perturbations that cause the satellites to drift in relation to the nominal constellation but that perturb the constellation shape to a much lesser extent, thus wasting much-valuable fuel. If, to mitigate this effect, the nominal positions are updated in real-time, then the global computation of consistent nominal positions for each satellite has to be carried out in a centralized node or cooperatively across the network in real-time. Nevertheless, this alternative requires tremendous communication load, which is unfeasible for large-scale networks.

1.3 Goals

Comparing the state-of-the-art with the technical challenges that emerge with the envisioned very large-scale applications, one of which is the LEO mega-constellation venture, it is clear that they are not addressed adequately. The goal of this thesis follows from this, with the aim of enabling ground-breaking applications with profound societal impact that rely on very large-scale systems.

This work addresses the problem of the design of decentralized and distributed control solutions to very large-scale networks of systems with a common control objective. The problem is formulated in a RHC framework considering: i) coupled LTV dynamics for each system and ii) coupled LTV tracking outputs expressed in a generic, possibly time-varying, topology to model the network-wise control objective. Note that, as aforementioned, a control solution for a network of systems with coupled nonlinear dynamics and coupled nonlinear tracking outputs can be obtained from this formulation making use of linearization techniques.

More specifically, the goal of this thesis is threefold. First, the objective is to formulate the decentralized control problem and formal distributed implementation feasibility constraints for very large-scale systems. Second, the goal is to devise a local linear state feedback solution to the RHC problem subject

to the communication, memory, and computational constraints which are critical for a feasible implementation to very large-scale systems. Third, the objective is to apply the developed distributed decentralized RHC algorithm to the pressing cooperative orbit control problem of mega-constellations of satellites in LEO to assess the performance and scalability of the proposed solution.

1.4 Solution overview

First, the decentralized distributed control problem is formulated as a RHC regulator problem. The decentralized framework employed in this work is such that each system is associated with a computational unit that computes its own control input making use of local communication and local state feedback exclusively. These local communication restrictions are formulated as a sparsity constraint. The communication, memory, and computational requirements are formalized in four constraints, three of which establish bounds on the asymptotic growth of these resources as the dimension of the network increases.

Second, the bulk of the theoretical contribution of this thesis lies on the development of a distributed decentralized RHC solution. This objective is tackled in two stages, following a divide-and-conquer approach. In a first instance, the necessary conditions of the decentralized optimization problem disregarding the communication, memory, and computational feasibility constraints are analyzed to devise a convex relaxation approach, designated by one-step relaxation. This approach is taken to reduce the separation to the optimal solution of the RHC regulator problem, but an intuitive analysis of the intricacies of the relaxation procedure is also conducted resorting to an equivalent formulation of the relaxation. Moreover, a decentralized tracking method is developed building on the regulator solution. The devised convex relaxation approach is also validated resorting to numerical simulations and experimental results. In a second instance, the communication, memory, and computational feasibility constraints are imposed building on the convex relaxation controller synthesis, for the particular case of decoupled dynamics. This is achieved as the result of an approximation that is introduced, which allows to decouple the gain synthesis procedure. This approximation is formally detailed, its origin and logic are explored, and its role on the decoupling of the contributions of each system to the global tracking cost is made clear.

Third, the set of relative orbital elements introduced in [78] is leveraged to formulate the cooperative orbit control problem in a novel cooperative framework. Instead of considering independent reference orbits for each satellite that makes up the constellation, a framework based on a coupled tracking objective is proposed to maintain constellation rigidity with emphasis on communication requirements and fuel saving. High-fidelity simulations are carried out to assess the scalability and performance of the proposed solution.

1.5 Contributions

The goals of this thesis were envisioned to push the boundaries of automatic control to bring pressing very large-scale tasks to fruition. The technical challenges that are hampering the transition from their conceptualization to deployment were identified. From the comparison of the state-of-the art solutions with these emerging technical challenges, there is a self-evident gap that this work set out to address. The main contributions of this thesis are:

- Formulation of the decentralized RHC problem and the distributed implementation feasibility constraints for very large-scale systems;
- Derivation of a well-performing convex relaxation procedure;
- Development of decentralized regulator and tracker methods based on efficient global closed-form solutions;
- Development of a distributed and decentralized RHC solution for the particular case of dynamically decoupled systems;
- Development of a well-performing distributed decentralized solution to the pressing on-board orbit control problem of satellites in a LEO mega-constellation.

This work has already been partially published in international peer-reviewed journals [79, 80] and it is also under peer-review in an international journal [81]. In particular, Sections 3.1, 3.2, and 3.3 are published in [79] with significant modifications; Section 3.4 is part of the work published in [80]; and Chapters 4 and 5 are under peer-review in [81].

Furthermore, emphasis is put on transparency and reproducibility of the simulation results presented in this thesis. Indeed, the implementation of the methods put forward in this work and the source-code of all simulation examples are available in well documented open-source repositories:

- DECENTER toolbox (`https://decenter2021.github.io`);
- Quadruple-Tank Setup (`https://github.com/decenter2021/quadruple-tank-setup`);
- osculating2mean package (`https://github.com/decenter2021/osculating2mean`);
- tudat-matlab-thrust-feedback package (`https://github.com/decenter2021/tudat-matlab-thrust-feedback`).

1.6 Thesis outline

This thesis is organized as follows. In Chapter 2, the decentralized RHC problem is formulated alongside the communication, computational, and memory constraints that the control solution must follow to allow for its implementation for very large-scale systems in real-time. In Chapter 3, the one-step relaxation is derived and validated. In Chapter 4, the proposed distributed and decentralized RHC

algorithm is derived. In Chapter 5, the distributed and decentralized RHC algorithm put forward in this thesis is applied to the orbit control problem of LEO mega-constellations. Chapter 6 presents the main conclusions of this work.

Chapter 2

Problem Statement and Notation

2.1 Notation

Throughout this work, $\mathrm{sgn}(x)$ and $|x|$ denote the sign and absolute value, respectively, of a real number x. The identity, null, and ones matrices, all of proper dimensions, are denoted by $\mathbf{I}$, $\mathbf{0}$, and $\mathbf{1}$, respectively. Alternatively, $\mathbf{I}_n$, $\mathbf{0}_{n\times m}$, and $\mathbf{1}_{n\times m}$ are also used to represent the $n \times n$ identity matrix and the $n \times m$ null and ones matrices, respectively. The entry (i, j) of a matrix $\mathbf{A}$ is denoted by $[\mathbf{A}]_{ij}$. The i-th component of a vector $\mathbf{v} \in \mathbb{R}^n$ is denoted by $[\mathbf{v}]_i$ and $\mathrm{diag}(\mathbf{v})$ denotes the $n \times n$ square diagonal matrix whose diagonal is $\mathbf{v}$. The column-wise concatenation of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$ is denoted by $\mathrm{col}(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ and $\mathrm{diag}(\mathbf{A}_1, ..., \mathbf{A}_N)$ denotes the block diagonal matrix whose diagonal blocks are given by matrices $\mathbf{A}_1, ..., \mathbf{A}_N$. The vectorization of a matrix $\mathbf{A}$, denoted herein by $\mathrm{vec}(\mathbf{A})$, returns a vector composed of the concatenated columns of $\mathbf{A}$. The Kronecker delta is denoted by $\boldsymbol{\delta}_{ij}$. Given a symmetric matrix $\mathbf{P}$, $\mathbf{P} \succ \mathbf{0}$ and $\mathbf{P} \succeq \mathbf{0}$ are used to point out that $\mathbf{P}$ is positive definite and positive semidefinite, respectively. The Kronecker product of two matrices $\mathbf{A}$ and $\mathbf{B}$ is denoted by $\mathbf{A} \otimes \mathbf{B}$. The cardinality of a set $\mathcal{A}$ is denoted by $|\mathcal{A}|$. The Cartesian product of two sets $\mathcal{A}$ and $\mathcal{B}$ is denoted by $\mathcal{A} \times \mathcal{B}$. The modulo operation is denoted by $a \bmod b$, which returns the remainder of the integer division of $a \in \mathbb{N}$ by $b \in \mathbb{N}$. The greatest integer less than or equal to $x \in \mathbb{R}$ is denoted by $\lfloor x \rfloor$.

2.2 Problem Statement

The problem statement is introduced in two steps. First, in Section 2.2.1, the models of the systems of the network are defined, which are posteriorly grouped to define a global model for the network. Second, in Section 2.2.2, the structure of the local RHC controllers is detailed and the control problem is formulated for the global controller. Third, in Section 2.2.3, the communication, computational, and memory constraints are defined. It is important to point out that this problem is stated and addressed for a generic network of systems with coupled LTV dynamics and coupled LTV tracking outputs that express a network-wise control objective. No further assumptions, neither on the dynamics of each system, nor on the tracking output function, nor on the topology of couplings are made.

2.2.1 Model of the network

Consider a network of N systems, $\mathcal{S}_i$ with $i = 1, \ldots, N$, each associated with one computational unit, $\mathcal{T}_i$. Each system is modeled by LTV dynamics, which are coupled with a set of other systems. Each system has also an LTV tracking output, which is coupled with another set of systems. The tracking outputs can express a control objective that is common to all the systems (for an example, see Chapter 5). The dynamic couplings can be represented by a directed graph $\mathcal{G}_d := (\mathcal{V}_d, \mathcal{E}_d)$, which may be time-varying, composed of a set $\mathcal{V}_d$ of vertices and a set $\mathcal{E}_d$ of directed edges. An edge e incident on vertices i and j, directed from j towards i, is denoted by $e = (j, i)$. For a vertex i, its in-degree, ${}^{d}\nu_i^-$, is the number of edges directed towards it, and its in-neighborhood, ${}^{d}\mathcal{D}_i^-$, is the set of indices of the vertices from which such edges originate. Conversely, for a vertex i, its out-degree, ${}^{d}\nu_i^+$, is the number of edges directed from it, and its out-neighborhood, ${}^{d}\mathcal{D}_i^+$, is the set of indices of the vertices towards which such edges are directed. For a more detailed overview of the elements of graph theory used to model this network, see [82] and [83]. In this framework, each system is represented by a vertex, i.e., system $\mathcal{S}_i$ is represented by vertex $i \in \mathcal{V}_d$, and, if the dynamics of $\mathcal{S}_i$ depend on the state of $\mathcal{S}_j$, then this coupling is represented by an edge directed from vertex j towards vertex i in $\mathcal{G}_d$, i.e., edge $e = (j, i) \in \mathcal{E}_d$. It is important to stress that the direction of the edge matters. Note, for instance, that the fact that the dynamics of $\mathcal{S}_i$ depend on the state of $\mathcal{S}_j$ does not, necessarily, imply the converse. Analogously, the tracking output couplings can be represented by a directed graph $\mathcal{G}_o$. It is assumed henceforth, without loss of generality, that $i \in {}^{d}\mathcal{D}_i^-$. Moreover, given that the local goal of each system $\mathcal{S}_i$ is to drive the tracking output to zero, it is also assumed that the tracking output of system $\mathcal{S}_i$ depends on its state, i.e. $i \in {}^{o}\mathcal{D}_i^-$. Thus, each vertex of both topology graphs is assumed to have a self-loop.

The dynamics of system $\mathcal{S}_i$ are modeled by the discrete-time LTV system

$$\begin{cases} \mathbf{x}_i(k+1) = \sum_{j \in {}^{d}\mathcal{D}_i^-} \mathbf{A}_{i,j}(k)\mathbf{x}_j(k) + \sum_{j \in {}^{d}\mathcal{D}_i^-} \mathbf{B}_{i,j}(k)\mathbf{u}_j(k) \\ \mathbf{z}_i(k) = \sum_{j \in {}^{o}\mathcal{D}_i^-} \mathbf{H}_{i,j}(k)\mathbf{x}_j(k), \end{cases} \tag{2.1}$$

where $\mathbf{x}_i(k) \in \mathbb{R}^{n_i}$ is the state vector, $\mathbf{u}_i(k) \in \mathbb{R}^{m_i}$ is the input vector, and $\mathbf{z}_i(t) \in \mathbb{R}^{o_i}$ is the tracking output vector, all of system $\mathcal{S}_i$; matrices $\mathbf{A}_{i,j}(k)$ with $j \in {}^{d}\mathcal{D}_i^-$, $\mathbf{B}_{i,j}(k)$ with $j \in {}^{d}\mathcal{D}_i^-$, and $\mathbf{H}_{i,j}(k)$ with $j \in {}^{o}\mathcal{D}_i^-$ are time-varying matrices that model the dynamics of system $\mathcal{S}_i$ and its tracking output couplings with the other systems in its in-neighborhood. Note that linearization techniques can be employed to approximate the dynamics of a nonlinear system with a nonlinear tracking output as an LTV system of the form of (2.1).

The global dynamics of the network can, then, be modeled by the discrete-time LTV system

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) \\ \mathbf{z}(k) = \mathbf{H}(k)\mathbf{x}(k), \end{cases} \tag{2.2}$$

where $\mathbf{x}(k) := \mathrm{col}(\mathbf{x}_1(k), \ldots, \mathbf{x}_N(k)) \in \mathbb{R}^n$ is the global state vector; $\mathbf{u}(k) := \mathrm{col}(\mathbf{u}_1(k), \ldots, \mathbf{u}_N(k)) \in \mathbb{R}^m$ is the global input vector; $\mathbf{z}(k) := \mathrm{col}(\mathbf{z}_1(k), \ldots, \mathbf{z}_N(k)) \in \mathbb{R}^o$ is the global tracking output vector; $\mathbf{A}(k)$ is

a block matrix whose block of indices (i, j) is $\mathbf{A}_{i,j}(k)$, if $j \in {}^{d}\mathcal{D}_i^-$, and $\mathbf{0}_{n_i \times n_j}$ otherwise; $\mathbf{B}(k)$ is a block matrix whose block of indices (i, j) is $\mathbf{B}_{i,j}(k)$, if $j \in {}^{d}\mathcal{D}_i^-$, and $\mathbf{0}_{n_i \times m_j}$ otherwise; and $\mathbf{H}(k)$ is a block matrix whose block of indices (i, j) is $\mathbf{H}_{i,j}(k)$, if $j \in {}^{o}\mathcal{D}_i^-$, and $\mathbf{0}_{o_i \times n_j}$ otherwise.

Before proceeding with the problem statement, it is worth pointing out that the network-wise control objective of virtually all large-scale networks can be expressed by sparse tracking couplings. In particular, ${}^{o}\nu_i^-$, the number of tracking output couplings of a system $\mathcal{S}_i$, is bounded and independent of N, the number of systems in the network. The novel distributed solution presented in Chapter 5 takes advantage of the sparsity of these couplings to allow a distributed and decentralized RHC algorithm under communication, computational, and memory limitations.

2.2.2 Decentralized receding horizon problem

The goal of the proposed controller is to regulate the global tracking output making use of local linear state feedback. In a centralized configuration, each system has access to the global state of the network, i.e., to the state of every other system, at the expense of all-to-all communication via a central node. In a decentralized configuration, that is not the case: each system $\mathcal{S}_i$ only has access to the state of a subset of systems. As before, the communication topology can be represented by a directed graph $\mathcal{G}_c$. If $\mathcal{S}_i$ has access to the state of $\mathcal{S}_j$ via directed communication from $\mathcal{S}_j$ to $\mathcal{S}_i$, then it is represented by an edge directed from vertex j towards vertex i in $\mathcal{G}_c$. Given that each system is naturally able to communicate with itself, it is assumed that $i \in {}^{c}\mathcal{D}_i^-$. Thus, at each discrete time instant k, only the states of the systems in the communication in-neighborhood of $\mathcal{S}_i$ are known to $\mathcal{T}_i$. Figure 2.1 depicts a scheme of the dynamic, output, and communication topologies. The control input of $\mathcal{S}_i$ is, thus, of the form

$$\mathbf{u}_i(k) = -\sum_{j \in {}^{c}\mathcal{D}_i^-} \mathbf{K}_{i,j}(k)\mathbf{x}_j(k), \tag{2.3}$$

where $\mathbf{K}_{i,j}(k)$ for $j \in {}^{c}\mathcal{D}_i^-$ are the controller gains of $\mathcal{S}_i$.

The goal is to design optimal controller gains according to a performance criteria. Note that, due to the dynamics and tracking couplings between systems, the gains of each system cannot be designed independently. For that reason, the local controllers are concatenated to define a global controller, which is used to formulate a global problem. The global control input is given by

$$\mathbf{u}(k) = -\mathbf{K}(k)\mathbf{x}(k), \tag{2.4}$$

where $\mathbf{K}(k)$ is the global gain matrix. Note that the global control law (2.4) is equivalent to the concatenation of the local control laws (2.3) if and only if $\mathbf{K}(k)$ follows the sparsity pattern of block matrix $\mathbf{E}_{\mathcal{G}_c}$, whose block of indices (i, j) is given by

$$\mathbf{E}_{\mathcal{G}_c\ i,j} = \begin{cases} \mathbf{1}_{m_i \times n_j}, & j \in {}^{c}\mathcal{D}_i^- \\ \mathbf{0}_{m_i \times n_j}, & j \notin {}^{c}\mathcal{D}_i^-. \end{cases}$$

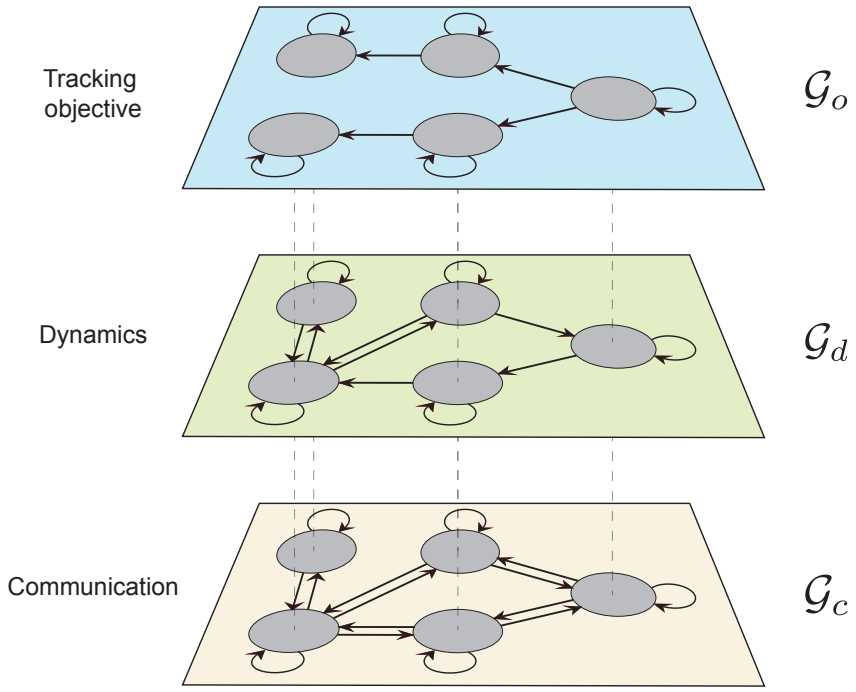


Figure 2.1: Scheme of the dynamics, tracking output, and communication topologies.

This sparsity condition is denoted as $\mathbf{K}(k) \in \mathrm{Sparse}(\mathbf{E}_{\mathcal{G}_c})$, with

$$\mathrm{Sparse}(\mathbf{E}) := \left\{ [\mathbf{K}]_{ij} \in \mathbb{R}^{m \times n} : [\mathbf{E}]_{ij} = 0 \implies [\mathbf{K}]_{ij} = 0;\ i = 1, ..., m,\ j = 1, ..., n \right\}.$$

If all-to-all communication were possible, then $\mathbf{E}_{\mathcal{G}_c} = \mathbf{1}$, corresponding to a centralized configuration. Note that, in a centralized framework, if the pair $(\mathbf{A}(k), \mathbf{B}(k))$ is uniformly completely controllable, then it is possible to design a globally exponential stable controller. However, in a decentralized framework, that is only a necessary condition.

The goal is to minimize the infinite-horizon performance cost

$$J_\infty := \sum_{i=1}^{N} J_{i\infty} = \sum_{i=1}^{N} \sum_{\tau=0}^{\infty} \left(\mathbf{z}_i^T(\tau) \mathbf{Q}_i(\tau) \mathbf{z}_i(\tau) + \mathbf{u}_i^T(\tau) \mathbf{R}_i(\tau) \mathbf{u}_i(\tau) \right),$$

where $\mathbf{Q}_i(\tau) \succeq 0$ and $\mathbf{R}_i(\tau) \succ 0$ are known time-varying matrices of appropriate dimensions that weigh the local tracking output and input of each system $\mathcal{S}_i$, respectively. The proposed method consists of an approximation to the solution of the infinite-horizon problem above, considering multiple finite-horizon problems with an associated cost of the form

$$J(k) := \sum_{i=1}^{N} \left(\mathbf{z}_i^T(k+H) \mathbf{Q}_i(k+H) \mathbf{z}_i(k+H) + \sum_{\tau=k}^{k+H-1} \left(\mathbf{z}_i^T(\tau) \mathbf{Q}_i(\tau) \mathbf{z}_i(\tau) + \mathbf{u}_i^T(\tau) \mathbf{R}_i(\tau) \mathbf{u}_i(\tau) \right) \right), \quad (2.5)$$

where $H \in \mathbb{N}$ denotes the length of the finite window. The extension of this problem to an infinite-horizon is achieved by making use of the RHC scheme. At each discrete time instant k, one considers a finite window $\{k, ..., k+H\}$, with H large enough so that the gains computed within that window converge

to those that would be obtained if an arbitrarily large window was used. Then, the gains that minimize $J(k)$ are computed for the appropriate window and only the first is actually used to compute the control action for that time instant, discarding the remaining gains. At the next time instant, $k+1$, a new finite window is considered and a new sequence of gains is computed to minimize $J(k+1)$, and so forth. To reduce the computational load, $d \in \mathbb{N}$ gains may be used, instead of just one, defining a new window and computing the gains associated with it every d time steps. Although the higher d is, the less the computational load is, if too large a value of d is chosen, a degradation of performance and robustness may occur.

To formulate the problem globally, (2.5) can be rewritten as

$$J(k) = \mathbf{z}^T(k+H)\mathbf{Q}(k+H)\mathbf{z}(k+H) + \sum_{\tau=k}^{k+H-1} \left(\mathbf{z}^T(\tau)\mathbf{Q}(\tau)\mathbf{z}(\tau) + \mathbf{u}^T(\tau)\mathbf{R}(\tau)\mathbf{u}(\tau)\right),$$

where $\mathbf{Q}(\tau) := \operatorname{diag}\left(\mathbf{Q}_1(\tau), \ldots, \mathbf{Q}_N(\tau)\right)$ and $\mathbf{R}(\tau) := \operatorname{diag}\left(\mathbf{R}_1(\tau), \ldots, \mathbf{R}_N(\tau)\right)$. Note that designing a decentralized controller for a network of systems, whose local dynamics are described by the LTV system (2.1), is equivalent to designing a controller (2.4) for the global network (2.2), whose gain must follow a sparsity pattern. One aims to optimally compute a sequence of gains that follow the sparsity pattern required for a fully decentralized configuration. For a finite-horizon, solve the optimization problem

$$\begin{aligned}
\underset{\substack{\mathbf{K}(\tau)\in\mathbb{R}^{m\times n}\\ \tau\in\{k,\ldots,k+H-1\}}}{\text{minimize}} \quad & J(k) \\
\text{subject to} \quad & \mathbf{K}(\tau) \in \operatorname{Sparse}(\mathbf{E}_{\mathcal{G}_c}),\ \tau = k, \ldots, k+H-1 \\
& \mathbf{u}(\tau) = -\mathbf{K}(\tau)\mathbf{x}(\tau),\ \tau = k, \ldots, k+H-1 \\
& \mathbf{x}(\tau+1) = \mathbf{A}(\tau)\mathbf{x}(\tau) + \mathbf{B}(\tau)\mathbf{u}(\tau),\ \tau = k, \ldots, k+H-1.
\end{aligned} \tag{2.6}$$

Even for systems with LTI dynamics, the optimization problem (2.6) is nonconvex and its optimal solution is still an open problem. In Chapter 3, (2.6) is relaxed so that it becomes convex, allowing for an approximate solution to the original problem. Figure 2.2 depicts a flowchart of the RHC scheme.

2.2.3 Implementation feasibility constraints

It is of the utmost importance to remark that the solution devised for (2.6) must be feasible to implement in real-time in a decentralized configuration. In particular, the procedure to compute each gain $\mathbf{K}_{i,j}(k)$, with $j \in {}^{c}\mathcal{D}_i^-$, in $\mathcal{T}_i$ must follow several constraints regarding communication, computational, and memory requirements. In this section, these constraints are presented with detail and a set of requirements for the distributed control solution is defined.

First, there are various communication requirements associated with the computation of the controller gains. A very important aspect to take into account is the synchronization of the data transmissions. On one hand, a variable stored in $\mathcal{T}_j$ at time instant k that is required to perform a computation in $\mathcal{T}_i$ at time instant k would have to be transmitted instantaneously. These are denoted as hard real-time transmissions, for which very complex synchronization algorithms are required. On the other hand, in a soft

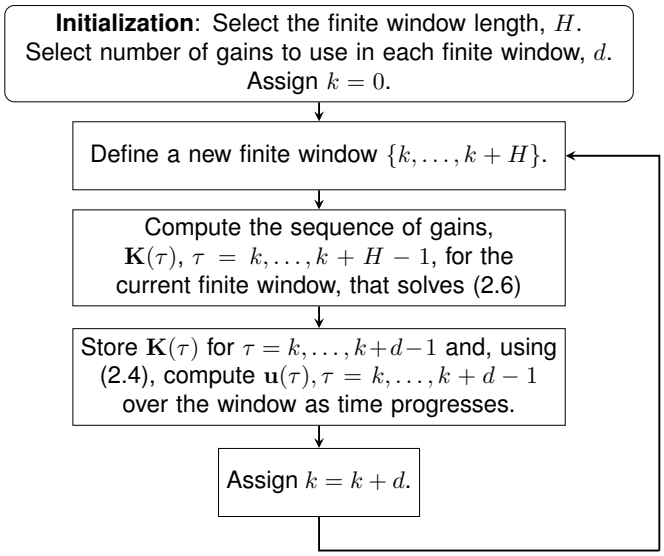


Figure 2.2: Flowchart of the RHC scheme for the infinite-horizon problem.

real-time transmission, the receiving computational unit only makes use of the transmitted data since, at least, the discrete time instant that follows the instant of the transmission. One can readily point out that the definition of the local control input (2.3) requires a hard real-time transmission. In fact, $\mathbf{x}_j(k)$, with $j \in {}^c\mathcal{D}_i^-$, has to be instantaneously transmitted to $\mathcal{T}_i$, because it is known to $\mathcal{T}_j$ only at time instant k and it is required in $\mathcal{T}_i$ at time instant k. For this case in particular, various techniques can be used to allow for a feasible implementation, since the state of a system is a small data transfer and it can be easily predicted over small time intervals. For that reason, in this thesis, the communication requirements are focused on the gain computation, for which hard real-time transmissions are not allowed. Second, the number of available communication links between systems is limited. Although there are decentralized solutions in the literature that require all-to-all communication (see Section 1.2), these are not scalable to very large-scale networks. Thus, to seek a scalable solution, the number of communications links established with each system must not increase with an increase of the number of systems in the network, i.e., the communication complexity of each system ought to be $\mathcal{O}(1)$ with the dimension of the network, N. It is very important to point out that the aforementioned communication link limitations are addressed at the protocol level, i.e., the restrictions are applied to the exchange of data between systems, not on the physical communication links. In particular, $\mathcal{T}_i$ is not allowed to access data from $\mathcal{T}_k$ via a path of systems through which the information could be retransmitted. Such a configuration would, for large-scale networks, increase communication delays, decrease the robustness of the architecture, result in an uneven distribution of the communication burden, and increase the complexity of the communication protocol. Third, the memory of each computational unit is limited. For that reason, the amount of data to be stored in each one must not scale with the dimension of the network. That is, the data storage complexity of each computational unit ought to grow with $\mathcal{O}(1)$ with the dimension of the network, N. Fourth, the computational resources available to each computational unit, which are required to implement the local control solution, are limited. The computational load of the global control algorithm must be distributed across all computational units in such a way that each carries out computations concerning their

system exclusively, which circumvents the curse of dimensionality. For that reason, the computational complexity of the floating-point operations carried out by each computational unit must grow with $\mathcal{O}(1)$ with the dimension of the network, N.

To sum-up, the control solution must satisfy the following constraints.

Constraint 1. Hard real-time transmissions are not allowed for the synthesis of controller gains.

Constraint 2. The communication complexity of each system ought to grow with $\mathcal{O}(1)$ with N.

Constraint 3. The data storage complexity of each computational unit ought to grow with $\mathcal{O}(1)$ with N.

Constraint 4. The computational complexity of each computational unit ought grow with $\mathcal{O}(1)$ with N.

The goal is to design a decentralized control solution that solves the optimization problem (2.6) subject to the communication, memory, and computational Constraints 1–4, which are critical for a feasible implementation to very large-scale systems. The decentralized gain synthesis procedure proposed in Chapter 4, which is based on a convex relaxation of (2.6), does not satisfy these constraints. In Chapter 4, that solution is leveraged to devise a distributed synthesis procedure that abides by these feasibility constraints.

Chapter 3

Decentralized linear quadratic control

In this thesis, a divide-and-conquer approach is followed. In this chapter, the decentralized RHC problem stated in Section 2.2.2 is addressed disregarding the computational, memory, and communication constraints put forward in Section 2.2.3. These results are then leveraged, in Chapter 4, to devise a distributed synthesis procedure that abides by these feasibility constraints for the particular case of dynamically decoupled systems. In this chapter, for simplicity, we consider time-invariant topologies, but the results in Chapter 4 are extended to consider time-varying topologies. This chapter is organized as follows. In Section 3.1, a convex relaxation to the RHC tracking output regulation problem is devised. In Section 3.2, a tracking solution is proposed building on the regulator results. Finally, in Sections 3.3 and 3.4, the proposed control solutions are validated resorting to numeric simulations and experimental results, respectively.

3.1 One-step convex relaxation

As aforementioned, the optimization problem (2.6) is nonconvex. Thus, to use standard optimization techniques, convex relaxation is performed. Albeit optimal for the modified problem, the relaxed solution is only an approximation to the solution of the original problem. For this reason, careful relaxation is necessary to ensure that the separation between both solutions is minimal. In this work, the proposed convex relaxation procedure, designated one-step relaxation, is derived from an analysis of the necessary conditions of a constrained minimum of (2.6). Recall that, in this chapter, the communication, memory, and computational Constraints 1–4 put forward in Section 2.2.3 are not taken into account. Despite that, in what follows, it will become clear that the results of this chapter can be leveraged to devise a distributed solution that abides by those constraints. Consider the following preliminary result.

Theorem 3.1. *The solutions to the necessary condition for a constrained minimum of* (2.6) *follow*

$$\begin{cases} \left[\left(\mathbf{S}(\tau)\mathbf{K}(\tau) - \mathbf{B}^T(\tau)\mathbf{P}(\tau+1)\mathbf{A}(\tau)\right)\mathbf{x}(\tau)\mathbf{x}^T(\tau)\right]_{ji} = 0 & , [\mathbf{E}_{\mathcal{G}_c}]_{ji} \neq 0 \\ [\mathbf{K}(\tau)]_{ji} = 0 & , [\mathbf{E}_{\mathcal{G}_c}]_{ji} = 0, \end{cases} \tag{3.1}$$

for $\tau = k, \ldots, k+H-1$*, where* $\mathbf{P}(\tau)$ *is a symmetric positive semidefinite matrix given by*

$$\begin{cases} \mathbf{P}(k+H) = \mathbf{H}^T(k+H)\mathbf{Q}(k+H)\mathbf{H}(k+H) \\ \mathbf{P}(\tau) = \mathbf{H}(\tau)^T\mathbf{Q}(\tau)\mathbf{H}(\tau) + \mathbf{K}^T(\tau)\mathbf{R}(\tau)\mathbf{K}(\tau) + (\mathbf{A}(\tau) - \mathbf{B}(\tau)\mathbf{K}(\tau))^T\mathbf{P}(\tau+1)(\mathbf{A}(\tau) - \mathbf{B}(\tau)\mathbf{K}(\tau)), \end{cases} \tag{3.2}$$

for $\tau = k, \ldots, k+H$*, and*

$$\mathbf{S}(\tau) := \mathbf{B}^T(\tau)\mathbf{P}(\tau+1)\mathbf{B}(\tau) + \mathbf{R}(\tau), \tag{3.3}$$

for $\tau = k, \ldots, k+H-1$*. Furthermore, it follows that*

$$V_k(\tau) = \mathbf{x}^T(\tau)\mathbf{P}(\tau)\mathbf{x}(\tau) \tag{3.4}$$

for $\tau = k, \ldots k+H$*, where* $V_k(\tau)$ *is the cost-to-go, which is defined as*

$$V_k(\tau) := \mathbf{z}^T(k+H)\mathbf{Q}(k+H)\mathbf{z}(k+H) + \sum_{s=\tau}^{k+H-1} \left(\mathbf{z}^T(s)\mathbf{Q}(s)\mathbf{z}(s) + \mathbf{u}^T(s)\mathbf{R}(s)\mathbf{u}(s)\right). \tag{3.5}$$

Proof. See Section A.1. □

Remark 3.1. The proof of Theorem 3.1 follows the Lagrange-multiplier approach. Nevertheless, it is possible to obtain the same result following an approach based on dynamic programming, which offers additional insight into the anatomy of the problem. This alternative derivation is presented in Section A.2.

Theorem 3.1 puts emphasis on the nonconvexity of (2.6) since there are, in general, multiple solutions to the necessary condition of a constrained minimum. In particular, it is interesting to notice the presence of the factor $\mathbf{x}(\tau)\mathbf{x}^T(\tau)$, which is of rank 1, in (3.1). To understand its origin and implications consider the following result.

Theorem 3.2. *Let* $\mathbf{x_c}(\tau), \tau = k, \ldots, k+H$*, and* $\mathbf{u_c}(\tau), \tau = k, \ldots, k+H-1$*, denote the states and inputs, respectively, that arise from the centralized solution to* (2.6)*, i.e., the solution with* $\mathbf{E}_{\mathcal{G}_c} = \mathbf{1}$*. If each node in* $\mathcal{G}_c$ *has a self-loop, i.e.* $i \in {}^c\mathcal{D}_i^- \ \forall i \in \mathcal{V}_c$*, and*

$$\sum_{j \in {}^c\mathcal{D}_i^-} |[\mathbf{x_c}(\tau)]_j| \neq 0 \vee [\mathbf{u_c}]_i = 0, \ \forall i \in \{1, \ldots, m\} \ \forall \tau \in \{k, \ldots, k+H-1\}, \tag{3.6}$$

then there exist $\mathbf{K}(\tau) \in \mathrm{Sparse}(\mathbf{E}_{\mathcal{G}_c}), \ \tau = k, \ldots, k+H-1$*, that solve* (2.6) *such that* $\mathbf{x}(\tau) = \mathbf{x_c}(\tau), \forall \tau \in \{k, \ldots, k+H-1\}$.

Proof. See Section A.3. □

At first sight Theorem 3.2 seems very encouraging since, under mild conditions, one can synthesize decentralized controller gains that attain optimal centralized performance. Nevertheless, to compute those gains, the state at the beginning of the window, i.e, $\mathbf{x}(k)$, would have to be known, which is impossible to achieve without all-to-all communication. The intuition behind the proposed one-step relaxation is to achieve a decentralized gain synthesis procedure that does not depend on the initial state of each

finite window. Given that the dependence of (3.1) on $\mathbf{x}(k)$ is expressed by the factor $\mathbf{x}(\tau)\mathbf{x}^T(\tau)$, the convex relaxation approach that is followed is to approximate a single solution to (3.1), not necessarily optimal, which follows

$$\begin{cases} \left[\mathbf{S}(\tau)\mathbf{K}(\tau) - \mathbf{B}^T(\tau)\mathbf{P}(\tau+1)\mathbf{A}(\tau)\right]_{ji} = 0, & [\mathbf{E}_{\mathcal{G}_c}]_{ji} \neq 0 \\ [\mathbf{K}(\tau)]_{ji} = 0, & [\mathbf{E}_{\mathcal{G}_c}]_{ji} = 0. \end{cases} \tag{3.7}$$

Theorem 3.3. *Let $\mathbf{l}_j$ denote a column vector whose entries are all set to zero except for the j-th one, which is set to 1, and $\boldsymbol{\mathcal{L}}_j := \mathrm{diag}(\mathbf{l}_j)$. Define a vector $\mathbf{m}_j \in \mathbb{R}^m$ to encode the non-zero entries in the j-th column of $\mathbf{K}(\tau)$ as*

$$\begin{cases} [\mathbf{m}_j]_i = 0, & [\mathbf{E}_{\mathcal{G}_c}]_{ij} = 0 \\ [\mathbf{m}_j]_i = 1, & [\mathbf{E}_{\mathcal{G}_c}]_{ij} \neq 0 \end{cases}, \; i = 1, ..., m \; ,$$

and let $\boldsymbol{\mathcal{M}}_j := \mathrm{diag}(\mathbf{m}_j)$. Then, the gain of the one-step sub-optimal solution to (2.6) *is given by*

$$\mathbf{K}(\tau) = \sum_{j=1}^{n} \left(\mathbf{I} - \boldsymbol{\mathcal{M}}_j + \boldsymbol{\mathcal{M}}_j \mathbf{S}(\tau) \boldsymbol{\mathcal{M}}_j\right)^{-1} \boldsymbol{\mathcal{M}}_j \mathbf{B}^T(\tau)\mathbf{P}(\tau+1)\mathbf{A}(\tau)\boldsymbol{\mathcal{L}}_j \; , \tag{3.8}$$

$\tau = k, \ldots, k+H-1$, *where $\mathbf{P}(\tau+1)$ is given recurrently by* (3.2).

Proof. See Section A.4. □

Remark 3.2. Notice that (3.8) is similar to the centralized solution to (2.6), given by

$$\mathbf{K}(\tau) = \mathbf{S}(\tau)^{-1}\mathbf{B}^T(\tau)\mathbf{P}(\tau+1)\mathbf{A}(\tau) \; .$$

The fundamental difference between them is that, imposing the sparsity constraint, the entries of the j-th column of $\mathbf{K}(\tau)$ depend on $\boldsymbol{\mathcal{M}}_j\mathbf{S}(\tau)\boldsymbol{\mathcal{M}}_j$, instead of $\mathbf{S}(\tau)$. This reduced form of $\mathbf{S}(\tau)$ corresponds to one of its principal submatrices, in which the i-th row and i-th column are replaced by zeros if $[\mathbf{E}_{\mathcal{G}_c}]_{ij} = 0$ for $i = 1, ..., m$. It is important to note that, due to the form of (3.8), this similarity does not mean that the global decentralized gain is obtained by setting to zero the entries of the unconstrained regulator gain corresponding to the null entries of the sparsity pattern. The solution is, in fact, much more intricate.

Remark 3.3. The sequence of gains that arises in Theorem 3.3 can only be computed backward in time. It is solved sequentially starting at $\tau = k+H-1$ with $\mathbf{P}(k+H)$ known by the boundary condition in (3.2). Then, the one-step solution is found by taking turns computing $\mathbf{K}(\tau)$ with (3.8) and $\mathbf{P}(\tau)$ with (3.2). Thus, for each time instant, the gain computation requires a window of the future dynamic matrices of the system to be known *a priori*. This is identical to the centralized solution for LTV systems. The application of this algorithm is, thus, possible either if one has a model of the evolution of the system with time or if it is used in combination with an online prediction algorithm.

Remark 3.4. The computation of the closed-form solution (3.8) requires $\mathcal{O}(n^4)$ floating point operations, using Gaussian elimination. Instead of using it, the exact numeric algorithm proposed in published work by the author [84] can be, alternatively, employed to compute each gain with a computational complexity

of $\mathcal{O}(|\chi|^3)$, where $|\chi|$ denotes the number of nonzero entries of $\mathbf{E}_{\mathcal{G}_c}$. Generally, in decentralized control applications, $|\chi| \approx cn$, where $c \in \mathbb{N}$ is a constant. This is the case for the networks of tanks considered in Sections 3.3 and 3.4 and for the LEO satellite mega-constellation considered in Chapter 5. Thus, it follows that a computational complexity of $\mathcal{O}(n^3)$ is achieved, which is equal to the computational complexity of the centralized solution. An efficient MATLAB implementation of the one-step solution can be found in the *DECENTER* toolbox, available at `https://decenter2021.github.io/documentation/LQROneStepLTV`.

Recall that the one-step relaxation stems from the analysis of the necessary condition of a constrained minimum of (2.6). The motivation behind this relaxation approach is to reduce the separation to the optimal solution. Nevertheless, it is also desirable to evaluate whether it has any physical meaning that can be leveraged to gather additional insight into the intricacies of the solution. In that regard, consider the following result.

Theorem 3.4. *The gain of the one-step solution presented in Theorem 3.3 is given by the solution to*

$$\begin{aligned} \underset{\mathbf{K}(\tau)\in\mathbb{R}^{m\times n}}{\textit{minimize}} \quad & \mathrm{tr}(\mathbf{P}(\tau)) \\ \textit{subject to} \quad & \mathbf{K}(\tau) \in \mathrm{Sparse}(\mathbf{E}_{\mathcal{G}_c}) \\ & (3.2), \end{aligned} \tag{3.9}$$

for $\tau = k, ..., k+H-1$*, where* $\mathbf{P}(\tau+1)$ *is given recurrently by* (3.2)*.*

Proof. See Section A.5. □

First, recall the definition of the cost-to-go in (3.5). Matrix $\mathbf{P}(\tau)$ can, thus, be intuitively regarded as a measure of the contribution of the correlation between the states of the systems to the cost-to-go. It is possible to notice from (3.9) that the one-step relaxation ignores the cross correlation components and attempts no minimize only the diagonal contributions. Second, similarly to the computation of the one-step solution according to Theorem 3.3, which is discussed in Remark 3.3, the optimization problems (3.9) are solved sequentially backward in time. In each time instant, the gain is obtained with the aim of minimizing $\mathrm{tr}(\mathbf{P}(\tau))$, which is related to $V_k(\tau)$ as previously discussed. Therefore, the one-step relaxation is greedy since it disregards the contribution of each gain to the overall cost of the whole window, which is given by $V_k(k) = J(k)$. Although this analysis suggests a degradation of performance, in what follows the one-step solution is shown to yield good performance, resorting to numerical simulations in Section 3.3 and experimental results in Section 3.4.

To sum up, the one-step relaxation is proposed to efficiently compute a sequence of gains that approximates the solution to the nonconvex optimization problem (2.6), which has to be solved at each time instant in a RHC scheme. Note that the problem is formulated globally, thus, the local decentralized controller gains can then be extracted from the globally synthesized sparse gain matrices, which allows for its decentralized implementation according to (2.3), leveraging local communication exclusively. It is very important to remark that the computation of the one-step solution according to Theorem 3.3 makes use of global matrices. For that reason, it follows that all-to-all communication for the one-step gain

synthesis procedure can only be avoided if the global computations are replicated in each computational unit. In this architecture, the global optimization problem (2.6) would have to be solved approximately in each computational unit at each time instant to extract the local gain corresponding to that system. Thus, according to Remark 3.4, the computational and memory complexity in each computational unit would grow with $\mathcal{O}(N^3)$ with respect to the dimension of the network. These requirements are not in line with the computational and memory feasibility constraints set forth in Section 2.2.3.

3.2 Extension to the tracking problem

More often than not, one is interested in tracking a reference signal, $\mathbf{r}(k)$, with the output of the system, $\mathbf{z}(k)$, instead of driving the output of the system to zero. On one hand, if the reference signal is feasible, in the sense that there exists a sequence of inputs that drive the output along the desired trajectory, the tracking problem degenerates into a regulation problem [85, Chapter 4]. In fact, in these conditions, it easy to define the tracking error dynamics and apply the one-step solution devised in Section 3.1 to regulate them. On the other hand, if the trajectory is not feasible, which is generally the case for piecewise constant reference signals, for instance, the extension is not as straightforward. This section details an approach to the design of a tracker, suitable both for centralized and decentralized configurations. For a decentralized configuration, the tracker is designed building on the results of the one-step relaxation put forward in Section 3.1.

3.2.1 Tracker design

The proposed tracker consists of a combination of feedforward and feedback terms. For each time instant, consider an equilibrium point consistent with the reference signal as if the system were LTI. The feedforward terms are designed to maintain such equilibrium, for each time instant as if the system were LTI, as well as to ensure the transition to the succeeding equilibrium point, since the system is, in fact, LTV. Writing the dynamics of the system alongside the feedforward terms yields an LTV system for the dynamics of the tracking error. The one-step solution, presented in Section 3.1, is then applied to the error dynamics, from which a feedback term stems. The origin and purpose of each of the different terms is made clearer in the following derivation.

It is well known that for perfect tracking to be possible one needs, in general, as many inputs as the dimension of the vector to track [86, Theorem 3.14], *i.e.*, $m = o$. For this reason, the tracker designed in this work assumes that the reference and input vectors have the same dimension, *i.e.*, $\mathbf{r}(k) \in \mathbb{R}^m$. Furthermore, it is assumed that, for $k \in \mathbb{N}_0$, $\mathbf{H}(k)$ has full rank, *i.e.*, $\operatorname{rank}(\mathbf{H}(k)) = m$, which is necessary if one aims to follow an arbitrary reference of dimension m.

First, an equilibrium point for time instant k, if the system were LTI, that follows the reference signal

$\mathbf{r}(k)$ is sought. To that purpose, define $\bar{\mathbf{x}}(k) \in \mathbb{R}^n$ and $\bar{\mathbf{u}}(k) \in \mathbb{R}^m$ such that

$$\begin{cases} \bar{\mathbf{x}}(k) = \mathbf{A}(k)\bar{\mathbf{x}}(k) + \mathbf{B}(k)\bar{\mathbf{u}}(k) \\ \mathbf{H}(k)\bar{\mathbf{x}}(k) = \mathbf{r}(k) \end{cases} \tag{3.10}$$

is satisfied for $k \in \mathbb{N}_0$. The augmented matrix of the linear system of equations (3.10) is given by

$$\left[\begin{array}{cc|c} \mathbf{A}(\tau) - \mathbf{I} & \mathbf{B}(\tau) & \mathbf{0} \\ \mathbf{H}(\tau) & \mathbf{0} & \mathbf{r}(\tau) \end{array}\right] . \tag{3.11}$$

It is possible to note that, provided that $\mathbf{H}(\tau)$ has full rank, then the rank of the augmented matrix is equal to the rank of the coefficient matrix, *i.e.*, the submatrix on the left of (3.11). It follows directly from Rouché–Capelli theorem [87, Theorem 2.38] that (3.10) has, at least, one solution $(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau))$. However, the tracker design that is proposed herein encompasses a more integrated approach, as it will be seen shortly, and the solution of (3.11) is not explicitly shown at this point. The proposed approach aims to drive the system to follow the sequence of pairs $(\bar{\mathbf{x}}(k), \bar{\mathbf{u}}(k))$, penalizing only the error in the tracking space. Define the tracking error in the space of the system state as $\mathbf{e}(k) := \mathbf{x}(k) - \bar{\mathbf{x}}(k)$. Using (2.2) and (3.10) allows to write the dynamics of the tracking error as

$$\mathbf{e}(k+1) = \mathbf{A}(k)\mathbf{e}(k) + \mathbf{B}(k)(\mathbf{u}(k) - \bar{\mathbf{u}}(k)) - (\bar{\mathbf{x}}(k+1) - \bar{\mathbf{x}}(k)) \ . \tag{3.12}$$

The regulator cannot be applied to (3.12) because of the presence of the last term. For that reason, one may attempt to write the difference $\bar{\mathbf{x}}(k+1) - \bar{\mathbf{x}}(k)$ as the combination of an additional feedforward control action, $\mathbf{u_a}(k)$, and a disturbance, $\mathbf{d}(k)$, *i.e.*,

$$\bar{\mathbf{x}}(k+1) - \bar{\mathbf{x}}(k) = \mathbf{B}(k)\mathbf{u_a}(k) + \mathbf{d}(k) \ , \tag{3.13}$$

which allows to rewrite the tracking error dynamics (3.12) as

$$\mathbf{e}(k+1) = \mathbf{A}(k)\mathbf{e}(k) + \mathbf{B}(k)(\mathbf{u}(k) - \bar{\mathbf{u}}(k) - \mathbf{u_a}(k)) - \mathbf{d}(k) \ . \tag{3.14}$$

Note that the tracking error, $\mathbf{e}(k+1)$, defined in this formulation, is given by the difference between the actual state of the system and $\bar{\mathbf{x}}(k+1)$, not just the difference between the output of the system and the reference signal. For that reason, the main concern is the minimization of the component of the error in the tracking space, *i.e.*, the column space of $\mathbf{H}(k+1)$. Therefore, instead of choosing $\mathbf{u_a}(k)$ such that the norm of the disturbance is minimal, it is selected such that the component of the disturbance in the tracking space is minimized. The design of the reference and feedforward terms then takes the form of

a quadratic optimization problem with linear equality constraints, for a finite window, given by

$$\begin{aligned} &\underset{\substack{\bar{\mathbf{x}}(\tau),\bar{\mathbf{u}}(\tau),\tau=k,\ldots,k+H \\ \mathbf{u_a}(\tau),\tau=k,\ldots,k+H-1}}{\text{minimize}} && \sum_{\tau=k}^{k+H-1} ||\mathbf{H}(\tau+1)\mathbf{d}(\tau)||^2 \\ &\text{subject to} && \begin{cases} \bar{\mathbf{x}}(\tau) = \mathbf{A}(\tau)\bar{\mathbf{x}}(\tau) + \mathbf{B}(\tau)\bar{\mathbf{u}}(\tau) \\ \mathbf{H}(\tau)\bar{\mathbf{x}}(\tau) = \mathbf{r}(\tau) \end{cases}, \tau = k, \ldots, k+H\,. \end{aligned} \tag{3.15}$$

Using (3.13) to expand the cost function of the optimization problem (3.15) yields a quadratic expression. Thus, it is possible to find the optimal solution for this optimization problem using well-known optimization techniques, as shown in the following result.

Theorem 3.5. *There is either one or infinitely many solutions, all globally optimal, to the optimization problem* (3.15)*, which are given by the solutions to the system of linear equations*

$$\mathbf{G}\bar{\chi} = \bar{\mathbf{r}}\,, \tag{3.16}$$

where $\bar{\chi} \in \mathbb{R}^{((3m+2n)H+2m+2n)}$ *corresponds to*

$$\bar{\chi} = \Big[\bar{\mathbf{x}}^T(k)\,\bar{\mathbf{u}}^T(k)\,\mathbf{u}_{\mathbf{a}}^T(k)\,\boldsymbol{\lambda}^T(k)\boldsymbol{\gamma}^T(k)\ldots\boldsymbol{\lambda}^T(k+H-1)\;\boldsymbol{\gamma}^T(k+H-1)\,\bar{\mathbf{x}}^T(k+H)\;\bar{\mathbf{u}}^T(k+H)\,\boldsymbol{\lambda}^T(k+H)\boldsymbol{\gamma}^T(k+H)\Big]^T,$$

where $\boldsymbol{\lambda}(\tau) \in \mathbb{R}^n$ *and* $\boldsymbol{\gamma}(\tau) \in \mathbb{R}^m$ *are Lagrange multipliers. The vector* $\bar{\mathbf{r}} \in \mathbb{R}^{((3m+2n)H+2m+2n)}$ *is written as*

$$\bar{\mathbf{r}} = \begin{bmatrix} \mathbf{0}_{1\times(2m+2n)} & \mathbf{r}^T(k) & \ldots & \mathbf{0}_{1\times(2m+2n)} & \mathbf{r}^T(k+H-1) & \mathbf{0}_{1\times(m+2n)} & \mathbf{r}^T(k+H) \end{bmatrix}^T\,,$$

and $\mathbf{G} \in \mathbb{R}^{((3m+2n)H+2m+2n)\times((3m+2n)H+2m+2n)}$ *is a symmetric block tridiagonal matrix given by*

$$\mathbf{G} = \begin{bmatrix} \bar{\boldsymbol{\alpha}}_k & \bar{\boldsymbol{\beta}}_k & & & \mathbf{0} \\ \bar{\boldsymbol{\beta}}_k^T & \bar{\boldsymbol{\alpha}}_{k+1} & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \bar{\boldsymbol{\alpha}}_{k+T-1} & \bar{\boldsymbol{\beta}}_{k+T-1} \\ \mathbf{0} & & & \bar{\boldsymbol{\beta}}_{k+T-1}^T & \bar{\boldsymbol{\alpha}}_{k+T} \end{bmatrix},$$

with

$$\bar{\boldsymbol{\alpha}}_k = \begin{bmatrix} \mathbf{H}^T(k+1)\mathbf{H}(k+1) & \mathbf{0}_{n\times m} & \mathbf{H}^T(k+1)\mathbf{H}(k+1)\mathbf{B}(k) & \mathbf{A}^T(k)-\mathbf{I} & \mathbf{H}^T(k) \\ \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m} & \mathbf{0}_{m\times m} & \mathbf{B}^T(k) & \mathbf{0}_{m\times m} \\ \mathbf{B}^T(k)\mathbf{H}^T(k+1)\mathbf{H}(k+1) & \mathbf{0}_{m\times m} & \mathbf{B}^T(k)\mathbf{H}^T(k+1)\mathbf{H}(k+1)\mathbf{B}(k) & \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m} \\ \mathbf{A}(k)-\mathbf{I} & \mathbf{B}(k) & \mathbf{0}_{n\times m} & \mathbf{0}_{n\times n} & \mathbf{0}_{n\times m} \\ \mathbf{H}(k) & \mathbf{0}_{m\times m} & \mathbf{0}_{m\times m} & \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m} \end{bmatrix},$$

$$\bar{\boldsymbol{\alpha}}_{\tau} = \begin{bmatrix} \mathbf{H}^T(\tau{+}1)\mathbf{H}(\tau{+}1) + \mathbf{H}^T(\tau)\mathbf{H}(\tau) & \mathbf{0}_{n\times m} & \mathbf{H}^T(\tau{+}1)\mathbf{H}(\tau{+}1)\mathbf{B}(\tau) & \mathbf{A}^T(\tau) - \mathbf{I} & \mathbf{H}^T(\tau) \\ \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m} & \mathbf{0}_{m\times m} & \mathbf{B}^T(\tau) & \mathbf{0}_{m\times m} \\ \mathbf{B}^T(\tau)\mathbf{H}^T(\tau{+}1)\mathbf{H}(\tau{+}1) & \mathbf{0}_{m\times m} & \mathbf{B}^T(\tau)\mathbf{H}^T(\tau{+}1)\mathbf{H}(\tau{+}1)\mathbf{B}(\tau) & \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m} \\ \mathbf{A}(\tau) - \mathbf{I} & \mathbf{B}(\tau) & \mathbf{0}_{n\times m} & \mathbf{0}_{n\times n} & \mathbf{0}_{n\times m} \\ \mathbf{H}(\tau) & \mathbf{0}_{m\times m} & \mathbf{0}_{m\times m} & \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m} \end{bmatrix},$$

for $\tau = k+1, \ldots, k+H-1$,

$$\bar{\boldsymbol{\alpha}}_{k+H} = \begin{bmatrix} \mathbf{H}^T(k{+}H)\mathbf{H}(k{+}H) & \mathbf{0}_{n\times m} & \mathbf{A}^T(k{+}H) - \mathbf{I} & \mathbf{H}^T(k{+}H) \\ \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m} & \mathbf{B}^T(k{+}H) & \mathbf{0}_{m\times m} \\ \mathbf{A}(k{+}H) - \mathbf{I} & \mathbf{B}(k{+}H) & \mathbf{0}_{n\times n} & \mathbf{0}_{n\times m} \\ \mathbf{H}(k{+}H) & \mathbf{0}_{m\times m} & \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m} \end{bmatrix},$$

and

$$\bar{\boldsymbol{\beta}}_{\tau} = \begin{bmatrix} -\mathbf{H}^T(\tau+1)\mathbf{H}(\tau+1) & \mathbf{0}_{n\times m} & \mathbf{0}_{m\times m} & \mathbf{0}_{n\times n} & \mathbf{0}_{n\times m} \\ \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m} & \mathbf{0}_{m\times m} & \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m} \\ -\mathbf{B}^T(\tau)\mathbf{H}^T(\tau+1)\mathbf{H}(\tau+1) & \mathbf{0}_{m\times m} & \mathbf{0}_{m\times m} & \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m} \\ \mathbf{0}_{n\times n} & \mathbf{0}_{n\times m} & \mathbf{0}_{n\times m} & \mathbf{0}_{n\times n} & \mathbf{0}_{n\times m} \\ \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m} & \mathbf{0}_{m\times m} & \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m} \end{bmatrix},$$

for $\tau = k, \ldots, k+H-1$.

Proof. See Section A.6 in Appendix A. □

Remark 3.5. Note that (3.10) is a system of linear equations with $n+m$ constraints and $n+m$ unknowns. If $\mathbf{H}(\tau)\mathbf{B}(\tau)$ is invertible, multiplying the first equation of (3.10) by $\mathbf{H}(\tau)$ and making use of the second equation yields $\bar{\mathbf{u}}(\tau) = (\mathbf{H}(\tau)\mathbf{B}(\tau))^{-1}(\mathbf{r}(\tau) - \mathbf{H}(\tau)\mathbf{A}(\tau)\bar{\mathbf{x}}(\tau))$. Substituting this for $\bar{\mathbf{u}}(\tau)$ in the first equation of (3.10) and solving for $\bar{\mathbf{x}}(\tau)$ it is possible to conclude that, for the particular case for which $\mathrm{rank}(\mathbf{H}(\tau)\mathbf{B}(\tau)) = m$ and $\mathrm{rank}(\mathbf{I} - \mathbf{A}(\tau) + \mathbf{B}(\tau)(\mathbf{H}(\tau)\mathbf{B}(\tau))^{-1}\mathbf{H}(\tau)\mathbf{A}(\tau)) = n$, the solution of (3.10) is unique and given by

$$\begin{cases} \bar{\mathbf{x}}(\tau) = \left[\mathbf{I} - \mathbf{A}(\tau) + \mathbf{B}(\tau)(\mathbf{H}(\tau)\mathbf{B}(\tau))^{-1}\mathbf{H}(\tau)\mathbf{A}(\tau)\right]^{-1}\mathbf{B}(\tau)(\mathbf{H}(\tau)\mathbf{B}(\tau))^{-1}\mathbf{r}(\tau) \\ \bar{\mathbf{u}}(\tau) = (\mathbf{H}(\tau)\mathbf{B}(\tau))^{-1}(\mathbf{r}(\tau) - \mathbf{H}(\tau)\mathbf{A}(\tau)\bar{\mathbf{x}}(\tau)) \end{cases}. \tag{3.17}$$

For this reason, the linear equality constraint fully defines $(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau))$ for the time instant τ. Moreover, if $\mathrm{rank}(\mathbf{H}(\tau+1)\mathbf{B}(\tau)) = m$, it is possible to achieve $\mathbf{H}(\tau+1)\mathbf{d}(k) = 0$. The feedforward term $\mathbf{u_a}(k)$ is, thus, computed as follows

$$\begin{aligned} \mathbf{H}(k+1)\left(\bar{\mathbf{x}}(k+1) - \bar{\mathbf{x}}(k)\right) &= \mathbf{H}(k+1)\mathbf{B}(k)\mathbf{u_a}(k) + \mathbf{H}(k+1)\mathbf{d}(k) \\ \iff \mathbf{u_a}(k) &= (\mathbf{H}(k+1)\mathbf{B}(k))^{-1}(\mathbf{r}(k+1) - \mathbf{H}(k+1)\bar{\mathbf{x}}(k)) \end{aligned}. \tag{3.18}$$

Therefore, for every time instant τ for which one or both of these particular cases are verified, the system of linear equations (3.16) may be reduced, allowing for a decrease in computational load.

Remark 3.6. The matrix $\mathbf{G}$ in the system of linear equations (3.16) is a symmetric block tridiagonal matrix. There are plenty of algorithms to solve systems of linear equations featuring such sort of matrices based on cyclic reduction [88], which allows parallelization [89].

An regulator, either decentralized using the method put forward in Section 3.1, or centralized, may be applied to the tracking error dynamics, given by (3.14). The control action for the output tracking problem is, then, given by

$$\mathbf{u}(k) = -\mathbf{K}(k)(\mathbf{x}(k) - \bar{\mathbf{x}}(k)) + \bar{\mathbf{u}}(k) + \mathbf{u_a}(k)\,, \tag{3.19}$$

where $\mathbf{K}(k)$ is the regulator feedback gain.

It is interesting to remark the effect of each of the three terms that make up the control action (3.19). First, $\bar{\mathbf{u}}(k)$ is a feedforward term that allows to maintain the output of the system constant and equal to the reference signal of the current time step, $\mathbf{r}(k)$, if the system were LTI, with dynamics defined by $\mathbf{A}(k)$ and $\mathbf{B}(k)$. Second, $\mathbf{u_a}(k)$ is another feedforward term that compensates the change in the reference signal, which is time-varying, in the following time step. Third, $-\mathbf{K}(k)(\mathbf{x}(k) - \bar{\mathbf{x}}(k))$ is a feedback term that drives to zero the component in the tracking space of the difference between the state of the system and $\bar{\mathbf{x}}(k)$. It is also interesting to point out that, if there is a feasible trajectory that follows the reference signal, then there exists $\mathbf{u_a}(k)$ for which the disturbance $\mathbf{d}(k)$ in (3.14) is null. Therefore, the feedback term vanishes as time goes to infinity and the actuation tends to be given by both feedforward terms. On top of that, if the reference is a step function, then the term $\mathbf{u_a}(k)$ vanishes as well.

3.2.2 Addition of integral action

If the model of the system is exact and the reference signal is a feasible trajectory, then the proposed output tracking design achieves null steady-state error. However, in the overwhelming majority of real scenarios that is not the case. Having that in mind, to improve steady-state performance and add robustness to the controller, an integral action feedback term may be included.

Consider new state variables that correspond to the integral of the tracking error, given by

$$\mathbf{x}_I(k) = \sum_{\tau=0}^{k} (\mathbf{z}(\tau) - \mathbf{r}(\tau)) = \sum_{\tau=0}^{k} \mathbf{H}(\tau)\mathbf{e}(\tau)\,.$$

Then, making use of the error dynamics (3.14), one can define the augmented system

$$\begin{bmatrix} \mathbf{e}(k+1) \\ \mathbf{x}_I(k+1) \end{bmatrix} = \boldsymbol{\mathcal{A}}(k) \begin{bmatrix} \mathbf{e}(k) \\ \mathbf{x}_I(k) \end{bmatrix} + \boldsymbol{\mathcal{B}}(k) \left(\mathbf{u}(k) - \bar{\mathbf{u}}(k) - \mathbf{u_a}(k)\right) - \begin{bmatrix} \mathbf{I}_n \\ \mathbf{H}(k+1) \end{bmatrix} \mathbf{d}(k)\,, \tag{3.20}$$

where

$$\boldsymbol{\mathcal{A}}(k) = \begin{bmatrix} \mathbf{A}(k) & \mathbf{0} \\ \mathbf{H}(k+1)\mathbf{A}(k) & \mathbf{I} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\mathcal{B}}(k) = \begin{bmatrix} \mathbf{B}(k) \\ \mathbf{H}(k+1)\mathbf{B}(k) \end{bmatrix}\,.$$

One can, now, add the integral action feedback term applying the regulator to the augmented system

(3.20) with weighting matrices $\boldsymbol{\mathcal{Q}}(k)$ and $\boldsymbol{\mathcal{R}}(k)$ given by

$$\boldsymbol{\mathcal{Q}}(k) = \begin{bmatrix} \mathbf{H}^T(k)\mathbf{Q}(k)\mathbf{H}(k) & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_I(k) \end{bmatrix} \text{ and } \quad \boldsymbol{\mathcal{R}}(k) = \mathbf{R}(k)\,,$$

where $\mathbf{Q}_I(k) \in \mathbb{R}^{m\times m}$ is the positive semidefinite matrix that weights the integral of the tracking error. Note, again, that this procedure is suitable both for a centralized configuration and for a decentralized configuration using the one-step design synthesis method put forward in Section 3.1. The resulting gain matrix has the form $\boldsymbol{\mathcal{K}}(k) = [\mathbf{K}(k) \quad \mathbf{K}_I(k)] \in \mathbb{R}^{m\times(n+m)}$, where $\mathbf{K}(k)$ and $\mathbf{K}_I(k) \in \mathbb{R}^{m\times m}$ are the error and integral action feedback gains, respectively. The control input of the proposed approach is, thus, given by

$$\mathbf{u}(k) = -\mathbf{K}(k)(\mathbf{x}(k) - \bar{\mathbf{x}}(k)) - \mathbf{K}_I(k)\mathbf{x}_I(k) + \bar{\mathbf{u}}(k) + \mathbf{u_a}(k)\,. \tag{3.21}$$

If there is a sudden significant change in the reference signal that cannot be attained by the system, the tracking error is significant until the controller has enough time to drive the output of the system to the reference. During that period, the integral states keep accumulating the tracking error, which leads to what is know as integral windup, resulting in a significant overshoot until the integral states decrease, in absolute value, to normal operation values. In this design, the integral action term is only included to provide for better performance after the transient state, rejecting disturbances. Thus, a very simple and effective anti-windup technique for this formulation is to saturate the integral state variables and disable the integrators while they are saturated. That is, the integral state is subject to an entrywise saturation

$$[\mathbf{x}_I(k)]_i = \begin{cases} [\mathbf{x}_I(k)]_i\,, & |[\mathbf{x}_I(k)]_i| \leq [\mathbf{x}_I^{\text{sat}}]_i \\ [\mathbf{x}_I^{\text{sat}}]_i \operatorname{sgn}\left([\mathbf{x}_I(k)]_i\right), & |[\mathbf{x}_I(k)]_i| > [\mathbf{x}_I^{\text{sat}}]_i \end{cases}, \; i = 1, ..., m\,,$$

where $\mathbf{x}_I^{\text{sat}} \in \mathbb{R}^m$ is a constant vector that holds the saturation limits of each of the corresponding entries of $\mathbf{x}_I(k)$. To disable the integrator whose state is saturated, one can subtract the difference between the integral state before and after the saturation from the corresponding integral state. The drawback of this anti-windup technique is that the magnitude of the perturbations that the controller is able to reject is limited by the chosen saturation limits. For this reason, the entries of $\mathbf{x}_I^{\text{sat}}$ are selected to be the minimum possible that still allow for the rejection of the perturbations with the expected highest magnitude that the system is subject to. Figure 3.1 depicts the block diagram of the proposed tracking system.

3.3 Numeric validation

The goal of this section is to validate the one-step convex relaxation approach and to show the scalability of the proposed methods resorting to numeric simulations. The methods put forward in this chapter are applied to a network of N tanks, which corresponds to the generalization of the quadruple-tank network introduced in [90]. This network is analogous to various industrial processes, which are among the vast range of applications modeled by large-scale networks of interconnected systems. Given that the

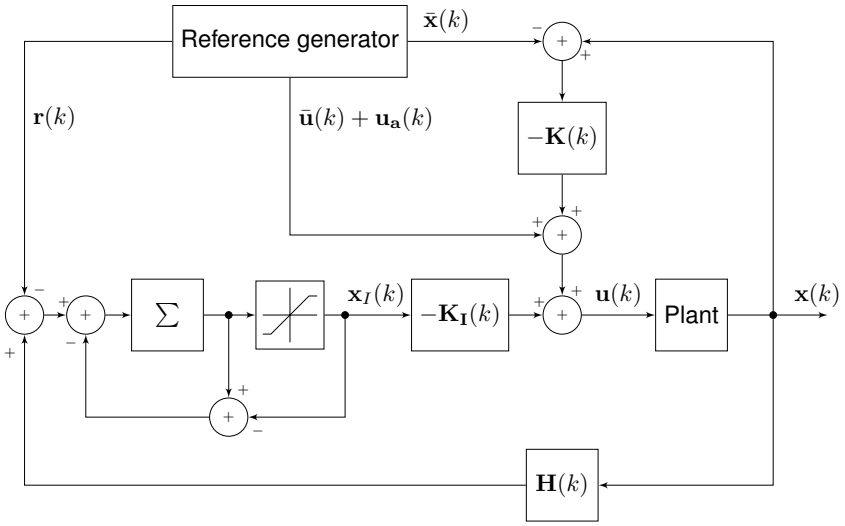


Figure 3.1: Block diagram of the proposed tracking system, with anti-windup integral action.

dynamics of the projected system are nonlinear, to employ the methods devised one can approximate its behavior by an LTV system, linearizing and discretizing its dynamics about successive equilibrium points.

3.3.1 N tanks network dynamics

Consider N interconnected tanks, as shown in Figure 3.2, where N is an even integer. The water level of tank i is denoted by h_i. The network is actuated by $N/2$ pumps, which are controlled by the lower tanks, whose inputs are denoted by u_i for $i = 1, ..., N/2$, in accordance with the schematic. Each pump is connected to a three-way valve that regulates the fraction of the flow, held constant, that goes to each of the tanks supplied by the pump. Each tank has a sensor, which measures its water level. Making use of mass balances and Bernoulli's law, the system dynamics, in the absence of noise, are given by

$$\begin{cases} A_i \dot{h}_i(t) = -a_i\sqrt{2gh_i(t)} + a_{\frac{N}{2}+i}\sqrt{2gh_{\frac{N}{2}+i}(t)} + \gamma_i k_i u_i(t), \;\; i = 1, ..., N/2 \\ A_i \dot{h}_i(t) = -a_i\sqrt{2gh_i(t)} + (1-\gamma_{i-\frac{N}{2}-1})k_{i-\frac{N}{2}-1}u_{i-\frac{N}{2}-1}(t), \;\; i = N/2+2, ..., N \\ A_{\frac{N}{2}+1}\dot{h}_{\frac{N}{2}+1}(t) = -a_{\frac{N}{2}+1}\sqrt{2gh_{\frac{N}{2}+1}(t)} + (1-\gamma_{\frac{N}{2}})k_{\frac{N}{2}}u_{\frac{N}{2}}(t) \end{cases}, \tag{3.22}$$

where A_i and a_i are the cross sections of tank i and of its outlet hole, respectively; the constant γ_i represents the fraction of the flow that passes through the valve i to the lower tanks; k_i is the constant of proportionality between the flow and the input of pump i; and g denotes the acceleration of gravity. Furthermore, the input of each pump is subject to a hard constraint $u_i \in [0, u^{\text{sat}}]$, where $u^{\text{sat}} \in \mathbb{R}^+$.

The nonlinear dynamics (3.22) are linearized about a given equilibrium point, characterized by equilibrium water levels, h_i^0, $i = 1, ..., N$; and inputs u_i^0, $i = 1, ..., N/2$. Writing the state and control vectors,

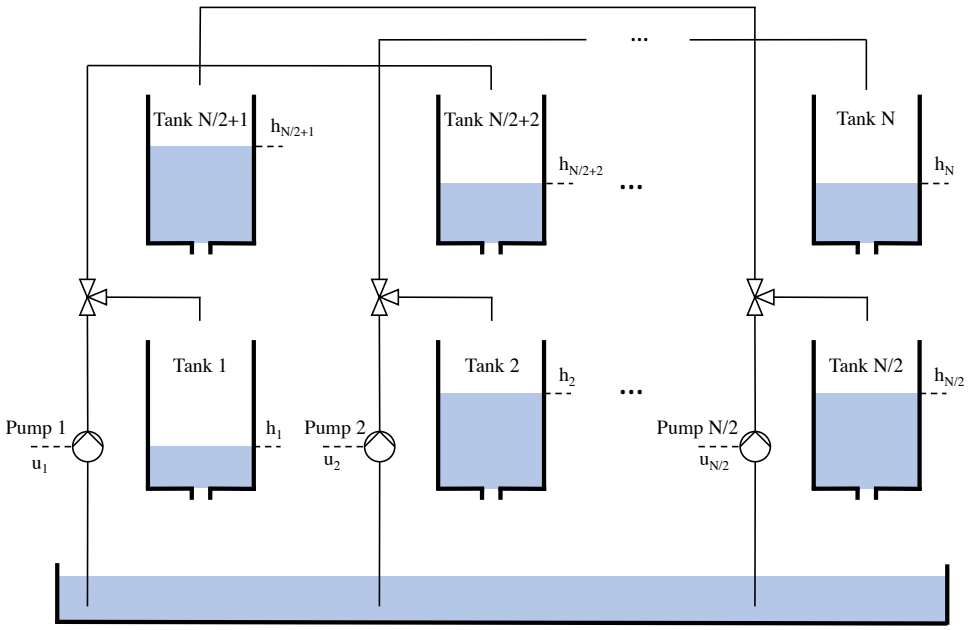


Figure 3.2: Schematic of the N tanks network.

respectively, as

$$\mathbf{x_c}(t) = \begin{bmatrix} h_1(t) - h_1^0 \\ \vdots \\ h_N(t) - h_N^0 \end{bmatrix} \quad \text{and} \quad \mathbf{u_c}(t) = \begin{bmatrix} u_1(t) - u_1^0 \\ \vdots \\ u_{\frac{N}{2}}(t) - u_{\frac{N}{2}}^0 \end{bmatrix},$$

the continuous-time linearized system is given by

$$\dot{\mathbf{x}}_\mathbf{c}(t) = \mathbf{A_c}(t)\mathbf{x_c}(t) + \mathbf{B_c}(t)\mathbf{u_c}(t) \tag{3.23}$$

with $\mathbf{A_c}(t) \in \mathbb{R}^{N\times N}$ and $\mathbf{B_c}(t) \in \mathbb{R}^{N\times N/2}$ given by

$$[\mathbf{A_c}(t)]_{ij} = \begin{cases} -1/T_i & , i = j \\ \frac{A_j}{A_i T_j} & , j = i + N/2 \\ 0 & , \text{otherwise} \end{cases},$$

$$[\mathbf{B_c}(t)]_{ij} = \begin{cases} \gamma_i k_i / A_i & , i = j \\ (1-\gamma_j)k_j/A_i & , j = i - N/2 - 1 \\ (1-\gamma_j)k_j/A_i & , i = N/2+1,\ \ j = N/2 \\ 0 & , \text{otherwise} \end{cases},$$

where T_i is the time constant of tank i, given by

$$T_i = \frac{A_i}{a_i}\sqrt{\frac{2h_i^0}{g}}.$$

Provided that this system is slow, one can assume that the water level measurements and control

inputs are updated with a constant period T_c. Under this assumption, the discretization of (3.23) yields

$$\mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) \,,$$

where

$$\mathbf{A}(k) = e^{\mathbf{A_c}(kT_c)T_c} \,,$$
$$\mathbf{B}(k) = \left(\int_0^{T_c} e^{\mathbf{A_c}(kT_c)\tau} d\tau \right) \mathbf{B_c}(kT_c),$$

and

$$\mathbf{u}(k) = \mathbf{u_c}(kT_c) \,.$$

It is important to remark that to perform the linearization, each local controller ought to access the necessary variables of the network that define the equilibrium point, through communication. Provided that the water levels change slowly, it may be carried out with a given periodicity, $T_{lin} = qT_c$, where q is an integer number, thereby reducing the computational load.

3.3.2 Controller implementation

The problem considered for this network is the design of a decentralized solution to control the water level of the lower tanks. For that reason, the output of the network is computed as in (2.2), using $\mathbf{H}(k) = \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{0}_{N/2} \end{bmatrix}$. Thus, the output dynamics are decoupled and the output coupling graph $\mathcal{G}_o$ is constituted by self-loops exclusively. The water level of the lower tanks are the variables used to define each equilibrium point, as a means of ensuring that the system dynamics used for their control are as accurate as possible. Recall, from Section 3.1, that the global computations of the one-step solution have to be replicated in each local computational unit. For this reason, each time a new linearization is performed, every computational unit has to receive through communication the water level of the lower tanks, compute the remaining variables that define the equilibrium point, and linearize the relevant entries of matrix $\mathbf{A}(k)$ about that point. It is import to note that this additional communication requirement for linearization purposes arises from the nonlinear nature of the plant. If the plant were LTV, all-to-all communication would not be necessary. The controller design approach consists of a local controller in each of the lower tanks, which computes the control action of the associated pump, making use of the measurement of its own water level only. Thus, the state feedback communication graph $\mathcal{G}_c$ is constituted by self-loops exclusively. The control action of the decentralized controllers is computed using the tracker design proposed in Section 3.2, that makes use of the one-step method developed in Section 3.1 to compute the feedback gains. Given that the reference signal one desires to track in the simulations is not a feasible trajectory, an integral feedback term is also included alongside an anti-windup technique, as detailed in Section 3.2.2, to improve the steady-state performance. Thus, for the

linearized system, each local controller computes an input of the form

$$[\mathbf{u}(k)]_i = -[\mathbf{K}(k)]_{ii}\left([\mathbf{x}(k)]_i - [\bar{\mathbf{x}}(k)]_i\right) - [\mathbf{K}_I(k)]_{ii}[\mathbf{x}_I(k)]_i + [\bar{\mathbf{u}}(k)]_i + [\mathbf{u_a}(k)]_i$$

for $i = 1, ..., N/2$, in such a way that each tank has only access to its measured water level and to the integral of its tracking error. Note that $u_i(k)$ and $[\mathbf{u}(k)]_i$ are distinct, the former is the input to pump i and the latter is the i-th component of $\mathbf{u}(k)$. It is important to remark that the dynamics of the quadruple-tank network verify the two conditions explored in Remark 3.5 for every time instant. Thus, the reference values $(\bar{\mathbf{x}}(k), \bar{\mathbf{u}}(k), \mathbf{u_a}(k))$ can be computed independently using (3.17) and (3.18) for each time instant. Furthermore, $(\bar{\mathbf{x}}(k), \bar{\mathbf{u}}(k))$ does not have, necessarily, to be calculated with (3.17) using the linearized system. In fact, the analogous nonlinear equation is

$$\begin{cases} 0 = -a_i\sqrt{2g\,\bar{h}_i(k)} + a_{\frac{N}{2}+i}\sqrt{2g\,\bar{h}_{\frac{N}{2}+i}(k)} + \gamma_i k_i \bar{u}_i(k), & i = 1, ..., N/2 \\ 0 = -a_i\sqrt{2g\,\bar{h}_i(k)} + (1-\gamma_{i-\frac{N}{2}-1})k_{i-\frac{N}{2}-1}\bar{u}_{i-\frac{N}{2}-1}(k), & i = N/2+2, ..., N \\ 0 = -a_{\frac{N}{2}+1}\sqrt{2g\,\bar{h}_{\frac{N}{2}+1}(k)} + (1-\gamma_{\frac{N}{2}})k_{\frac{N}{2}}\bar{u}_{\frac{N}{2}}(k) & \\ \bar{h}_i(k) = [\mathbf{r}(k)]_i, & i = 1, ..., N/2 \end{cases}, \tag{3.24}$$

with

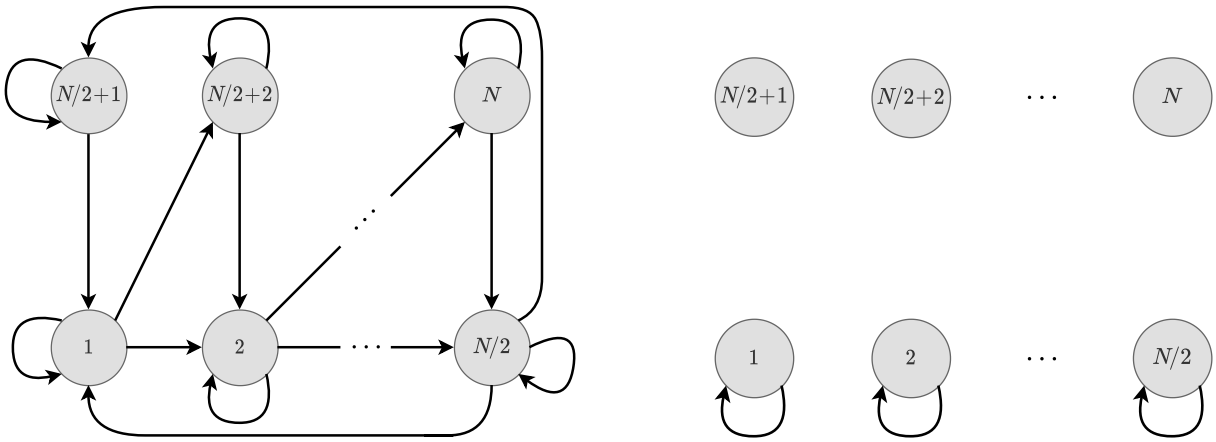
$$[\bar{\mathbf{x}}(k)]_i = \bar{h}_i(k) - h_i^0(k)\,, \quad [\bar{\mathbf{u}}(k)]_i = \bar{u}_i(k) - u_i^0(k),$$

for $i = 1, \ldots, N$, where $h_i^0(k)$ and $u_i^0(k)$ are the equilibrium water level of tank i and equilibrium input of pump i, respectively, computed in the last linearization prior to the time instant k. The unique closed-form solution to (3.24) can found by algebraic manipulation, as detailed in [79]. It is also interesting to point out that the closed-form solution to (3.24) can also be used to compute the equilibrium point for each linearization, given that the water levels of the lower tanks are the variables chosen to define the equilibrium point. Note that the use of the nonlinear equilibrium solution to find $\bar{\mathbf{x}}(k)$ and $\bar{\mathbf{u}}(k)$ is not only computationally efficient but also much more accurate because it does not rely on the linearization, which is only updated every q time steps. The actual pump input is, thus, computed making use of

$$u_i(k) = -[\mathbf{K}(k)]_{ii}\left(h_i(k) - \bar{h}_i(k)\right) - [\mathbf{K}_I(k)]_{ii}[\mathbf{x}_I(k)]_i + [\bar{\mathbf{u}}(k)]_i + [\mathbf{u_a}(k)]_i + u_i^0(k)\,, \tag{3.25}$$

for $i = 1, \ldots, N/2$. Although the controller design put forward is not projected to handle hard input constraints, as this network requires, to meet those constraints on the control inputs of the pumps, the computed inputs, given by (3.25), are saturated. Comparing the local controller (3.25) with the global controller (3.21), it follows that the feedback gain of the augmented system, $\mathcal{K}(k) = [\mathbf{K}(k) \quad \mathbf{K}_I(k)]$, must follow the sparsity pattern defined by $\mathbf{E}_{\mathcal{G}_c} = \begin{bmatrix}\mathbf{I}_{N/2} & \mathbf{0}_{N/2} & \mathbf{I}_{N/2}\end{bmatrix}$. The scheme of the dynamic, output, and communication topologies of the network of N tanks is depicted in Figure 3.3.

Given that both the centralized and one-step methods require the dynamics of the system in a time window that spans future instants, and considering that the dynamics of the network vary with its state vector, it is not possible to simulate this method online without the use of a mechanism that predicts the



(a) $\mathcal{G}_d$. (b) $\mathcal{G}_o = \mathcal{G}_c$.

Figure 3.3: Scheme of the dynamic, output, and communication topologies of the network of N tanks.

future evolution of the state vector, thus allowing to obtain the linearized dynamics. For that reason, a very simple iterative linear quadratic regulator (iLQR) smoothing scheme, based on [91, 92], is used. This approach is an iterative algorithm, applied to a finite window H, which consists of backward and forward passes carried out in turns, as a means of computing the control action for the nonlinear system. For the initialization of the algorithm, start with an initial control input and the resulting trajectory for the window considered, that is, $\mathbf{u}(\tau)$ for $\tau = k, ..., k + H - 1$ and $\mathbf{x}(\tau)$ for $\tau = k, ..., k + H$. From this initial trajectory, one can linearize the system dynamics around successive equilibrium points and find an LTV system that approximates the nonlinear system throughout the window. In this particular case, for the initialization, an initial control input was chosen such that the water level in the lower tanks remains constant throughout the window, which approximates the nonlinear dynamics by an LTI system, due to the way the linearization was defined. Then run the backward and forward passes, in this order, in turns, until convergence is reached. The backward pass consists of the computation of the control action, using (3.25), considering the approximate LTV system given by the previous forward pass. Note that the equilibrium water levels in the lower tanks and the feedforward terms of the control input only have to be computed once, since they do not depend on the evolution of the system. The forward pass is the simulation of the nonlinear system using the control action computed in the previous backward pass, whose trajectory is used to update the LTV system that approximates the dynamics of the nonlinear system throughout the window. The algorithm stops when the maximum relative difference of the norm of the actuation throughout the window, computed in two consecutive iterations, is below 10^{-4}. The linearizations performed in each iteration are also carried out with a periodicity T_{lin}. After convergence has been reached, one can make use of the first d gains of the window. A new window $\{k + d, ..., k + d + H - 1\}$ is then defined and so forth. A flowchart of the iLQR scheme used for the gain computation for a finite window is presented in Figure 3.4.

Alternatively, the linearization could have been carried out about the equilibrium points defined by the reference signal, at the expense of less accurate model dynamics. This linearization scheme would not require all-to-all communication for the linearization. Nevertheless, since the goal of this section is the validation of the convex relaxation proposed in Section 3.1 for LTV dynamics, the iLQR approach is

followed to assess the performance of the relaxation for accurate model dynamics. In Chapters 4 and 5 the focus is turned to the distributed implementation aspects.

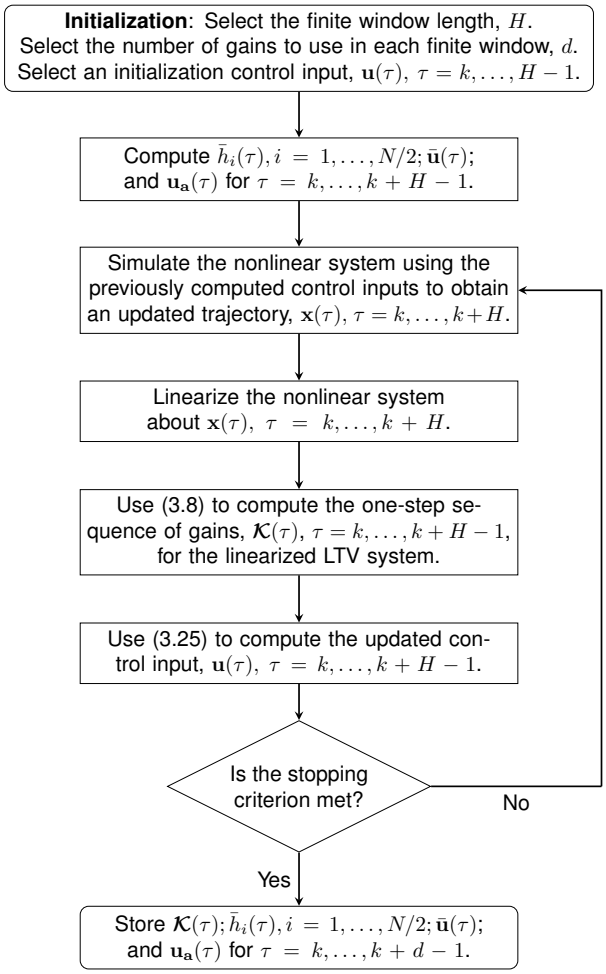


Figure 3.4: Flowchart of the iLQR scheme used for the gain computation for a finite window.

3.3.3 Simulation results

The network was simulated for $N = 40$ tanks and the values of its physical constants are presented in Table 3.1. The sampling time was set to $T_c = 1\,\mathrm{s}$ and the linearization period to $T_{lin} = 10T_c$. The initial level of the tanks is set to $h_i = 20\,\mathrm{cm}$ for $i = 1, ..., N$. The reference signal is given by

$$[\mathbf{r}(t)]_i = \begin{cases} 25 + 5\,\mathrm{sgn}\left(\sin\left(2\pi t/100\right)\right), & i = 4n+1 \\ 30 + 10\cos\left(t/35\right), & i = 4n+2 \\ 25 + 5\,\mathrm{sgn}\left(\sin\left(2\pi t/200\right)\right), & i = 4n+3 \\ 30 + 10\cos\left(t/50\right), & i = 4n+4 \end{cases}, \quad n \in \{0, ..., N/4-1\},$$

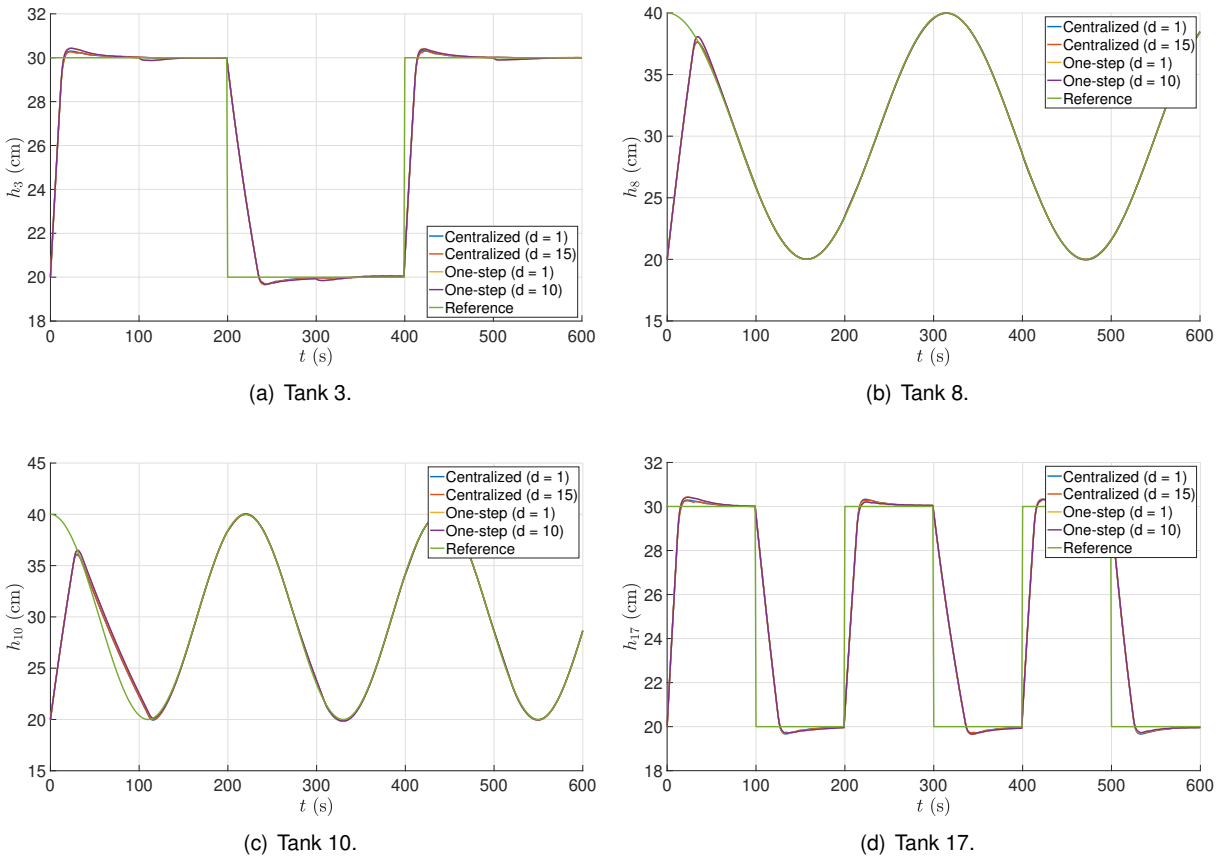
which is sampled with a periodicity of T_c.

The RHC scheme detailed in Section 2.2.2 is employed with a finite window length of $H = 30$, roughly twice the time constant of the slowest pole of the system, which is enough for the centralized and one-

Table 3.1: Values of the physical constants of the N tanks network.

Constant	Value
A_i, i odd	$28\ \mathrm{cm}^2$
A_i, i even	$32\ \mathrm{cm}^2$
a_i, i odd $\leq N/2$	$0.071\ \mathrm{cm}^2$
a_i, i even $\leq N/2$	$0.057\ \mathrm{cm}^2$
a_i, $i > N/2$	$0.040\ \mathrm{cm}^2$
g	$981\ \mathrm{cm\,s}^{-2}$
k_i	$3.33\ \mathrm{cm}^3\,\mathrm{s}^{-1}\,\mathrm{V}^{-1}$
γ_i, i odd	0.7
γ_i, i even	0.6
u^{sat}	$12\ \mathrm{V}$

step methods to approximate the infinite-horizon solution. The proposed method is compared with the analogous centralized solution, *i.e.*, using the unconstrained linear quadratic regulator (LQR) solution to compute the LQR feedback gains of (3.25). Due to the nonlinear dynamics of the network, the controller parameters were tuned empirically. The weighting matrices were set to $\mathbf{R}(k) = \mathbf{I}_{N/2}$, $\mathbf{Q} = 20\mathbf{I}_{N/2}$, and $\mathbf{Q}_I = 0.05\mathbf{I}_{N/2}$. The integral state saturation limits vector was selected as $[\mathbf{x_I}^{\mathrm{sat}}]_i = 10\ \mathrm{cm}$ for $i = 1, ..., N/2$, for both methods.



(a) Tank 3.

(b) Tank 8.

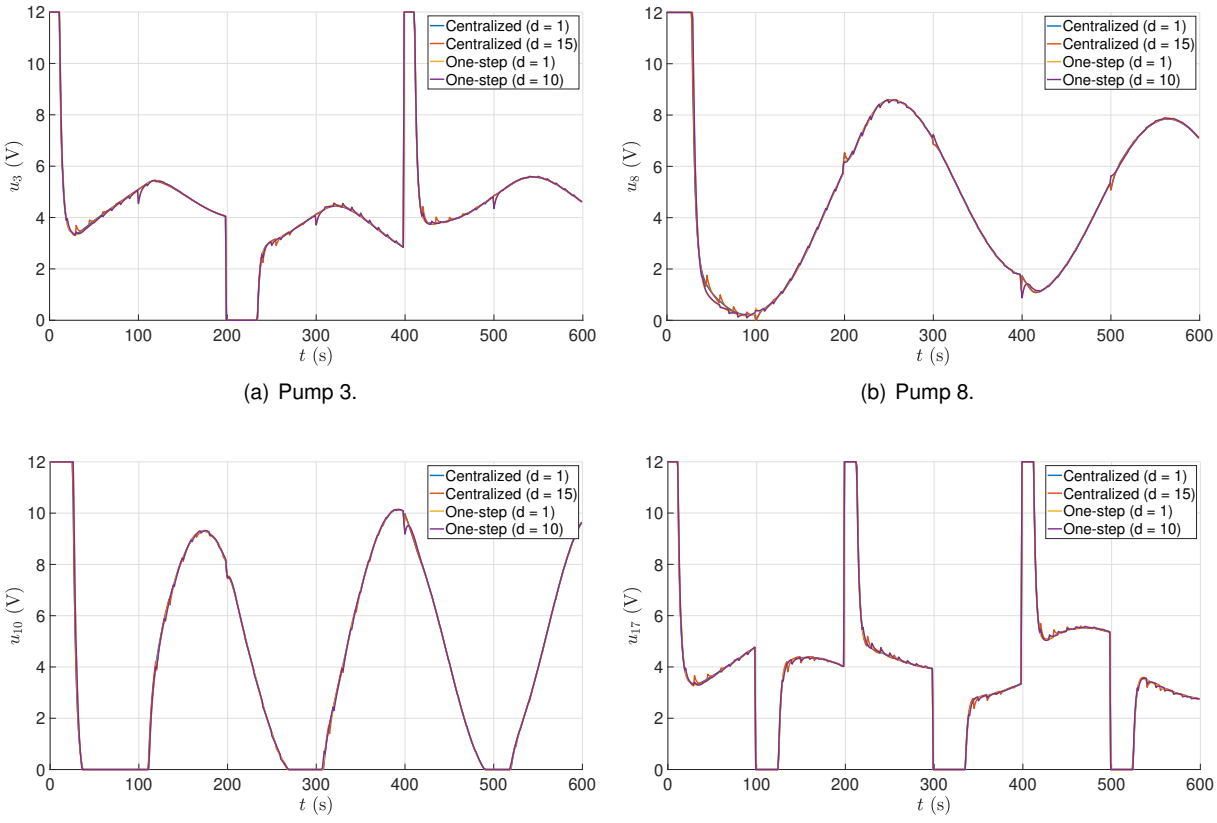
(c) Tank 10.

(d) Tank 17.

Figure 3.5: Evolution of the water levels of the network of N tanks.

Figure 3.5 depicts the nonlinear simulation of the water level of four tanks, one for each of the four types of reference signals, for the centralized and one-step methods. No sensor or process noise was

considered in this simulation. The simulation was also carried out making use of $d = 15$ and $d = 10$ of the gains computed for each window instead of just one, for the centralized and decentralized solutions, respectively. Figure 3.6 depicts the evolution of the input to the pumps associated with the tanks whose water level is plotted in Figure 3.5. First, it is clear that none of the solutions diverge and all drive the system output very close to the reference signal, yielding a very good performance. Note that the system is nonlinear and the control inputs are subject to hard constraints, none of which were taken into account in the design of the decentralized controller. Moreover, the variation of the water level in the tanks is considerable, which allows for a significant variation in the dynamics of the system throughout the simulation. Second, the fact that the proposed method for the computation of the decentralized feedback gains has a closed-form solution allows for its use in large-scale networks, as in this example, showing the scalability of the methods put forward. Third, it is interesting to notice that despite having considered a fully decentralized design, the performance obtained with the centralized solution is not considerably better. In fact, given that the use of a decentralized solution reduces tremendously the communication needs, particularly for a large-scale network as the case being considered, the use of a decentralized solution is very compelling. Fourth, regarding the proposed RHC scheme, it is clear that the use of more that one gain of each finite window does not introduce a noticeable loss of performance, yielding results very similar to those which make use of just the first gain. For this reason, it is possible to reduce the computational cost of this control approach dramatically, without any significant loss of performance.



(a) Pump 3.

(b) Pump 8.

(c) Pump 10.

(d) Pump 17.

Figure 3.6: Inputs of the pumps of the network of N tanks.

3.4 Experimental validation

In this section, the one-step convex relaxation approach is validated resorting to experimental results in a network of four tanks, which is a particularization of the N tanks network detailed in Section 3.3 for $N = 4$. The experimental setup is proposed and thoroughly analyzed in published work by the author [80]. That work aims to provide the community of control educators, practitioners, and researchers with an open-source low-cost experimental setup and dedicated interface, which is flexible and very easily reproducible. The setup can be adjusted to allow for several configurations for flexibility. A user-friendly dedicated MATLAB/Simulink interface with a personal computer is developed. It supports a seamless shift between a numeric simulation of the quadruple-tank process dynamics with processs and sensor noise and the interface with the physical experimental plant. All the materials and parts are inexpensive, can be readily obtained, and allow for a fast assembly. The computer-aided design (CAD) models of all the parts, the required technical drawings, the wiring schematics, the printed circuit board (PCB) design, the MATLAB/Simulink interface, assembly tutorials, and several examples are provided in the open-source repository available at `https://github.com/decenter2021/quadruple-tank-setup`. As a result, each setup is very easily reproducible, costs under 650€, and takes roughly 4 hours to assemble.

Four pumps are used in this setup: i) two for the actuation; and ii) two additional pumps that can be used as disturbance water flows that can be connected to any two of the four tanks. The scheme of the experimental quadruple-tank network with the addition of disturbance flows is depicted in Figure 3.7. The CAD model and physical implementation of the setup are presented in Figure 3.8. It consists of a 1 meter tall structure of 4 tanks supported by acrylic slabs. The water flows are directed using transparent flexible tubing. The water level is measured using one resistive water level sensor in each of the tanks. The valves, as well as several other components of the plant, are 3D printed. The setup is controlled via the circuit on the top slab of the plant with a dedicated PCB, which can be interfaced with MATLAB/Simulink via Universal Serial Bus (USB). The parameters of the tanks and pumps of the setup used for the experimental simulations are depicted in Tables 3.2 and 3.3, respectively.

Table 3.2: Parameters of the tanks of the quadruple-tank experimental setup.

Parameter	Tank 1	Tank 2	Tank 3	Tank 4
A	29.79 cm^2	29.79 cm^2	20.74 cm^2	20.49 cm^2
a	0.1241 cm^2	0.1350 cm^2	0.03327 cm^2	0.03310 cm^2

For the RHC algorithm, a finite window length of $H = 30$ and $d = 10$ are chosen. A minimum relative improvement of 10^{-4} is selected as the stopping criterion for the iLQR procedure. The weighting matrices were set to $\mathbf{R}(k) = \mathbf{I}_2$, $\mathbf{Q} = 20\mathbf{I}_2$, and $\mathbf{Q}_I = 0.05\mathbf{I}_2$. The integral state saturation limits vector was selected as $[\mathbf{x_I}^{\mathrm{sat}}]_1 = [\mathbf{x_I}^{\mathrm{sat}}]_2 = 10\ \mathrm{cm}$. Impulsive disturbance flows are felt at $t = 250$ s in tank 2 and at $t = 300$ s in tank 1. A constant disturbance flow is also felt starting at $t = 400$ s in tank 2 until the end of the simulation. The output, as well as the control action, are presented in Figure 3.9 for a numeric simulation and an experimental result. The evolution of the numeric and experimental results is

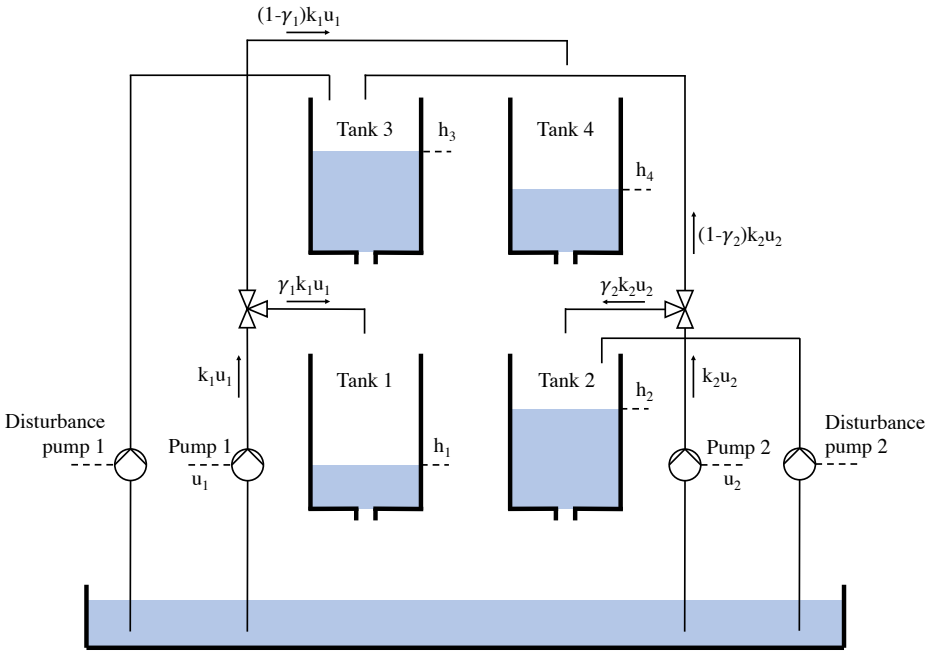
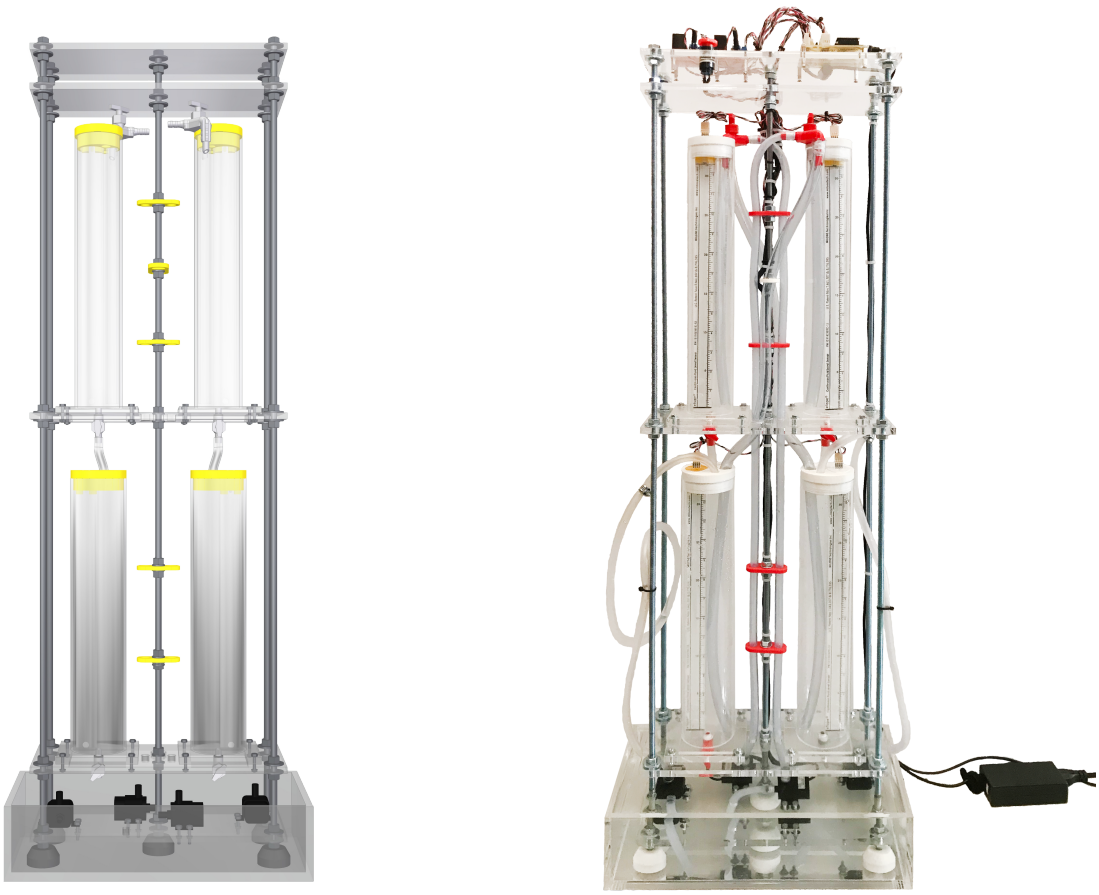


Figure 3.7: Schematic of the experimental quadruple-tank network with the addition of disturbance flows to tanks 2 and 3.



(a) CAD model.

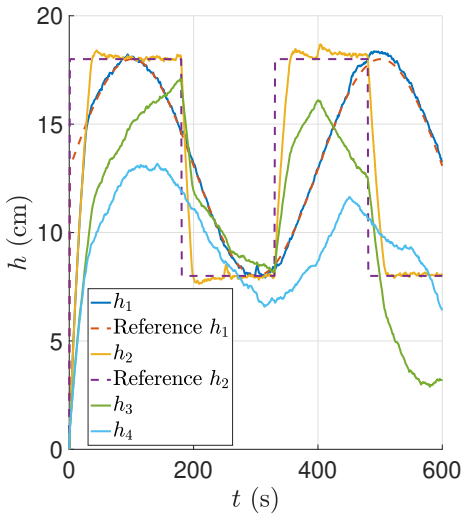
(b) Physical implementation.

Figure 3.8: Quadruple-tank process setup.

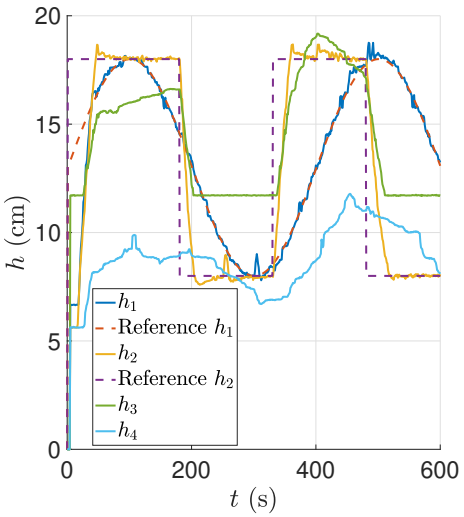
Table 3.3: Parameters of the pumps of the quadruple-tank experimental setup.

Parameter	Pump 1	Pump 2
$u^{\text{sat}-}$	0 V	0 V
$u^{\text{sat}+}$	12 V	12 V
c_1	39.01 cm^3s^{-1}	41.15 cm^3s^{-1}
c_2	-1.214 $\times 10^{-3}$	9.874 $\times 10^{-3}$
c_3	33.86 $\times 10^{-3}$	43.04 $\times 10^{-3}$
k	3.137 $\text{cm}^3\text{s}^{-1}\text{V}^{-1}$	3.314 $\text{cm}^3\text{s}^{-1}\text{V}^{-1}$
γ	0.7569	0.7563

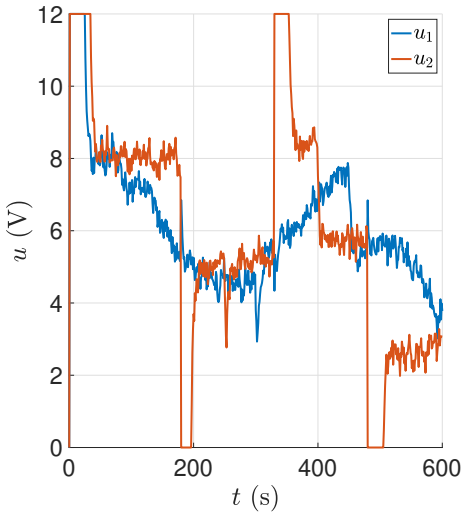
very similar except for tank 3, which suggests a poorer identification for this tank. It is also possible to see that the synthesized decentralized control law is able to successfully track a time-varying reference signal and to reject impulsive and constant disturbances.



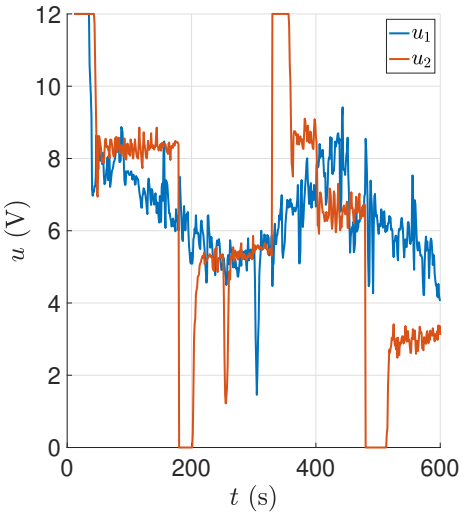
(a) Numerical results: evolution of the water level.



(b) Experimental results: evolution of the water level.



(c) Numerical results: evolution of the control action.



(d) Experimental results: evolution of the control action.

Figure 3.9: Evolution of the output and control action of the quadruple-tank process decentralized tracker for both numerical simulations and experimental results.

Chapter 4

Distributed and Decentralized RHC

In this chapter, the one-step relaxation proposed in Chapter 3 is leveraged to design a decentralized control solution that approximates the solution to the optimization problem (2.6) subject to the communication, computational, and memory restrictions in Constraints 1–4 put forward in Section 2.2.3, which are critical for a feasible implementation to very large-scale systems. In this chapter, the particular case of dynamically decoupled systems is considered. The derivation of the distributed and decentralized RHC solution follows two steps: i) decoupling of the real-time gain synthesis procedure, in Section 4.2; and ii) scheduling of the computations, in Section 4.3. In Section 4.4, the communication, computational, and memory requirements of the proposed solution are analyzed. This solution is presented, in a first instance, for a time-invariant network topology and, in Section 4.5, it is extended to a time-varying topology.

4.1 Preliminaries

Before proceeding with the derivation, it is convenient to make a few considerations to lighten the notation employed henceforth. Dynamically decoupled systems are considered, i.e., $(i,j) \in \mathcal{E}_d \implies i = j$, thus, $\mathbf{A} := \operatorname{diag}(\mathbf{A}_1, \ldots, \mathbf{A}_N)$ and $\mathbf{B} := \operatorname{diag}(\mathbf{B}_1, \ldots, \mathbf{B}_N)$, where $\mathbf{A}_i$ and $\mathbf{B}_i$ denote, for simplicity, matrices $\mathbf{A}_{i,i}$ and $\mathbf{B}_{i,i}$, respectively, for $i \in \{1, \ldots, N\}$. Furthermore, local state feedback via directed communication is allowed according to graph $\mathcal{G}_c$, which, for the sake of simplicity, is selected to be equal to the tracking output coupling graph $\mathcal{G}_o$. These graphs are henceforth denoted by $\mathcal{G} = \mathcal{G}_o = \mathcal{G}_c$.

4.2 Gain synthesis decoupling

Define a block decomposition of $\mathbf{P}(\tau)$ and $\mathbf{S}(\tau)$, whose blocks of indices (i,j) are denoted by $\mathbf{P}_{i,j}(\tau) \in \mathbb{R}^{n_i \times n_j}$ and $\mathbf{S}_{i,j}(\tau) \in \mathbb{R}^{m_i \times m_j}$, respectively. Making use of this block decomposition, one can rewrite (3.2) in a decoupled manner for each of the blocks of $\mathbf{P}(\tau)$ as a function of the local dynamics

and tracking output matrices as

$$\mathbf{P}_{p,q}(k+H) = \sum_{r \in \mathcal{D}_p^+ \cap \mathcal{D}_q^+} \mathbf{H}_{r,i}^T(k+H)\mathbf{Q}_r(k+H)\mathbf{H}_{r,j}(k+H)$$

and

$$\begin{aligned}\mathbf{P}_{p,q}(\tau) = & \sum_{r \in \mathcal{D}_p^+ \cap \mathcal{D}_q^+} \mathbf{H}_{r,i}^T(\tau)\mathbf{Q}_r(\tau)\mathbf{H}_{r,j}(\tau) + \sum_{r \in \mathcal{D}_p^+ \cap \mathcal{D}_q^+} \mathbf{K}_{r,i}^T(\tau)\mathbf{R}_r(\tau)\mathbf{K}_{r,j}(\tau) \\ & + \sum_{r \in \mathcal{D}_p^+} \sum_{s \in \mathcal{D}_q^+} \left(\mathbf{A}_p(\tau)\boldsymbol{\delta}_{pr} - \mathbf{B}_r(\tau)\mathbf{K}_{r,p}(\tau)\right)^T \mathbf{P}_{r,s}(\tau+1)\left(\mathbf{A}_q(\tau)\boldsymbol{\delta}_{qs} - \mathbf{B}_s(\tau)\mathbf{K}_{s,q}(\tau)\right).\end{aligned} \tag{4.1}$$

One can also express the blocks of $\mathbf{S}(\tau)$ as a function of the blocks of $\mathbf{P}(\tau+1)$ as

$$\mathbf{S}_{i,j}(\tau) = \mathbf{B}_i^T(\tau)\mathbf{P}_{i,j}(\tau+1)\mathbf{B}_j(\tau) + \boldsymbol{\delta}_{ij}\mathbf{R}_j(\tau),$$

which follows immediately from (3.3). Moreover, leveraging the aforementioned block decomposition, the relaxed conditions (3.7) of the feedback gains of the form $\mathbf{K}_{j,i}(\tau)$ can also be written in a decoupled manner, for each $i \in \{1, \ldots, N\}$, as

$$\begin{cases} \sum_{p \in \mathcal{D}_i^+} \mathbf{S}_{j,p}(\tau)\mathbf{K}_{p,i}(\tau) - \mathbf{B}_j^T(\tau)\mathbf{P}_{j,i}(\tau+1)\mathbf{A}_i(\tau) = \mathbf{0}, & j \in \mathcal{D}_i^+ \\ \mathbf{K}_{j,i}(\tau) = \mathbf{0}, & j \notin \mathcal{D}_i^+. \end{cases} \tag{4.2}$$

For each set $\mathcal{D}_i^+$, let $\mathcal{D}_i^+ = \left\{p_1^i, \ldots, p_{|\mathcal{D}_i^+|}^i\right\}$. Then, concatenating the feedback gains of the form $\mathbf{K}_{j,i}(\tau)$, with $j \in \mathcal{D}_i^+$, and combining the corresponding decoupled relaxed conditions of the first member of (4.2), it follows that

$$\tilde{\mathbf{K}}_i(\tau) = \tilde{\mathbf{S}}_i(\tau)^{-1}\tilde{\mathbf{P}}_i(\tau+1), \tag{4.3}$$

where

$$\tilde{\mathbf{K}}_i(\tau) := \begin{bmatrix} \mathbf{K}_{p_1^i,i} \\ \vdots \\ \mathbf{K}_{p_{|\mathcal{D}_i^+|}^i,i} \end{bmatrix},$$

$$\tilde{\mathbf{S}}_i(\tau) := \begin{bmatrix} \mathbf{S}_{p_1^i,p_1^i} & \cdots & \mathbf{S}_{p_1^i,p_{|\mathcal{D}_i^+|}^i} \\ \vdots & \ddots & \vdots \\ \mathbf{S}_{p_{|\mathcal{D}_i^+|}^i,p_1^i} & \cdots & \mathbf{S}_{p_{|\mathcal{D}_i^+|}^i,p_{|\mathcal{D}_i^+|}^i} \end{bmatrix}, \tag{4.4}$$

and

$$\tilde{\mathbf{P}}_i(\tau+1) := \begin{bmatrix} \mathbf{B}_{p_1^i}^T(\tau)\mathbf{P}_{p_1^i,i}(\tau+1)\mathbf{A}_i(\tau) \\ \vdots \\ \mathbf{B}_{p_{|\mathcal{D}_i^+|}^i}^T(\tau)\mathbf{P}_{p_{|\mathcal{D}_i^+|}^i,i}(\tau+1)\mathbf{A}_i(\tau) \end{bmatrix}. \tag{4.5}$$

Note that, although (4.3) provides an expression to compute the gains of systems $\mathcal{S}_j$ with $j \in \mathcal{D}_i^+$ in

relation to the state of system $\mathcal{S}_i$, it cannot be computed without full knowledge of $\mathbf{P}(\tau+1)$. This is due to the fact that the propagation of $\mathbf{P}_{j,i}(\tau)$ with $j \in \mathcal{D}_i^+$ in (4.1) depends on several $\mathbf{P}_{r,s}(\tau+1)$ with $r \in \mathcal{D}_j^+$ and $s \in \mathcal{D}_i^+$, which in turn depends on blocks of $\mathbf{P}(\tau+2)$ that are even less coupled with $\mathcal{S}_i$, and so on. For this reason, this result is not suitable for implementation in a distributed framework. In this section, under a reasonable approximation, the computation of these gains is decoupled. Then, an algorithm is put forward to allow for their distributed computation across the computational units of each system such that the communication, memory, and computational constraints presented in Section 2.2.3 are met.

Approximation 4.1. Consider $\mathbf{P}_{p,q}(\tau)$, with $p \in \mathcal{D}_i^+$ and $q \in \mathcal{D}_i^+$ for some i, and $\mathbf{P}_{r,s}(\tau+1)$, with $r \in \mathcal{D}_p^+$ and $s \in \mathcal{D}_q^+$. In the decentralized algorithm put forward in this work, $\mathbf{P}_{r,s}(\tau+1)$ is considered to be null in the computation of $\mathbf{P}_{p,q}(\tau)$ in the computational unit $\mathcal{T}_i$ if $(r,s) \notin \psi_i$, where

$$\psi_i = \bigcup_{j \in \mathcal{D}_i^+} \phi_j, \tag{4.6}$$

with

$$\phi_i := \mathcal{D}_i^+ \times \mathcal{D}_i^+ = \{(p,q) \in \mathbb{N}^2 : p \in \mathcal{D}_i^+ \wedge q \in \mathcal{D}_i^+\}. \tag{4.7}$$

The main result of this chapter is supported by Approximation 4.1. Next, it is argued that this approximation makes sense in the context of RHC of a large-scale network. Note that, from (3.4) in Theorem 3.1, $\mathbf{P}_{r,s}(\tau)$ is a measure of the contribution of the correlation between the states of systems $\mathcal{S}_r$ and $\mathcal{S}_s$ to the global cost. Consider Figure 4.1, which represents the topology of Approximation 4.1 in a graph. Intuitively, it is expected that the influence of $\mathbf{P}_{r,s}(\tau+1)$ is more dominant in the computation of $\mathbf{K}_{p,i}(k)$ for $p \in \mathcal{D}_i^+$ if both the states of $\mathcal{S}_r$ and $\mathcal{S}_s$ are coupled with the output of a system $\mathcal{S}_k$ that is coupled with the output of $\mathcal{S}_i$. Having this in mind, to decouple the gain synthesis of each local controller, each computational unit $\mathcal{T}_i$ keeps and updates each $\mathbf{P}_{p,q}(\tau)$ with $(p,q) \in \phi_i$. Henceforth, the approximation of matrix $\mathbf{P}_{p,q}(\tau)$ that stored and updated in $\mathcal{T}_i$ is denoted by $\mathbf{P}_{i,(p,q)}(\tau)$. Thus, making

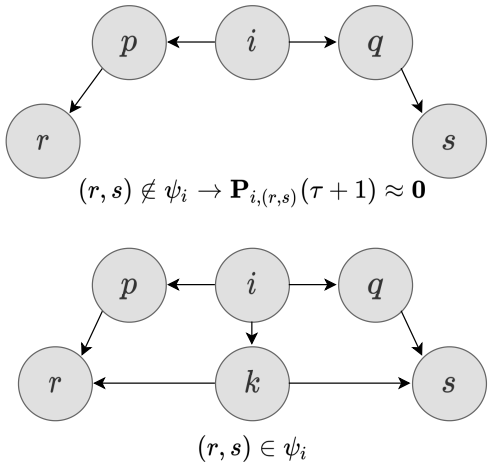


Figure 4.1: Graphic illustration of Approximation 4.1.

use of Approximation 4.1, (4.1) becomes

$$\begin{aligned}\mathbf{P}_{i,(p,q)}(\tau) = &\sum_{r\in\mathcal{D}_p^+\cap\mathcal{D}_q^+} \mathbf{H}_{r,i}^T(\tau)\mathbf{Q}_r(\tau)\mathbf{H}_{r,j}(\tau) + \sum_{r\in\mathcal{D}_p^+\cap\mathcal{D}_q^+} \mathbf{K}_{r,i}^T(\tau)\mathbf{R}_r(\tau)\mathbf{K}_{r,j}(\tau) \\ &+ \sum_{r\in\mathcal{D}_p^+} \sum_{\substack{s\in\mathcal{D}_q^+ \\ (r,s)\in\psi_i}} \left(\mathbf{A}_p(\tau)\boldsymbol{\delta}_{pr} - \mathbf{B}_r(\tau)\mathbf{K}_{r,p}(\tau)\right)^T \mathbf{P}_{\mathcal{D}_i^+,(r,s)}(\tau+1)\left(\mathbf{A}_q(\tau)\boldsymbol{\delta}_{qs} - \mathbf{B}_s(\tau)\mathbf{K}_{s,q}(\tau)\right),\end{aligned} \tag{4.8}$$

where the subscript $\mathcal{D}_i^+$ in $\mathbf{P}_{\mathcal{D}_i^+,(r,s)}(\tau+1)$ indicates, by abuse of notation, that $\mathbf{P}_{\mathcal{D}_i^+,(r,s)}(\tau+1)$ is computed in $\mathcal{T}_k$ with $k\in\mathcal{D}_i^+$. Note that, with Approximation 4.1, the propagation of $\mathbf{P}(\tau)$ in (4.8) can be computed in a distributed manner. It is important to remark that $\mathbf{P}_{\mathcal{D}_i^+,(r,s)}(\tau+1)$, inside the summation in (4.8), is not necessarily computed in $\mathcal{T}_i$, since only $\mathbf{P}_{i,(p,q)}(k|k)$, with $(p,q)\in\phi_i$, are updated in $\mathcal{T}_i$. Therefore, $\mathcal{S}_i$ has to receive $\mathbf{P}_{k,(r,s)}(\tau+1)$ through communication from a system $\mathcal{S}_k$, with $k\in\mathcal{D}_i^+$.

Lemma 4.1. *The approximation induced by Approximation 4.1 is exact if the topology of the network follows*

$$\forall i\in\{1,\ldots,N\}\ \forall p,q\in\mathcal{D}_i^+ \quad \mathcal{D}_p^+\times\mathcal{D}_q^+ \subseteq \bigcup_{j\in\mathcal{D}_i^+} \mathcal{D}_j^+\times\mathcal{D}_j^+.$$

Proof. Consider computational unit $\mathcal{T}_i$. Note that $\cup_{j\in\mathcal{D}_i^+}\mathcal{D}_j^+\times\mathcal{D}_j^+ = \cup_{j\in\mathcal{D}_i^+}\phi_j = \psi_i$, i.e., the set of systems pairs (r,s) whose matrix $\mathbf{P}_{r,s}(\tau+1)$ is known by the out-neighborhood of $\mathcal{T}_i$. According to (4.1), the exact propagation in $\mathcal{T}_i$ requires that, for every $p,q\in\mathcal{D}_i^+$, $\mathbf{P}_{r,s}(\tau+1)$ is accessible to $\mathcal{T}_i$, for every $r\in\mathcal{D}_p^+$ and $s\in\mathcal{D}_q^+$, i.e., $\mathbf{P}_{r,s}(\tau+1)$ with $(r,s)\in\mathcal{D}_p^+\times\mathcal{D}_q^+$. Thus, if these are contained in ψ_i for every system $\mathcal{S}_i$, then the decoupled propagation is exact. □

According to Lemma 4.1, the approximation induced by Approximation 4.1, which is used in the main result of this chapter, is exact for some network topologies. In these cases, it is possible to distribute the computations of the globally synthesized controller across the computational units without disregarding any components of $\mathbf{P}(\tau+1)$. Examples of these topologies are string, tree, and ring configurations.

After introducing Approximation 4.1, which allows for the decoupling of the gain synthesis procedure, it is possible to state the proposed RHC algorithm, as presented in the following result.

Theorem 4.1 (Distributed and decentralized RHC)**.** *The solution of the optimization problem* (2.6) *under the one-step convex relaxation* (3.7)*, under Approximation 4.1, and subject to the communication, computational, and memory constraints detailed in Section 2.2.3 is given, for a time-invariant coupling topology, by the local filter iterations presented in Algorithm 1 (see Remarks 4.1–4.3 for some details regarding its implementation).*

Proof. The local gain computation is performed according to (4.3). Under Approximation 4.1, the distributed propagation of $\mathbf{P}(\tau)$ is computed according to (4.8). The variables required by the gain computation and covariance propagation that are unknown to $\mathcal{T}_i$ can obtained via local communication with $\mathcal{T}_j$ for $j\in\mathcal{D}_i^+$. Finally, for considerations regarding the scheduling of the computations, see Section 4.3, and for a detailed analysis of the communication, computational, and memory requirements of this algorithm, see Section 4.4. □

Algorithm 1 Distributed and decentralized RHC algorithm for the local gain synthesis of a new window of gains at time instant k in computational unit $\mathcal{T}_i$, for a time-invariant coupling topology.

Output: $\mathbf{K}_{i,p}(\tau), \forall p \in \mathcal{D}_i^-, \tau = k, \ldots, k+d-1$
Step 1: **Predict**: $\mathbf{A}_i(\tau), \mathbf{B}_i(\tau), \mathbf{R}_i(\tau), \tau = k, \ldots, k+H-1$;
$\mathbf{H}_{i,p}(\tau), \forall p \in \mathcal{D}_i^-, \tau = k+1, \ldots, k+H$;
$\mathbf{Q}_i(\tau), \tau = k+1, \ldots, k+H$.
Step 2: **Transmit**: $\mathbf{Q}_i(k+H)^{1/2}\mathbf{H}_{i,p}(k+H), \forall p \in \mathcal{D}_i^-$ to $\forall p \in \mathcal{D}_i^- \setminus \{i\}$.
Step 3: **Receive**: $\mathbf{Q}_p(k+H)^{1/2}\mathbf{H}_{p,i}(k+H)$ from $\forall p \in \mathcal{D}_i^+ \setminus \{i\}$.
Step 4: **For**: $\tau = k+H-1, \ldots, k$
Step 4.1: **Transmit**: $\mathbf{R}_i(\tau), \mathbf{B}_i(\tau)$ to $\forall p \in \mathcal{D}_i^- \setminus \{i\}$;
$\mathbf{Q}_p(\tau+1)^{1/2}\mathbf{H}_{p,i}(\tau+1), \forall p \in \mathcal{D}_i^+$ to $\forall q \in \mathcal{D}_i^- \setminus \{i\}$;
If: $\tau \neq k$
$\mathbf{Q}_i(\tau)^{1/2}\mathbf{H}_{i,p}(\tau), \forall p \in \mathcal{D}_i^-$ to $\forall p \in \mathcal{D}_i^- \setminus \{i\}$;
End if
If: $\tau \neq k+H-1$
$\mathbf{R}_p(\tau+1), \mathbf{B}_p(\tau+1), \forall p \in \mathcal{D}_i^+$ to $\forall q \in \mathcal{D}_i^- \setminus \{i\}$;
$\mathbf{A}_i(\tau+1)$ to $\forall p \in \mathcal{D}_i^-$;
$\mathbf{K}_{p,i}(\tau+1), \forall p \in \mathcal{D}_i^+$ to $\forall q \in \mathcal{D}_i^- \setminus \{i\}$;
$\mathbf{P}_{i,(p,q)}(\tau+1)$ for some $(p,q) \in \phi_i$ to $\forall l \in \mathcal{D}_i^- \setminus \{i\}$.
End if
Step 4.2: **Receive**: $\mathbf{R}_p(\tau), \mathbf{B}_p(\tau)$ from $\forall p \in \mathcal{D}_i^+ \setminus \{i\}$;
$\mathbf{Q}_p(\tau+1)^{1/2}\mathbf{H}_{r,p}(\tau+1), \forall r \in \mathcal{D}_i^+$ from $\forall p \in \mathcal{D}_i^+ \setminus \{i\}$;
If: $\tau \neq k$
$\mathbf{Q}_p(\tau)^{1/2}\mathbf{H}_{p,i}(\tau), \forall p \in \mathcal{D}_i^+$ from $p \in \mathcal{D}_i^+ \setminus \{i\}$;
End if
If: $\tau \neq k+H-1$
$\mathbf{R}_r(\tau+1), \mathbf{B}_r(\tau+1), \forall r \in \mathcal{D}_p^+$ from $\forall p \in \mathcal{D}_i^+ \setminus \{i\}$;
$\mathbf{A}_p(\tau+1)$ from $p \in \mathcal{D}_i^-$;
$\mathbf{K}_{r,p}(\tau+1), \forall r \in \mathcal{D}_p^+$ from $\forall p \in \mathcal{D}_i^+ \setminus \{i\}$;
$\mathbf{P}_{p,(r,s)}(\tau+1)$ for some $(r,s) \in \phi_p$ from $\forall p \in \mathcal{D}_i^+ \setminus \{i\}$.
End if
Step 4.3: **Compute**:
If: $\tau = k+H-1$
$\mathbf{P}_{i,(p,q)}(\tau+1) \leftarrow \sum_{r \in \mathcal{D}_p^+ \cap \mathcal{D}_q^+} \mathbf{H}_{r,p}^T(\tau+1)\mathbf{Q}_r(\tau+1)\mathbf{H}_{r,q}(\tau+1), \forall (p,q) \in \phi_i$;
Else
Compute $\mathbf{P}_{i,(p,q)}(\tau+1) \forall (p,q) \in \phi_i$ making use of (4.8).
End if
Step 4.4: **Compute**:
$\mathbf{S}_{p,q}(\tau) \leftarrow \mathbf{B}_p^T(\tau)\mathbf{P}_{i,(p,q)}(\tau+1)\mathbf{B}_q(\tau) + \boldsymbol{\delta}_{pq}\mathbf{R}_q(\tau), \forall (p,q) \in \phi_i$;
Compute $\tilde{\mathbf{S}}_i(\tau)$ and $\tilde{\mathbf{P}}_i(\tau+1)$ making use of (4.4) and (4.5);
$\tilde{\mathbf{K}}_i(\tau) \leftarrow \tilde{\mathbf{S}}_i(\tau)^{-1}\tilde{\mathbf{P}}_i(\tau+1)$;
End for
Step 5: **Transmit**: $\mathbf{K}_{p,i}(\tau), \tau = k, \ldots, k+d-1$ to $\forall p \in \mathcal{D}_i^+ \setminus \{i\}$.
Step 6: **Receive**: $\mathbf{K}_{i,p}(\tau), \tau = k, \ldots, k+d-1$ from $\forall p \in \mathcal{D}_i^- \setminus \{i\}$.

Remark 4.1. Note that, in Algorithm 1, the data transmission and reception procedure is described generically. In fact, only a fraction of the data in the memory of a computational unit $\mathcal{T}_i$ has to be transmitted to a neighboring computational unit $\mathcal{T}_r$, with $r \in \mathcal{D}_i^-$. As an example, not all $\mathbf{P}_{i,(p,q)}(\tau+1), \forall (p,q) \in \phi_i$, are required for the computations performed in $\mathcal{T}_r$. Therefore, the data transmission protocol should be adapted to each application, depending on the network topology, to keep data transmissions to a minimum.

Remark 4.2. For a computational unit $\mathcal{T}_i$, in the computation in Step 4.3, there are matrices $\mathbf{P}_{\mathcal{D}_i^+,(r,s)}(\tau+1)$ which have to be retrieved either from memory, if $(r,s) \in \phi_i$, or from another computational unit $\mathcal{T}_j$, with $j \in \mathcal{D}_i^+$, in Step 3. Oftentimes, $\mathbf{P}_{\mathcal{D}_i^+,(r,s)}(\tau)$ is available from more than one source, so a selection rule should be put in place. There are several alternatives for this rule, which depend significantly on the application. In what follows a few examples are given. First, one can reduce the communication cost by prioritizing content in the memory of $\mathcal{T}_i$ and, if it is not available, retrieve it from $\mathcal{T}_j$ with which communication is cheaper. Second, one can prevent delays by choosing the content which can be retrieved the fastest. Third, one can associate, in each computational unit $\mathcal{T}_i$, a scalar loss $L_{i,(p,q)} \in \mathbb{R}$ to the computation of $\mathbf{P}_{i,(p,q)}(\tau), \forall (p,q) \in \phi_i$, that is an empirical metric of its accuracy. An example of an empirical metric is the number of times that Approximation 4.1 was used in the computation of $\mathbf{P}_{i,(p,q)}(\tau)$, i.e., the number of terms of the summation in (4.1) that were disregarded, which is given by

$$L_{i,(p,q)} = \left| \left(\mathcal{D}_p^+ \times \mathcal{D}_q^+ \right) \setminus \psi_i \right|. \tag{4.9}$$

On one hand, for a time-invariant topology, the empirical losses of this example are constant with time and, thus, a computational unit may retrieve only $\mathbf{P}_{\mathcal{D}_i^+,(r,s)}(\tau+1)$ from the computational unit $\mathcal{T}_k$, with $k \in \mathcal{D}_i^+$, which is known to have the lowest $L_{k,(r,s)}$, thereby reducing communication pressure. On the other hand, if the topology is time-varying, this metric has to be updated at each time instant in Step 4.3 of Algorithm 1. In general, in Step 4.2 all the available $\mathbf{P}_{\mathcal{D}_i^+,(r,s)}(\tau+1)$ are retrieved with their respective losses, and either their weighted average or the average of the ones with the minimum loss is considered.

Remark 4.3. Note that the computations performed in Step 4.3 of Algorithm 1 do not guarantee the positive definiteness of the resulting matrices, which may lead to numerical problems. To prevent it, the computations in these steps should be written in matrix form as follows. Let $\mathcal{D}_p^+ = \{r_1^p, \ldots, r_{|\mathcal{D}_p^+|}^p\}$ and consider $\tilde{\mathbf{W}}_p(\tau)$ of dimension $n_p \times \sum_{r \in \mathcal{D}_p^+} n_r$ and $\tilde{\mathbf{P}}_{\mathcal{D}_i^+}^{(p,q)}(\tau+1)$ of dimension $\sum_{r \in \mathcal{D}_p^+} n_r \times \sum_{s \in \mathcal{D}_q^+} n_s$, which are defined below. Consider the decomposition of $\tilde{\mathbf{W}}_p(\tau)$ as a block row, whose block of index j is denoted by $\left[\tilde{\mathbf{W}}_p(\tau)\right]_{(1,j)} \in \mathbb{R}^{n_p \times n_{r_j^p}}$, and the block matrix decomposition of $\tilde{\mathbf{P}}_{\mathcal{D}_i^+}^{(p,q)}(\tau+1)$, whose block of indices (i,j) is denoted by $\left[\tilde{\mathbf{P}}_{\mathcal{D}_i^+}^{(p,q)}(\tau+1)\right]_{(i,j)} \in \mathbb{R}^{n_{r_i} \times n_{r_j^q}}$. If one defines

$$\left[\tilde{\mathbf{W}}_p(\tau)\right]_{(1,j)} := \mathbf{A}_p(\tau)\boldsymbol{\delta}_{pr_j^p} - \mathbf{B}_{r_j^p}(\tau)\mathbf{K}_{r_j^p,p}(\tau)$$

and

$$\left[\tilde{\mathbf{P}}^{(p,q)}_{\mathcal{D}_i^+}(\tau+1)\right]_{(i,j)} := \begin{cases} \mathbf{P}_{\mathcal{D}_i^+,(r_i^p,r_j^q)}(\tau+1), & (r_i^p, r_j^q) \in \psi_i \\ \mathbf{0}, & (r_i^p, r_j^q) \notin \psi_i \end{cases}$$

then, (4.8) may be rewritten as

$$\begin{aligned} \mathbf{P}_{i,(p,q)}(\tau) = & \sum_{r \in \mathcal{D}_p^+ \cap \mathcal{D}_q^+} \mathbf{H}_{r,i}^T(\tau)\mathbf{Q}_r(\tau)\mathbf{H}_{r,j}(\tau) + \sum_{r \in \mathcal{D}_p^+ \cap \mathcal{D}_q^+} \mathbf{K}_{r,i}^T(\tau)\mathbf{R}_r(\tau)\mathbf{K}_{r,j}(\tau) \\ & + \tilde{\mathbf{W}}_p(k)\tilde{\mathbf{P}}^{(p,q)}_{\mathcal{D}_i^+}(\tau+1)\tilde{\mathbf{W}}_q^T(k). \end{aligned} \tag{4.10}$$

The propagation step in matrix form in (4.10) is more robust to numerical error than (4.8). Additionally, in networks with few communication links, numerical problems can be prevented by ensuring the positive definiteness of $\tilde{\mathbf{P}}^{(p,q)}_{\mathcal{D}_i^+}(\tau+1)$ with $p=q$ before performing the computation in (4.10). This should be done by: i) performing an eigendecomposition of $\tilde{\mathbf{P}}^{(p,q)}_{\mathcal{D}_i^+}(\tau+1)$; ii) forcing the negative eigenvalues to a positive value, for instance the lowest positive eigenvalue; and iii) recomputing $\tilde{\mathbf{P}}^{(p,q)}_{\mathcal{D}_i^+}(\tau+1)$ from the inverse of the decomposition. This procedure maintains the directions of the contribution of the state correlations to the global cost encoded in $\mathbf{P}(\tau+1)$.

4.3 Scheduling

In Section 4.2, under appropriate approximations, an algorithm was devised for the decoupled synthesis of the RHC gains over a finite-window $\{k, \ldots, k+H-1\}$. This algorithm allows for distributing the global computation across the computational units of the systems that make up the network. Nevertheless, recall that, as put forward in Section 2.2.2, for the application of this framework to the infinite-horizon problem, a new window of gains of length H has to be computed every d time steps, of which only d gains are used to compute the control input according to (2.3). Furthermore, the decoupled gain computation in Algorithm 1 is carried out backward in time. Thus, at the time instant that corresponds to the beginning of each window, all RHC gains over that window must have already been computed. Since these computations involve several communication instances, the steps in Algorithm 1 have to be properly scheduled to make use of soft real-time data transmissions only, as required by the communication constraints for a feasible large-scale implementation detailed in Section 2.2.3. This issue is addressed in this section.

Denote the control discretization time by T_c, which is the sampling time of the LTV system (2.1). Let T_t denote the interval of time between allowed communication instances, i.e., T_t must be set so that it is greater than the time it takes for a system to communicate with the systems in its neighborhood and to perform floating-point operations with the data received in that transmission. Generally, T_c is significantly larger than the minimum achievable T_t. It is important to remark that, in general, parallel to the control computations, there are also estimation algorithms running over the network. These usually require a higher rate of communication than control algorithms. Thus, one can set T_t to the sampling time of the estimation algorithm.

Algorithm 1 requires $H+2$ data transmissions between systems in the neighborhood of each other. Only after the last transmission of Algorithm 1, the gains $\mathbf{K}_{i,p}(\tau), \forall p \in \mathcal{D}_i^-, \tau = k, \ldots, k+d-1$ are available to $\mathcal{T}_i$. To compute the control feedback according to (2.3), these gains need to be available in $\mathcal{S}_i$ at $t = kT_c$. Therefore, to avoid hard real-time transmissions, they must be sent at most at $t = kT_c - T_t$. Thus, the proposed approach is to schedule the $H+2$ transmissions of Algorithm 1 such that they are performed at a rate of $1/T_t$. Thus, it takes $\Delta^- := (H+2)T_t$ to run Algorithm 1. For that reason, the computation of the RHC gains over the window $\{k, \ldots, k+H-1\}$ must start at $t = kT_c - \Delta^-$. A scheme of the proposed scheduling of Algorithm 1 over a timeline is depicted in Figure 4.2, for the illustrative time interval $[kT_c; (k+d)T_c[$. In this time interval, the control input is computed according to (2.3), making use of the gains in the window $\{k, \ldots, k+H-1\}$, which is represented in the line adjacent to the timeline that comprises $kT_c \leq t < (k+d)T_c$. As aforementioned, the computation of these gains must be carried out starting at $t = kT_c - \Delta^-$ and they only become available for use in the control law at $t = kT_c$, which is represented by the left rectangle with diagonal fill in the scheme. Moreover, during the application of the control inputs in $kT_c \leq t < (k+d)T_c$, the computation of the gains corresponding to the next finite window needs to start being carried out, which is represented in the scheme by the rectangle with diagonal fill in the right.

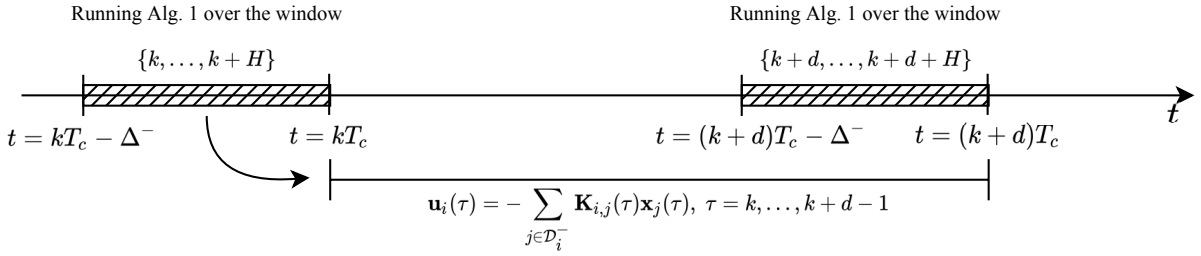


Figure 4.2: Illustration of scheduling of Algorithm 1 over a timeline.

First, it is worth remarking that the proposed scheduling requires that the dynamics of a system $\mathcal{S}_i$ over the window $\{k, \ldots, k+H-1\}$ are predicted at $t = kT_c - \Delta^-$. Thus, if Δ^- is too large, the quality of the prediction may be degraded. Second, if $dT_c < \Delta^-$, i.e., $d/(H+2) < T_t/T_c$, the gain computation of two consecutive windows overlaps (rectangles with diagonal fill in Figure 4.2 overlap). If T_t is not large enough to handle twice the communication and computational pressure, then it may lead to the unfeasibility of the control solution.

4.4 Communication, computational, and memory requirements

In this section, the communication, computational, and memory requirements of Algorithm 1 are analyzed and it is verified that the Constraints 1–4 imposed in Section 2.2.3 are followed. To allow for a clearer analysis and lighter notation, in this section, the complexity is written for an homogeneous network, i.e., a network of identical systems. In particular, systems with the same order n_1, output dimension o_1, dimension of in-neighborhood ν_1^-, and dimension of out-neighborhood ν_1^+ are considered.

First, making use of the scheduling of the computations in Algorithm 1 proposed in Section 4.3, only soft real-time transmissions are required for the distributed computation of the controller gains. Second, in Section 4.1, the graph $\mathcal{G}_c$ was defined to allow for local state feedback via directed communication according to the topology of the tracking output couplings, which is characterized by $\mathcal{G}_o$. It follows from Algorithm 1 that one directed transmission on the opposite direction of the links in $\mathcal{G}_c$ is also required in the synthesis procedure of each window in Steps 5 and 6. Thus, communication between systems is restricted to the tracking output coupling graph with undirected edges. In this configuration, system $\mathcal{S}_i$ can only receive information from system $\mathcal{S}_j$ if the tracking output of $\mathcal{S}_j$ is coupled with the state of $\mathcal{S}_i$ or the converse, i.e., $j \in \mathcal{D}_i^+ \cup \mathcal{D}_i^-$. In fact, system $\mathcal{S}_i$ requires the exchange of data through communication with at most $\max\left(\nu_i^-, \nu_i^+\right) - 1$ systems in each iteration of the filter. Thus, the communication complexity of system $\mathcal{S}_i$ is $\mathcal{O}\left(\max\left(\nu_i^-, \nu_i^+\right)\right)$. Given that neither ν_i^-, the number of systems whose state is coupled with the output of $\mathcal{S}_i$, nor ν_i^+, the number of systems whose tracking output is coupled with the state of $\mathcal{S}_i$, increase with the size of the network, then the communication complexity of each system grows with $\mathcal{O}(1)$ with the dimension of the network.

Third, according to Algorithm 1, each system $\mathcal{S}_i$ has to store in memory: i) the matrices that model the dynamics of $\mathcal{S}_i$ over the window $\{k, \ldots, k+H-1\}$; ii) the tracking output and control input weighting matrices of $\mathcal{S}_i$; iii) $\mathbf{P}_{i,(p,q)}(\tau+1)$, with $(p,q) \in \phi_i$; and iv) the sequences of controller gains $\mathbf{K}_{i,p}(\tau), \forall p \in \mathcal{D}_i^-, \tau = k, \ldots, k+d-1$ and $\mathbf{K}_{p,i}(\tau), \forall p \in \mathcal{D}_i^+, \tau = k, \ldots, k+H-1$. The data storage complexity of these is: i) $\mathcal{O}\left(\max\left(n_1, m_1, \nu_1^- o_1\right) nH\right)$; ii) $\mathcal{O}\left(\max\left(m_1^2, o_1^2\right) H\right)$; iii) $\mathcal{O}((\nu_1^+)^2 n_1^2)$; and iv) $\mathcal{O}(m_1 n_1 \max(\nu_1^+, \nu_1^-)H)$, respectively. Of course there are auxiliary variables, which are required to be handled at each iteration, that require temporary memory storage, but these do not have a greater memory footprint than the aforementioned variables. Thus, the data storage complexity of each system grows with $\mathcal{O}(1)$ with the dimension of the network, since the memory required in each computational unit does not scale with the number of systems in the network.

Fourth, as far as the computational complexity is concerned, the most intensive computations of Algorithm 1 are the propagation of $\mathbf{P}_{i,(p,q)}(\tau)$, with $(p,q) \in \phi_i$, in Step 4.3. These computations require $\mathcal{O}((\nu_1^-)^4 n_1^3)$ floating-point operations in total, which grows with $\mathcal{O}(1)$ with the dimension of the network.

It is possible to conclude that the algorithm put forward in this thesis follows the constraints presented in Section 2.2.3 to be suitable to be implemented in very large-scale networks.

4.5 Extension to time-varying coupling topologies

Oftentimes, the tracking output couplings between systems vary with time due to: i) the failure of systems of the network; ii) the introduction of new systems in the network; or iii) switching tracking configurations. In this section, the distributed and decentralized control solution is extended to allow for a time-varying tracking output coupling topology. To that purpose, consider, now, a time-varying directed graph $\mathcal{G}(k)$, with time-varying in-degree $\nu_i^-(k)$, in-neighborhood $\mathcal{D}_i^-(k)$, out-degree $\nu_i^+(k)$, out-neighborhood $\mathcal{D}_i^+(k)$, and define $\phi_i(k)$ and $\psi_i(k)$ analogously to (4.7) and (4.6), respectively, for system $\mathcal{S}_i$.

The extension of Algorithm 1 to a time-varying tracking output coupling topology is given by Algorithm 2. Notice that a similar outline is followed. It is important to remark that, in Algorithm 2, $\mathbf{P}_{i,(p,q)}(\tau+1)$ is only computed between systems $\mathcal{S}_p$ and $\mathcal{S}_q$ whose states are both coupled with the output of $\mathcal{S}_i$ in the previous time instant, i.e., $(p,q) \in \phi_i(\tau)$. This allows to lessen the communication burden, since the computations are more efficiently distributed across all computational units.

Even though the extension to a time-varying topology is quite straightforward as far as distributing the gain synthesis across of the systems is concerned, that is not the case for the scheduling of the computations over time. Recall the scheduling procedure proposed in Section 4.3. It is possible to point out that the fact that the computation of the RHC gains over the window $\{k, \ldots, k+H-1\}$ must start at $t = kT_c - \Delta^-$, requires that at $t = kT_c - \Delta^-$ a system $\mathcal{S}_i$ receives data through communication from $\mathcal{S}_p$, with $p \in \mathcal{D}_i^+(k+H)$. That is, at $t = kT_c - \Delta^-$ communication between $\mathcal{S}_i$ and a system with which $\mathcal{S}_i$ is coupled at $t = (k+H)T_c$ is required. In most applications, if two systems are coupled at a given time instant, communication between them is probably feasible at that time instant. However, at a given time instant, communication between two systems that are coupled an interval of time $HT_c + \Delta^-$ later may not be feasible due to the changing spatial configuration of the network over time. For example, consider the problem of maintaining a formation of unnamed aerial vehicles (UAVs). At a given time instant, two UAVs may be separated by an obstacle. If a tracking output coupling is expected to be established between them in the future and if H is required to be large, then communication between them may be required when they are still separated by the obstacle, which may be difficult to achieve. It is important to point out that the effect of this aspect varies greatly with the application in question. On one hand, in applications with slowly time-varying coupling topologies and spatial configurations, the communication requirements needed to implement the scheduling procedure proposed in Section 4.3 are likely feasible. On the other hand, in applications that, in the time frame of the receding horizon window, there are significant spatial configuration changes that impede communication links between systems, then some additional considerations should be taken into account.

Consider the in-neighborhood and out-neighborhood $\mathcal{D}_i^-(\tau)$ and $\mathcal{D}_i^+(\tau)$, for $\tau = k, \ldots, k+H$, that would be obtained if no restrictions on the establishment of the communication links existed. Define $\mathcal{C}_i(\tau)$ as the set of systems with which system $\mathcal{S}_i$ can establish a communication link, in an undirected sense, at $t = kT_c - (\tau - k + 2)T_t$, for $\tau = k, \ldots, k+H$. Then, consider instead an in-neighborhood and out-neighborhood restricted to the set of systems with which communication is feasible, i.e.,

$$\tilde{\mathcal{D}}_i^{\pm}(\tau) := \mathcal{D}_i^{\pm}(\tau) \cap \mathcal{C}_i(\tau), \tag{4.11}$$

for $\tau = k, \ldots, k+H$. Making use of these restricted neighborhoods, the communication requirements in Algorithm 2 are feasible. Note that $\tilde{\phi}(\tau)$ is defined analogously. It is important to point out that to compute the first d gains of the window, which are actually used to compute the control input with (2.3), no restrictions can be enforced, i.e., $\mathcal{D}_i^{\pm}(\tau) \subseteq \mathcal{C}_i(\tau)$. Therefore, d should be sufficiently small to allow for that. In Chapter 5, this scheme is applied to the mega-constellation orbit control problem, for which the communication links between satellites are heavily restricted by the Earth's curvature.

Algorithm 2 One-step RHC algorithm for the local gain synthesis of a new window of gains at time instant k in computational unit $\mathcal{T}_i$ for a time-varying coupling topology.

Output: $\mathbf{K}_{i,p}(\tau), \forall p \in \tilde{\mathcal{D}}_i^-(\tau), \tau = k, \ldots, k+d-1$

Step 1: **Predict**: $\tilde{\mathcal{D}}_i^-(\tau), \tilde{\mathcal{D}}_i^-(\tau), \tau = k, \ldots, k+H$, according to (4.11);
$\mathbf{A}_i(\tau), \mathbf{B}_i(\tau), \mathbf{R}_i(\tau), \tau = k, \ldots, k+H-1$;
$\mathbf{H}_{i,p}(\tau), \forall p \in \tilde{\mathcal{D}}_i^-(\tau), \tau = k+1, \ldots, k+H$;
$\mathbf{Q}_i(\tau), \tau = k+1, \ldots, k+H$.

Step 2: **Transmit**: $\mathbf{Q}_i(k+H)^{1/2}\mathbf{H}_{i,p}(k+H), \forall p \in \tilde{\mathcal{D}}_i^-(k+H)$ to $\forall p \in \tilde{\mathcal{D}}_i^-(k+H) \setminus \{i\}$.

Step 3: **Receive**: $\mathbf{Q}_p(k+H)^{1/2}\mathbf{H}_{p,i}(k+H)$ from $\forall p \in \tilde{\mathcal{D}}_i^+(k+H) \setminus \{i\}$.

Step 4: **For**: $\tau = k+H-1, \ldots, k$

Step 4.1: **Transmit**: $\mathbf{R}_i(\tau), \mathbf{B}_i(\tau)$ to $\forall p \in \tilde{\mathcal{D}}_i^-(\tau) \setminus \{i\}$;
$\mathbf{Q}_p(\tau+1)^{1/2}\mathbf{H}_{p,i}(\tau+1), \forall p \in \tilde{\mathcal{D}}_i^+(\tau+1)$ to $\forall q \in \tilde{\mathcal{D}}_i^-(\tau) \setminus \{i\}$;
If: $\tau \neq k$
$\mathbf{Q}_i(\tau)^{1/2}\mathbf{H}_{i,p}(\tau), \forall p \in \tilde{\mathcal{D}}_i^-(\tau)$ to $\forall p \in \tilde{\mathcal{D}}_i^-(\tau) \setminus \{i\}$;
End if
If: $\tau \neq k+H-1$
$\mathbf{R}_p(\tau+1), \mathbf{B}_p(\tau+1), \forall p \in \tilde{\mathcal{D}}_i^+(\tau+1)$ to $\forall q \in \tilde{\mathcal{D}}_i^-(\tau) \setminus \{i\}$;
$\mathbf{A}_i(\tau+1)$ to $\forall p \in \tilde{\mathcal{D}}_i^-(\tau)$;
$\mathbf{K}_{p,i}(\tau+1), \forall p \in \tilde{\mathcal{D}}_i^+(\tau+1)$ to $\forall q \in \tilde{\mathcal{D}}_i^-(\tau) \setminus \{i\}$;
$\mathbf{P}_{i,(p,q)}(\tau+1)$ for some $(p,q) \in \phi_i(\tau+1)$ to $\forall l \in \tilde{\mathcal{D}}_i^-(\tau) \setminus \{i\}$.
End if

Step 4.2: **Receive**: $\mathbf{R}_p(\tau), \mathbf{B}_p(\tau)$ from $\forall p \in \tilde{\mathcal{D}}_i^+(\tau) \setminus \{i\}$;
$\mathbf{Q}_p(\tau+1)^{1/2}\mathbf{H}_{r,p}(\tau+1), \forall r \in \tilde{\mathcal{D}}_i^+(\tau+1)$ from $\forall p \in \tilde{\mathcal{D}}_i^+(\tau) \setminus \{i\}$;
If: $\tau \neq k$
$\mathbf{Q}_p(\tau)^{1/2}\mathbf{H}_{p,i}(\tau), \forall p \in \tilde{\mathcal{D}}_i^+(\tau)$ from $p \in \tilde{\mathcal{D}}_i^+(\tau) \setminus \{i\}$;
End if
If: $\tau \neq k+H-1$
$\mathbf{R}_r(\tau+1), \mathbf{B}_r(\tau+1), \forall r \in \tilde{\mathcal{D}}_p^+(\tau+1)$ from $\forall p \in \tilde{\mathcal{D}}_i^+(\tau) \setminus \{i\}$;
$\mathbf{A}_p(\tau+1)$ from $p \in \tilde{\mathcal{D}}_i^-(\tau)$;
$\mathbf{K}_{r,p}(\tau+1), \forall r \in \tilde{\mathcal{D}}_p^+(\tau+1)$ from $\forall p \in \tilde{\mathcal{D}}_i^+(\tau) \setminus \{i\}$;
$\mathbf{P}_{p,(r,s)}(\tau+1)$ for some $(r,s) \in \tilde{\phi}_p(\tau+1)$ from $\forall p \in \tilde{\mathcal{D}}_i^+(\tau) \setminus \{i\}$.
End if

Step 4.3: **Compute**:
If: $\tau = k+H-1$
$$\mathbf{P}_{i,(p,q)}(\tau+1) \leftarrow \sum_{r \in \tilde{\mathcal{D}}_p^+(\tau+1) \cap \tilde{\mathcal{D}}_q^+(\tau+1)} \mathbf{H}_{r,p}^T(\tau+1)\mathbf{Q}_r(\tau+1)\mathbf{H}_{r,q}(\tau+1), \forall (p,q) \in \tilde{\phi}_i(\tau);$$
Else
$$\mathbf{P}_{i,(p,q)}(\tau+1) \leftarrow \sum_{r \in \tilde{\mathcal{D}}_p^+(\tau+1) \cap \tilde{\mathcal{D}}_q^+(\tau+1)} \big(\mathbf{H}_{r,i}^T(\tau+1)\mathbf{Q}_r(\tau+1)\mathbf{H}_{r,j}(\tau+1) +$$
$$\mathbf{K}_{r,i}^T(\tau+1)\mathbf{R}_r(\tau+1)\mathbf{K}_{r,j}(\tau+1)\big) + \tilde{\mathbf{W}}_p(k+1)\tilde{\mathbf{P}}^{(p,q)}_{\tilde{\mathcal{D}}_i^+(\tau+1)}(\tau+2)\tilde{\mathbf{W}}_q^T(k), \forall (p,q) \in \tilde{\phi}_i(\tau).$$
End if

Step 4.4: **Compute**:
$\mathbf{S}_{p,q}(\tau) \leftarrow \mathbf{B}_p^T(\tau)\mathbf{P}_{i,(p,q)}(\tau+1)\mathbf{B}_q(\tau) + \boldsymbol{\delta}_{pq}\mathbf{R}_p(\tau), \forall (p,q) \in \tilde{\phi}_i(\tau)$;
Compute $\tilde{\mathbf{S}}_i(\tau)$ and $\tilde{\mathbf{P}}_i(\tau+1)$ making use of (4.4) and (4.5);
$\tilde{\mathbf{K}}_i(\tau) \leftarrow \tilde{\mathbf{S}}_i(\tau)^{-1}\tilde{\mathbf{P}}_i(\tau+1)$;

End for

Step 5: **Transmit**: $\mathbf{K}_{p,i}(\tau), \tau = k, \ldots, k+d-1$ to $\forall p \in \mathcal{D}_i^+(\tau) \setminus \{i\}$.

Step 6: **Receive**: $\mathbf{K}_{i,p}(\tau), \tau = k, \ldots, k+d-1$ from $\forall p \in \mathcal{D}_i^-(\tau) \setminus \{i\}$.

Chapter 5

Application to on-board orbit control of LEO mega-constellations

In this chapter, the distributed decentralized RHC algorithm developed in Chapter 4 is applied to the cooperative orbit control problem of LEO mega-constellations. The scheme presented in this chapter is novel and it is developed aiming for efficiency and fuel saving in a distributed and decentralized framework. Note that, unlike the state-of-the-art decentralized control algorithms, it follows the communication, computational, and memory constraints presented in Section 2.2.3, which are necessary for an effective real-time implementation in very large-scale networks such as a mega-constellation. All source code of the implementation of the proposed algorithm to this problem is available as an example in the DECENTER Toolbox at `https://decenter2021.github.io/examples/DDRHCStarlink`.

5.1 Mega-constellation model

Consider a constellation with a total of T satellites. The satellites are evenly distributed over P orbital planes at a nominal inclination $\bar{i}$ and with a nominal relative phasing between adjacent planes of $\bar{\beta} = 2\pi F/T$, where F is the phasing parameter. Such a configuration is designated as a Walker constellation and it is denoted by $\bar{i} : T/P/F$. The nominal orbits are circular and have a semi-major axis of $\bar{a}$. This constellation can be modeled as a network of coupled systems, $\mathcal{S}_j$, each associated with a computational unit $\mathcal{T}_j$, with $j = 1, ..., T$. Let $\mathbf{p}_i \in \mathbb{R}^3$ and $\mathbf{v}_i \in \mathbb{R}^3$ denote the position and velocity vectors, respectively, of $\mathcal{S}_i$ expressed in the J2000 Earth centered inertial (ECI) frame. The dynamics of each satellite of the constellation are modeled independently, since there is no dynamical coupling between them. Each satellite $\mathcal{S}_i$ is equipped with Hall effect thrusters, aligned according to the local TNW frame (x axis along the velocity vector, z axis along the orbit's angular momentum vector, and y axis completes the right-handed coordinate system) that generate a force $\mathbf{u}_i \in \mathbb{R}^3$ expressed in the TNW local frame. Each thruster has a maximum thrust, C_{t1}. The model of the dynamics of a single satellite $\mathcal{S}_i$ is, thus,

given by

$$\begin{cases} \dot{\mathbf{p}}_i = \mathbf{v}_i \\ \dot{\mathbf{v}}_i = -\mu \mathbf{p}_i / ||\mathbf{p}_i||_2^3 + \mathbf{a}_i^{J_2} + \mathbf{a}_i^P + \mathbf{R}_{TNW}^{ECI} \mathbf{u}_i / m_i \\ \dot{m}_i = -||\mathbf{u}_i||_1 / (I_i^{sp} g_0), \end{cases} \tag{5.1}$$

where m_i denotes the mass of the satellite, μ denotes the gravitational parameter of the Earth, $\mathbf{a}_i^{J_2} \in \mathbb{R}^3$ and $\mathbf{a}_i^P \in \mathbb{R}^3$ denote the perturbation accelerations of the effect of J_2 and all other perturbations, respectively, $\mathbf{R}_{TNW}^{ECI}$ denotes the rotation matrix from the TNW local frame to the J2000 ECI frame, I_i^{sp} denotes the specific impulse of the Hall effect thrusters of $\mathcal{S}_i$, and g_0 denotes the standard gravity acceleration.

In this application, for control law synthesis purposes, the parameterization of the orbits of each satellite of the constellation is achieved by the set of non-singular mean orbital elements for near-circular inclined orbits $(a, u, e_x, e_y, i, \Omega)$, respectively semi-major axis, mean argument of latitude, two eccentricity vector components, inclination, and longitude of ascending node. These can be related with the more common Keplerian set $(a, e, i, \Omega, \omega, M)$, which has a singularity for circular orbits, with

$$\begin{cases} u = M + \omega \\ e_x = e \cos \omega \\ e_y = e \sin \omega, \end{cases}$$

where e denotes the eccentricity, ω denotes the argument of perigee, and M denotes the mean anomaly. Denote the state of a satellite $\mathcal{S}_i$, made up of the aforementioned six non-singular mean orbital elements, by

$$\mathbf{x}_i(t) = \left[a_i(t) \; u_i(t) \; e_{xi}(t) \; e_{yi}(t) \; i_i(t) \; \Omega_i(t) \right]^T .$$

The satellite orbital mechanics (5.1) are nonlinear and, thus, have to be linearized to employ the distributed and decentralized method put forward in this work. The linearization of the dynamics of each satellite is carried out about a nominal orbit. These are defined such that the set of nominal orbits of all satellites makes up a consistent nominal constellation, in the sense that the nominal separations: i) along-track; ii) inter-plane; and iii) in relative phasing between adjacent planes are enforced. It is very important to remark that this nominal constellation is used for linearization purposes only – it is not employed for bounding-box tracking of each individual satellite at any point. The necessity of enforcing constellation-wide constraints in the definition of each nominal orbit is made clear in the formulation of the orbit control problem as a RHC problem in Section 5.2.

The nominal state of $\mathcal{S}_i$ at time instant t, $\bar{\mathbf{x}}_i(t) = \left[\bar{a}_i(t)\ \bar{u}_i(t)\ \bar{e}_{xi}(t)\ \bar{e}_{yi}(t)\ \bar{i}_i(t)\ \bar{\Omega}_i(t)\right]^T$, is defined by

$$\begin{cases} \bar{a}_i(t) = \bar{a} \\ \bar{u}_i(t) = \bar{u}_{t_0} + ((i-1) \bmod T/P)\, 2\pi P/T + \lfloor (i-1)P/T \rfloor 2\pi F/T + (\dot{M} + \dot{\omega})(t - t_0) \\ \bar{e}_{x,i}(t) = 0 \\ \bar{e}_{y,i}(t) = 0 \\ \bar{i}_i(t) = \bar{i} \\ \bar{\Omega}_i(t) = \bar{\Omega}_{t_0} + \lfloor (i-1)P/T \rfloor 2\pi/P + \dot{\Omega}(t - t_0). \end{cases} \tag{5.2}$$

Above, $\dot{M}$, $\dot{\omega}$, and $\dot{\Omega}$ are the secular rates, including the effect of J_2, on the mean anomaly, argument of perigee, and longitude of ascending node, respectively, which are given by [93, Chapter 8]

$$\dot{M} = \frac{3\bar{n}R_{\oplus}^2 J_2}{4\bar{a}^2}\left(3\sin^2\bar{i} - 2\right),$$

$$\dot{\omega} = \frac{3\bar{n}R_{\oplus}^2 J_2}{4\bar{a}^2}\left(4 - 5\sin^2\bar{i}\right),$$

and

$$\dot{\Omega} = -\frac{3\bar{n}R_{\oplus}^2 J_2}{2\bar{a}^2}\cos\bar{i},$$

for a nominal orbit, where $\bar{n} := \sqrt{\mu/\bar{a}^3}$, and $R_{\oplus}$ denotes the Earth's mean equatorial radius. Note that the nominal orbits of all satellites in (5.2) depend on three constellation-wise parameters $(t_0, \bar{u}_{t_0}, \bar{\Omega}_{t_0})$, whose physical meaning is that the nominal orbit of $\mathcal{S}_1$ has mean argument of latitude $\bar{u}_{t_0}$ and longitude of ascending node $\bar{\Omega}_{t_0}$ at time instant t_0. These three parameters are designed herein as the anchor of the nominal constellation.

There are a few aspects worth pointing out regarding the anchor of the nominal constellation. First, all satellites must agree on an anchor for the nominal constellation at any time instant. Second, to minimize linearization errors, the anchor should be selected such that the nominal position of each satellite is as close as possible to their actual position. Thus, since the position of the satellites drifts away from their nominal position with time, due to neglected secular effects, other perturbations, and maneuvers, the anchor must be updated from time to time. Third, note that the evolution of the nominal states takes the effect of the Earth's oblateness into account, which significantly decreases the frequency with which the anchor has to be updated. Fourth, the computation of the anchor for a time instant t_0 should be performed in accordance with an optimization problem of the form

$$\underset{\bar{u}_{t_0}, \bar{\Omega}_{t_0}}{\text{minimize}} \quad \sum_{i=1}^{T} \left(\alpha\left(u_i(t_0) - \bar{u}_i(t_0)\right) + \alpha\left(\Omega_i(t_0) - \bar{\Omega}_i(t_0)\right)\right),$$

where $\alpha : \mathbb{R} \to \mathbb{R}$ is a convex function. Given that $\bar{u}_i(t_0)$ does not depend on $\bar{\Omega}_{t_0}$ and $\bar{\Omega}_i(t_0)$ does not depend on $\bar{u}_{t_0}$, the optimization above can be decoupled into two problems: one corresponding to $\bar{u}_{t_0}$ and other to $\bar{\Omega}_{t_0}$. Fifth, for the sake of simplicity, in this application, $\alpha(\cdot) = (\cdot)^2$ is chosen, which leads to

the closed-form solution

$$\begin{cases} \bar{u}_{t_0} = \sum_{i=1}^{T} \left(u_i(t_0) - ((i-1) \bmod T/P)\, 2\pi P/T - \lfloor (i-1)P/T \rfloor 2\pi F/T \right) \\ \bar{\Omega}_{t_0} = \sum_{i=1}^{T} \left(\Omega_i(t_0) - \lfloor (i-1)P/T \rfloor 2\pi/P \right). \end{cases} \tag{5.3}$$

However, to improve the robustness to outliers, the ℓ_1 norm, i.e., $\alpha(\cdot) = |\cdot|$, or Huber loss function could be used instead, still leading to a convex optimization problem. Sixth, even though the optimization problem above cannot be easily decoupled to be distributed across the computational units of the satellites in the network, it is not a serious issue, since it is only required to be solved sporadically. Thus, it can either be: i) computed in a centralized node and then the solution broadcast to the network; or ii) solved distributively over a period of time making use of distributed gradient methods with asymptotic consensus guarantees [94]. It is worth pointing out that a new anchor $(t_0, \bar{u}_{t_0}, \bar{\Omega}_{t_0})$ can be used starting at $t = t_0 + \Delta$, where Δ can be as large as necessary to allow for communication and centralized or cooperative computations.

The evolution of the state of $\mathcal{S}_i$ is linearized about the aforementioned nominal orbits, defining a relative position $\delta\mathbf{x}_i(t)$ based the set of orbital elements $\delta\mathbf{x}_i(t) := [a_i(t)\ \delta u_i(t)\ \delta e_{x,i}(t)\ \delta e_{y,i}(t)\ \delta i_i(t)\ \delta\Omega_i(t)]$, introduced in [78], which is defined as

$$\delta\mathbf{x}_i(t) := \begin{bmatrix} a_i(t)/\bar{a}_i(t) - 1 \\ u_i(t) - \bar{u}_i(t) + \left(\Omega_i(t) - \bar{\Omega}_i(t)\right) \cos \bar{i}_i(t) \\ e_{x,i}(t) - \bar{e}_{x,i}(t) \\ e_{y,i}(t) - \bar{e}_{y,i}(t) \\ i_i(t) - \bar{i}_i(t) \\ \left(\Omega_i(t) - \bar{\Omega}_i(t)\right) \sin \bar{i}_i(t) \end{bmatrix}. \tag{5.4}$$

This set parameterizes the position of the satellite, $\mathbf{x}_i(t)$, in relation to its nominal position, $\bar{\mathbf{x}}_i(t)$. In [95–97], the dynamics (5.1), taking the effect of J_2 into account but neglecting the remaining perturbations, are linearized about near-circular nominal orbits. Making use of Floquet theory, system transition and convolution matrices are derived to write the discrete-time LTV system

$$\delta\mathbf{x}_i((k+1)T_c) = \mathbf{A}_i(k)\delta\mathbf{x}_i(kT_c) + \mathbf{B}_i(k)\mathbf{u}_i(kT_c)/m_i(kT_c) \tag{5.5}$$

with a sampling time T_c and assuming that $\mathbf{u}_i(t)$ and $m_i(t)$ remain constant over each interval $[kT_c; (k+1)T_c[$. Given that electric propulsion is employed, which has very reduced propellant mass rates, the constant mass approximation is very reasonable (as an example, the Hall effect thrusters considered in the illustrative simulations in the sequel reach a mass rate of the order of 10^{-6} Kg/s at full throttle). Henceforth, to alleviate the notation, the continuous time instant $t = kT_c$ is denoted by the discrete-time index k. As an example, $\delta\mathbf{x}_i(kT_c)$ and $\mathbf{u}_i(kT_c)$ are denoted by $\delta\mathbf{x}_i(k)$ and $\mathbf{u}_i(k)$, respectively. For circular nominal orbits and following the notation herein, the state transition matrix $\mathbf{A}_i(k)$ is given, according to

[97], by

$$\mathbf{A}_i(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -\bar{\Lambda}T_c & 1 & 0 & 0 & -4\bar{K}T_c \sin 2\bar{i} & 0 \\ 0 & 0 & \cos\Delta\omega & -\sin\Delta\omega & 0 & 0 \\ 0 & 0 & \sin\Delta\omega & \cos\Delta\omega & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{7}{2}\bar{K}T_c \sin 2\bar{i} & 0 & 0 & 0 & 2\bar{K}T_c \sin^2\bar{i} & 1 \end{bmatrix}$$

and the convolution matrix $\mathbf{B}_i(k)$ by

$$\mathbf{B}_i(k) = \begin{bmatrix} \frac{2\Delta u}{\bar{n}\bar{a}\bar{W}} & 0 & 0 \\ -\frac{\bar{\Lambda}\Delta u^2}{\bar{n}\bar{a}\bar{W}^2} & \frac{2\Delta u}{\bar{n}\bar{a}\bar{W}} & \Psi_{2,3} \\ 2\Psi_{4,2} & \Psi_{3,2} & 0 \\ -2\Psi_{3,2} & \Psi_{4,2} & 0 \\ 0 & 0 & \Psi_{5,3} \\ \frac{7}{2}\frac{\bar{K}\Delta u^2 \sin 2\bar{i}}{\bar{n}\bar{a}\bar{W}^2} & 0 & \Psi_{6,3} \end{bmatrix}, \tag{5.6}$$

where

$$\Psi_{2,3} := \frac{4\bar{K}\sin 2\bar{i}}{\bar{n}\bar{a}\bar{W}^2}\left(\cos(\bar{u}_i(kT_c+T_c)) - \cos(\bar{u}_i(kT_c)) + \sin(\bar{u}_i(kT_c))\Delta u\right),$$

$$\Psi_{3,2} := \frac{\cos(\bar{u}_i(kT_c+T_c)) - \cos(\bar{u}_i(kT_c) + \bar{C}\Delta u)}{\bar{n}\bar{a}(1-\bar{C})\bar{W}},$$

$$\Psi_{4,2} := \frac{\sin(\bar{u}_i(kT_c+T_c)) - \sin(\bar{u}_i(kT_c) + \bar{C}\Delta u)}{\bar{n}\bar{a}(1-\bar{C})\bar{W}},$$

$$\Psi_{5,3} := \frac{\sin(\bar{u}_i(kT_c+T_c)) - \sin(\bar{u}_i(kT_c))}{\bar{n}\bar{a}\bar{W}},$$

$$\Psi_{6,3} := -\frac{\bar{W}+2\bar{K}\sin^2\bar{i}}{\bar{n}\bar{a}\bar{W}^2}(\cos(\bar{u}_i(kT_c+T_c)) - \cos(\bar{u}_i(kT_c))) - 2\bar{K}\sin^2\bar{i}\sin(\bar{u}_i(kT_c))\Delta u/\left(\bar{n}\bar{a}\bar{W}^2\right),$$

$$\Delta u := \bar{W}T_c,$$

$$\Delta\omega := \bar{K}(5\cos^2\bar{i} - 1)T_c,$$

$$\bar{K} := (3/4)\bar{n}R_{\oplus}^2 J_2/\bar{a}^2,$$

$$\bar{\Lambda} := (3/2)\bar{n} + (7/2)\bar{K}(3\cos^2\bar{i} - 1),$$

$$\bar{W} := \bar{n} + \bar{K}(8\cos^2\bar{i} - 2), \quad \text{and}$$

$$\bar{C} := \bar{K}(5\cos^2\bar{i} - 1)/\bar{W}.$$

Note that $\mathbf{A}_i(k)$ is time-invariant and equal for all satellites, whereas $\mathbf{B}_i(k)$ is time-varying and depends on the known nominal evolution of the mean argument of latitude. Thus, these matrices can be easily predicted in $\mathcal{T}_i$ over a window of future time instants.

5.2 Controller implementation

Now that the constellation has been modeled as an LTV system, the constellation orbit control problem has to be formulated as a RHC problem. In this work, in an attempt to reduce fuel consumption and to follow the communication, computational, and memory constraints detailed in Section 2.2.3, a control scheme is devised such that the satellites control their position relative to each other. On one hand, the semi-major axis, eccentricity, and inclination of the orbit of each satellite may be controlled in a decoupled fashion, thus an inertial tracking output component given by

$$\mathbf{z}_{i,in}(k) = \begin{bmatrix} a_i(k) - \bar{a}_i(k) \\ e_{x,i}(k) - \bar{e}_{x,i}(k) \\ e_{y,i}(k) - \bar{e}_{y,i}(k) \\ i_i(k) - \bar{i}_i(k) \end{bmatrix} = \begin{bmatrix} \bar{a}_i(k)\delta a_i(k) \\ \delta e_{x,i}(k) \\ \delta e_{y,i}(k) \\ \delta i_i(k) \end{bmatrix}$$

is considered for each satellite $\mathcal{S}_i$, which is not coupled with any other satellites. Note that driving $\mathbf{z}_{i,in}(k)$ to zero is equivalent to driving the semi-major axis, eccentricity, and inclination to their nominal values. On the other hand, to maintain the shape of the constellation, $\delta u_i(k)$ and $\delta\Omega_i(k)$ ought to be controlled in relation to other satellites. To achieve a distributed solution, each satellite ought to be coupled with only a small number of satellites, which does not scale with the number of satellites in the constellation. Furthermore, the satellites with which $\mathcal{S}_i$ is coupled should be in its proximity, which is more convenient to establish communication links. Thus, it is considered that two satellites are coupled if they are within a tracking range R of each other, i.e., $||\mathbf{p}_i - \mathbf{p}_j|| \leq R$, up to a maximum of $|\mathcal{D}^-|_{\max}$ satellites in $\mathcal{D}_i^-$. If more than $|\mathcal{D}^-|_{\max} - 1$ satellites other than $\mathcal{S}_i$ are within a tracking range of $\mathcal{S}_i$, only the $|\mathcal{D}^-|_{\max} - 1$ closest are considered. Since the nominal evolution of the constellation is known, it is easy to predict the coupling topology over a window of future time instants. Let $\mathcal{D}_i^- \setminus \{i\} = \left\{j_1^i, \ldots, j_{|\mathcal{D}_i^-|-1}^i\right\}$. Then the relative tracking output component is given by

$$\mathbf{z}_{i,rel}(k) := \operatorname{col}\left(\mathbf{z}_{i,j_1^i}^{ref}(k), \ldots, \mathbf{z}_{i,j_{|\mathcal{D}_i^-|-1}^i}^{ref}(k)\right)$$

with

$$\begin{aligned} \mathbf{z}_{i,j}^{ref}(k) &:= \begin{bmatrix} u_i(k) - u_j(k) - (\bar{u}_i(k) - \bar{u}_j(k)) \\ \Omega_i(k) - \Omega_j(k) - \left(\bar{\Omega}_i(k) - \bar{\Omega}_j(k)\right) \end{bmatrix} \\ &= \begin{bmatrix} \delta u_i(k) - \delta u_j(k) - (\delta\Omega_i(k) - \delta\Omega_j(k)) / \tan\bar{i} \\ (\delta\Omega_i(k) - \delta\Omega_j(k)) / \sin\bar{i} \end{bmatrix}. \end{aligned} \tag{5.7}$$

Thus, if $\mathcal{G}(k)$ contains a directed spanning tree, driving $\mathbf{z}_{i,j}^{ref}(k)$ to zero maintains the shape of the constellation. It is interesting to point out that the definition of the relative tracking output (5.7) makes use of the nominal constellation just to retrieve the nominal angular spacing in u and Ω between $\mathcal{S}_i$ and $\mathcal{S}_j$. Thus, the actual position of the satellite is inevitably going to slowly drift away from the nominal constellation,

while maintaining the desired shape. Defining the tracking output of $\mathcal{S}_i$ as $\mathbf{z}_i(k) := \mathrm{col}(\mathbf{z}_{i,rel}(k), \mathbf{z}_{i,in}(k))$, it can be written as

$$\mathbf{z}_i(k) = \sum_{p \in \mathcal{D}_i^-} \mathbf{H}_{i,p}(k) \delta \mathbf{x}_p(k) \tag{5.8}$$

with

$$\mathbf{H}_{i,p}(k) := \begin{cases} \begin{bmatrix} \mathbf{1}_{|\mathcal{D}_i^-| \times 1} \otimes \mathbf{H}_{rel} \\ \mathbf{H}_{in} \end{bmatrix}, & p = i \\ -\begin{bmatrix} \mathbf{l}_k \otimes \mathbf{H}_{rel} \\ \mathbf{0}_{4 \times 6} \end{bmatrix}, & p = j_{i,k}, \end{cases}$$

where

$$\mathbf{H}_{in} := \begin{bmatrix} \bar{a} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

and

$$\mathbf{H}_{rel} := \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & -1/\tan \bar{i} \\ 0 & 0 & 0 & 0 & 0 & 1/\sin \bar{i} \end{bmatrix}.$$

The tracking output weighting matrices $\mathbf{Q}_i(k)$ are of the form

$$\mathbf{Q}_i(k) := \mathrm{diag}\left(\mathbf{I}_{|\mathcal{D}_i^-|-1} \otimes \mathbf{Q}_i^{rel}(k), \mathbf{Q}_i^{in}(k) \right),$$

where $\mathbf{Q}_i^{rel}(k) \in \mathbb{R}^{2\times 2}$, $\mathbf{Q}_i^{in}(k) \in \mathbb{R}^{4\times 4}$, and $\mathbf{R}_i(k) \in \mathbb{R}^{3\times 3}$ are defined in the sequel for the illustrative constellation under study.

The evolution of the state of each satellite state is modeled by the LTV system (5.5) and the constellation orbit control problem is formulated as the regulation of the tracking output (5.8). Thus, we are in the conditions of applying the distributed and decentralized RHC method put forward in Chapter 4. The tracking output coupling graph $\mathcal{G}(k)$ is time-varying, thus one has to follow Algorithm 2. A selection rule based on the time-varying empirical loss analogous to (4.9) is employed, as detailed in Remark 4.2. To avoid numerical problems, the procedures detailed in Remark 4.3 are employed. Considering the convolution matrix as defined in (5.6), one obtains the feedback law

$$\mathbf{u}_i(k) = -m_i(k) \sum_{j \in \mathcal{D}_i^-(k)} \mathbf{K}_{i,j}(k) \delta \mathbf{x}_j(k).$$

However, more often than not, the relative mean orbital elements are not readily available to each satellite from on-board sensors or filters. It is more usual for each satellite to have access to its position and velocity in a Cartesian coordinate system, which can be obtained making use of a global navigation satellite system (GNSS) receiver, for instance. Although it is easy to compute the relative mean orbital elements from the mean orbital elements and nominal orbital elements according to (5.4), it is not

straightforward to obtain the mean orbital elements from the Cartesian position and velocity. In fact, if ones applies the Keplerian orbit transformation directly to the Cartesian position and velocity, one obtains osculating orbital elements, which contain very significant short-period oscillations due to the Earth's uneven gravity field and other perturbations. For instance, in the conditions of the illustrative simulations in the sequel, these oscillations lead to differences between osculating and mean orbital elements that reach 6 km in the semi-major axis. If one neglected these differences, the controller feedback would attempt to counteract these natural oscillations, thus wasting fuel and degrading tracking performance. To aim for meter-level tracking accuracy and to reduce fuel consumption, a transformation inspired in the one proposed in [98] is employed to account for the Earth's uneven gravity field making use of the spherical harmonic expansion up to a desired degree, which is the main source of these oscillations. An open-source MATLAB implementation and thorough documentation of this transformation is made available in the *osculating2mean* toolbox at `https://github.com/decenter2021/osculating2mean`, which is based on [99–101] and [102]. In the simulation results in the sequel, the Earth's gravity field EGM96 spherical harmonic expansion [103] up to degree and order 12 is employed.

5.3 Illustrative mega-constellation and tuning

The illustrative mega-constellation of a single shell, inspired in the first shell of the Starlink constellation, was chosen to assess the performance of the method devised in this thesis. In this section, the illustrative mega-constellation is described and the parameters of the control solution are tuned. The constellation is a Walker $53.0\,\mathrm{deg} : 1584/72/17$. The phasing parameter of this Starlink shell is not publicly available, so it was chosen according to [104] such that the minimum distance between satellites is maximized. All satellites are assumed to be identical. The parameters that characterize the illustrative constellation are presented in Table 5.1.

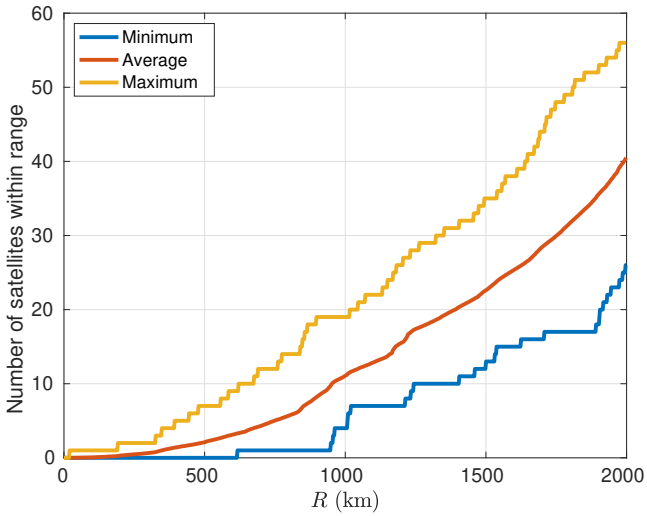


Figure 5.1: Number of satellites within ISL range at 0 TDB seconds since J2000.

The evolution of the minimum, maximum, and average number of satellites within tracking output

Table 5.1: Parameters of the constellation.

Configuration	
Inclination ($\bar{i}$)	$53.0\,\mathrm{deg}$
Number of satellites (T)	1584
Number of orbital planes (P)	72
Phasing parameter (F)	17
Semi-major axis ($\bar{a}$)	$6921.0\,\mathrm{km}$
Eccentricity ($\bar{e}$)	0
Satellites	
Initial mass	$260\,\mathrm{kg}$
Drag coefficient (C_D)	2.2
Section area (A)	$24.0\,\mathrm{m}^2$
Solar radiation pressure coefficient (C_R)	1.2
Solar radiation pressure area (SRPA)	$10.0\,\mathrm{m}^2$
Electric thrusters	
Maximum thrust (C_{t1})	$0.068\,\mathrm{N}$
Specific impulse (I_{sp})	$1640.0\,\mathrm{s}$

coupling range of each satellite of the constellation as a function of R, at 0 Dynamical Barycentric Time (TDB) seconds since J2000, is depicted in Figure 5.1. A tracking output coupling range of $R = 750$ km is considered. It allows for every satellite of the constellation to share a tracking output coupling with, at least, another satellite at any time. The maximum in-neighborhood cardinality is set to $|\mathcal{D}^-|_{\mathsf{max}} = 6$, which allows for each satellite to establish tracking output couplings with up to 5 other satellites. In Figure 5.2 a snapshot of the projection of the position of each satellite of the constellation over the Earth's surface, as well as the tracking output couplings, at 0 TDB seconds since J2000 is shown. An animation of the evolution of the ground track of the constellation and of the tracking output couplings can be viewed in the website of the DECENTER Toolbox. It is interesting to note that, due to the higher density of satellites at extreme latitudes, much more couplings are established. This fact allows for more accurate control in these regions, which is desirable to avoid collisions.

The control discretization time is set to $T_c = 10\,\mathrm{s}$, which is small enough to achieve a good approximation of the continuous-time nonlinear dynamics model and large enough such that the control input update frequency is attainable by the Hall effect thrusters. The tracking output coupling topology varies greatly with time, thus the parameters H and d of the scheduling of the RHC distributed and decentralized algorithm have to be tuned thoughtfully, according to the limitations and communication requirements pointed out in Section 4.5. The interval of time between allowed communications is set to $T_t = 1\,\mathrm{s}$, which is the period of GNSS signals and the sampling time of a filter relying on them. Due to the curvature of the Earth and the low altitude of this LEO shell, the satellites quickly lose line-of-sight between each other. Therefore, at a given time instant, communication between two satellites that are coupled an interval of time later is not necessarily feasible. The approximate theoretical line-of-sight range such that the ISL do not enter the atmosphere any lower than the Thermosphere is given by

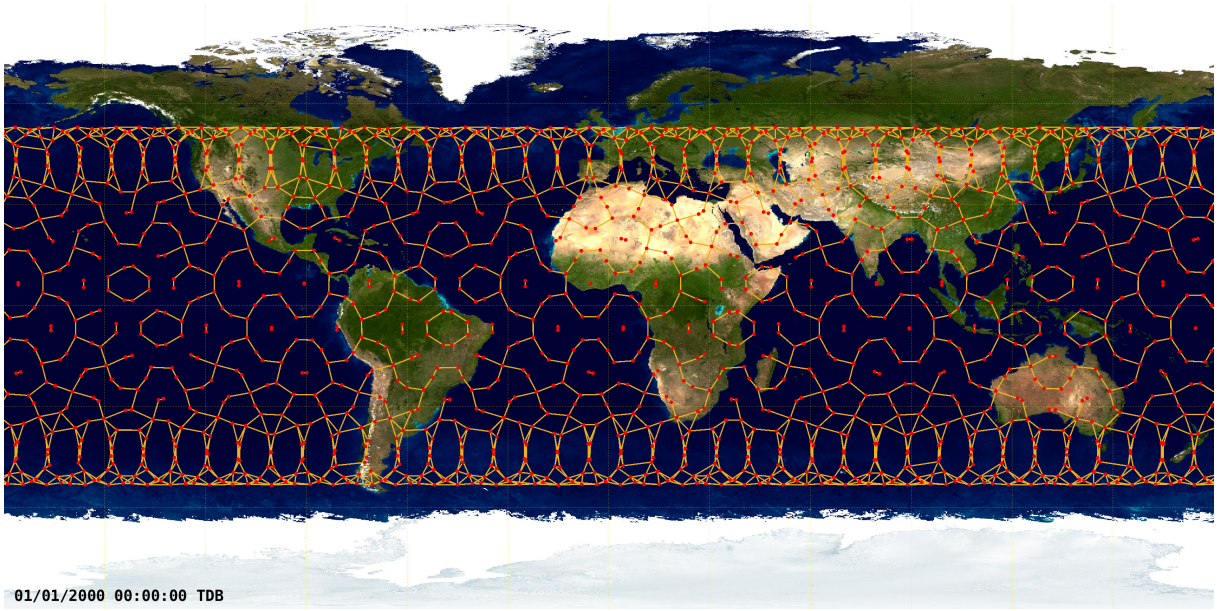


Figure 5.2: Snapshot of ground track and tracking output couplings of the simulated constellation at 0 TDB seconds since J2000.

$R_{LOS} = 2\sqrt{\bar{a}^2 - (R_{\oplus} + 80\,\mathrm{km})^2} = 5,014\,\mathrm{km}$ [105]. This range is supported by an ISL system, since multiple laser ISLs of up to $4,900\,\mathrm{km}$ have been reported between LEO satellites since 2008 [106]. To evaluate the volatility of tracking couplings between satellites, a pair of satellites that establishes a tracking coupling in a particular time instant is chosen at random and then the positions of these satellites are backtracked until they are out of line-of-sight range. This procedure is repeated several times. Figure 5.3 depicts an histogram of the interval of time that the pairs of satellites remained in line-of-sight range.

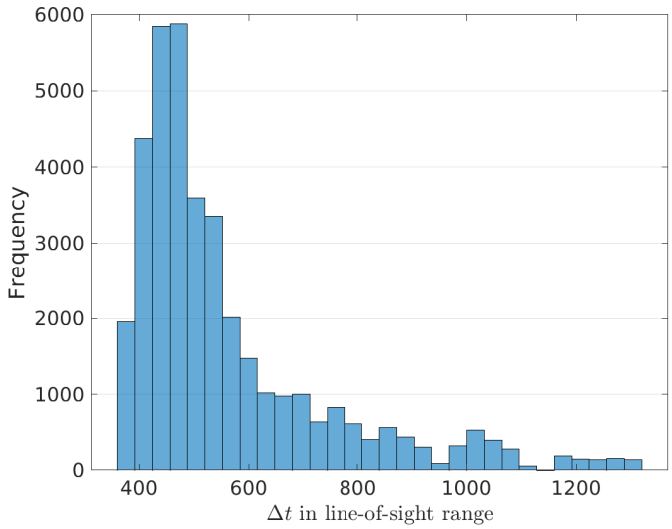


Figure 5.3: Intervals of time two satellites are in line-of-sight range before establishing a tracking coupling.

Consider the computation of the RHC gains over a generic window $\{k, \ldots, k+H-1\}$, according to the scheduling solution proposed in Section 4.5. The computation of this window starts at $t_s = kT_c - (H+2)T_t$ with the computation of the gains at its end, which corresponds to time instant $t_e = kT_c + HT_c$.

If none of the satellites that are coupled at t_e are in line-of-sight at t_s, then it is pointless to consider such parameter H, since the restricted coupling neighborhood (4.11) of each satellite would not contain any other satellites due to communication constraints. Notice that, in Figure 5.3, no pair of satellites remained in line-of-sight range for more than $\Delta t_{max} = 1320\,\mathrm{s}$. Therefore, the constraint

$$(H+2)T_t + HT_c < \Delta t_{max} \tag{5.9}$$

arises from this observation. Likewise, no restriction due to communication requirements can be enforced for the first d discrete time instants of each RHC finite-window, as concluded in Section 4.5. The computation of the first d gains of the window starts at $t_s = kT_c - (d+1)T_t$ with the computation of the gain that corresponds to time instant $t_e = kT_c + dT_c$. To ensure that the communication requirements are met, the satellites that are coupled at t_e must be in line-of-sight, at least, since t_s. From Figure 5.3, the minimum time for the maintenance of line-of-sight is $\Delta t_{min} = 360\,\mathrm{s}$. Thus, the constraint

$$(d+1)T_t + T_c d < \Delta t_{min}. \tag{5.10}$$

arises from this observation. After algebraic manipulation of (5.9) and (5.10) and considering non-overlapping computation windows, as described in Section 4.3, one obtains

$$\begin{cases} H < (\Delta t_{max} - 2T_t)/(T_t + T_c) \\ d < (\Delta t_{min} - T_t)/(T_t + T_c) \\ d \geq (H+2)T_t/T_c \end{cases} = \begin{cases} H < 120.7 \\ d < 32.6 \\ d \geq 0.2 + H/10\,. \end{cases}$$

The parameters $H = 100$ and $d = 25$ were chosen from the rather tight constraints above. Note that these are still valid even if T_t is doubled.

The weighting matrices $\mathbf{Q}_i^{rel}(k) \in \mathbb{R}^{2\times 2}$, $\mathbf{Q}_i^{in}(k) \in \mathbb{R}^{4\times 4}$, and $\mathbf{R}_i(k) \in \mathbb{R}^{3\times 3}$ were adjusted empirically and set to $\mathbf{Q}_i^{rel}(k) = (1/10^{-4})^2\mathbf{I}$, $\mathbf{Q}_i^{in}(k) = \mathrm{diag}(1/(\bar{a}10^{-4})^2), 1/(0.5\times 10^{-2})^2\mathbf{I}_2, 1/(10^{-2})^2)$, and $\mathbf{R}_i(k) = (1/C_{t1})^2\mathbf{I}_2$.

5.4 Simulation results

In this section, the simulation results are presented for the aforementioned illustrative mega-constellation. A realistic nonlinear numerical simulation was computed making use of the high-fidelity TU Delft Astrodynamic Toolbox[1] (TUDAT) [107]. The orbit propagation of the satellites of the constellation makes use of NASA's SPICE ephemerides, and accounts for several perturbations:

1. Earth's gravity field EGM96 spherical harmonic expansion [103] up to degree and order 24;
2. Atmospheric drag NRLMSISE-00 model [108], assuming constant drag coefficient and section area;

[1] TUDAT documentation available at https://docs.tudat.space/ and source code at https://github.com/tudat-team/tudat-bundle/.

3. Cannon ball solar radiation pressure, assuming constant reflectivity coefficient and radiation area;

4. Third-body perturbations of the Sun, Moon, Venus, Mars, and Jupiter.

The numerical propagation is assured by a fourth-order Runge-Kutta integration method with fixed step-size of $T_c = 10$ s. The feedback control law computation is carried out in MATLAB. To implement the thrust feedback interface between TUDAT and MATLAB the *tudat-matlab-thrust-feedback* package, made available at `https://github.com/decenter2021/tudat-matlab-thrust-feedback`, is employed. A scheme of the simulation environment is depicted in Figure 5.4.

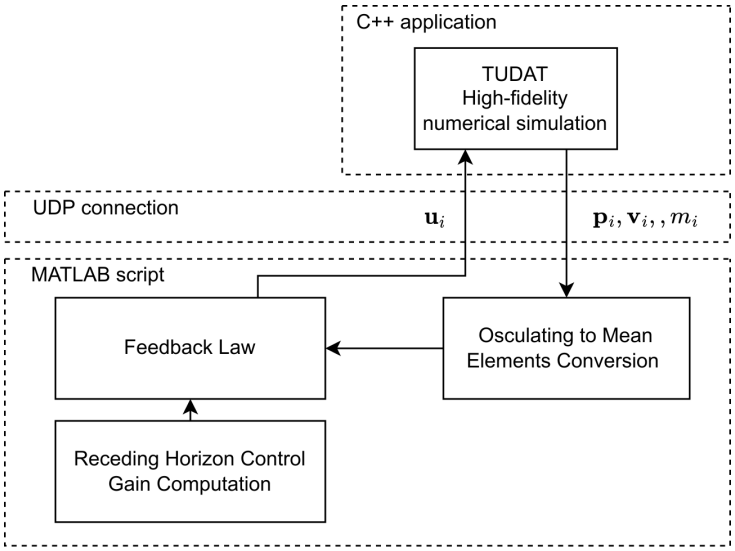


Figure 5.4: Scheme of the simulation environment.

Note that, for this single shell, the concatenation of the states of every satellite amounts to $6 \times 1584 = 9504$ states, which is the dimension of the global system of an equivalent centralized framework. Thus, the implementation of a centralized control algorithm would require performing computations, in real-time, with very large dimension matrices. For instance, matrix $\mathbf{P}(\tau)$, which is non-sparse in general, would amount to 722.6×10^6 bytes in double precision. Moreover, in a centralized framework, there would have to be all-to-all communication of large amounts of data over long distances via the MCC, which is achieved by several ground stations scattered across the Earth. On top of that, in areas where it may not be possible to ensure a direct link with a ground station, the information must flow through a path of satellites to an available ground station, thereby putting a lot of pressure on the communication system of the satellites. As a result, the intensive real-time global computations and sheer communication requirements render an equivalent centralized framework inviable.

A simulation of the mega-constellation during 12 orbital periods is carried out. An anchor for the nominal constellation is computed at 0 TDB seconds since J2000, according to (5.3). The anchor is not updated during the simulation. The evolution of the mean absolute error (MAE) in the semi-major axis, eccentricity, and inclination is depicted in Figures 5.5(a) and 5.5(b). To evaluate the performance of the relative tracking between the satellites, the mean argument of latitude error, $e_{u_i}(k)$, and longitude of ascending node error, $e_{\Omega_i}(k)$, are defined for each satellite i. Consider an instantaneous hypothetical anchor, computed according to (5.3), for each time instant k. These errors are defined as

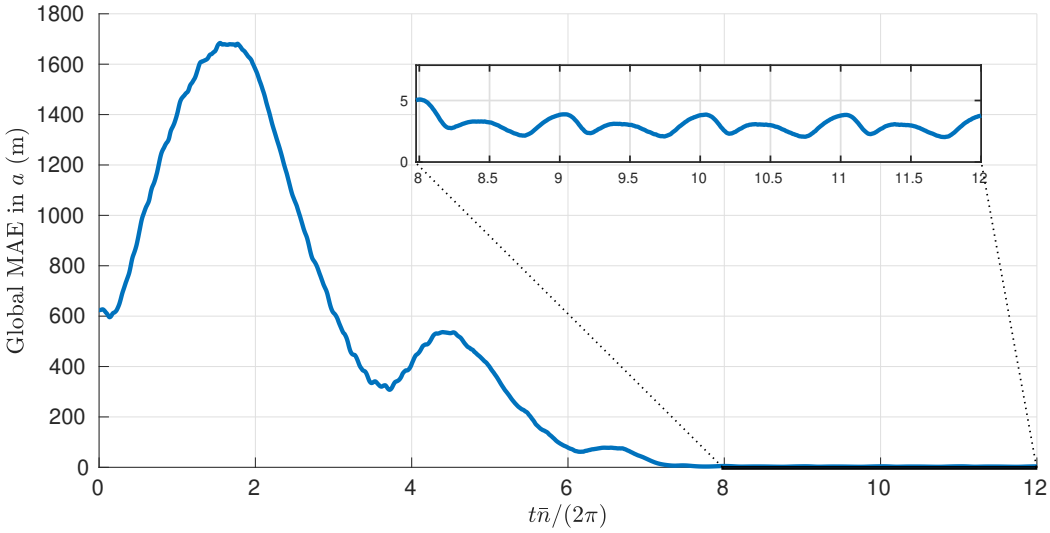
$e_{u_i}(k) := u_i(k) - \bar{u}_i(k)$ and $e_{\Omega_i}(k) := \Omega_i(k) - \bar{\Omega}_i(k)$, where $\bar{u}_i(k)$ and $\bar{\Omega}_i(k)$ are computed according to (5.2) making use of the aforementioned instantaneous hypothetical anchor for time instant k. It is very important to remark that these anchors are only employed for performance assessment purposes in a post-processing step, they are not involved in the control law in any way. The evolution of the MAE of the mean argument of latitude and longitude of ascending node is depicted in Figure 5.5(c). The steady-state MAE, obtained by averaging the MAE of the last three orbital periods of the simulation, is depicted in Table 5.2.

Table 5.2: Steady-state MAE.

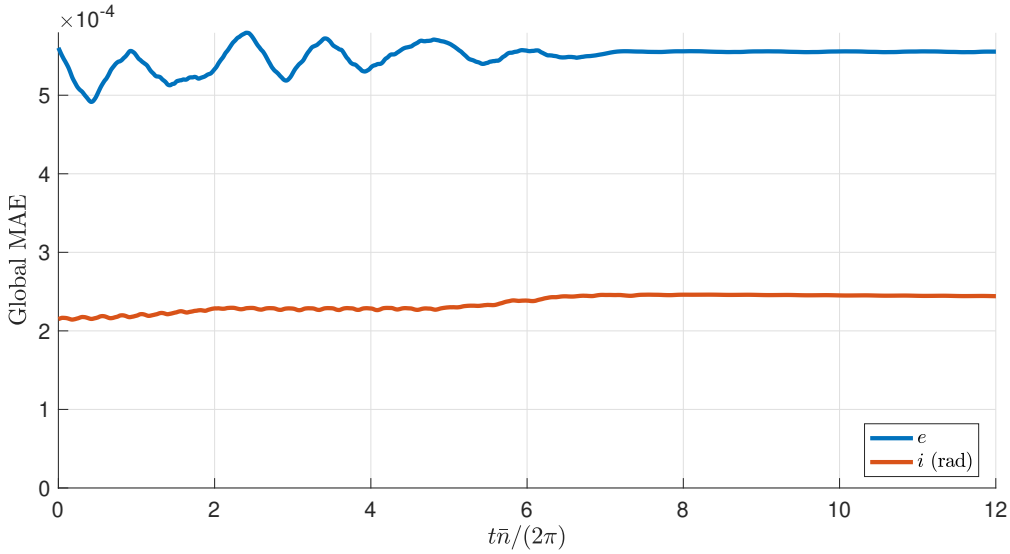
	$a - \bar{a}$ (m)	e	$i - \bar{i}$ (rad)	$\bar{a}e_{u_i}$ (m)	$\bar{a}e_{\Omega_i}$ (m)
Steady-state MAE	2.887	5.554×10^{-4}	2.450×10^{-4}	4.253	3.582

First, it is visible that the satellites of the constellation are successfully driven to their nominal semi-major axis and relative separations, despite the large initial errors. Second, although the method proposed in this thesis is designed for LTV systems under very strict communication, computational, and memory limitations, it is able to perform well in a network of systems with nonlinear dynamics. Third, in this simulation there was no need to update the anchor, confirming that its update period is large enough to allow for either a centralized or distributed computation. Fourth, it is visible in Table 5.2 that this solution reaches meter-level accuracy, not only for the semi-major axis, but also for the relative tracking components.

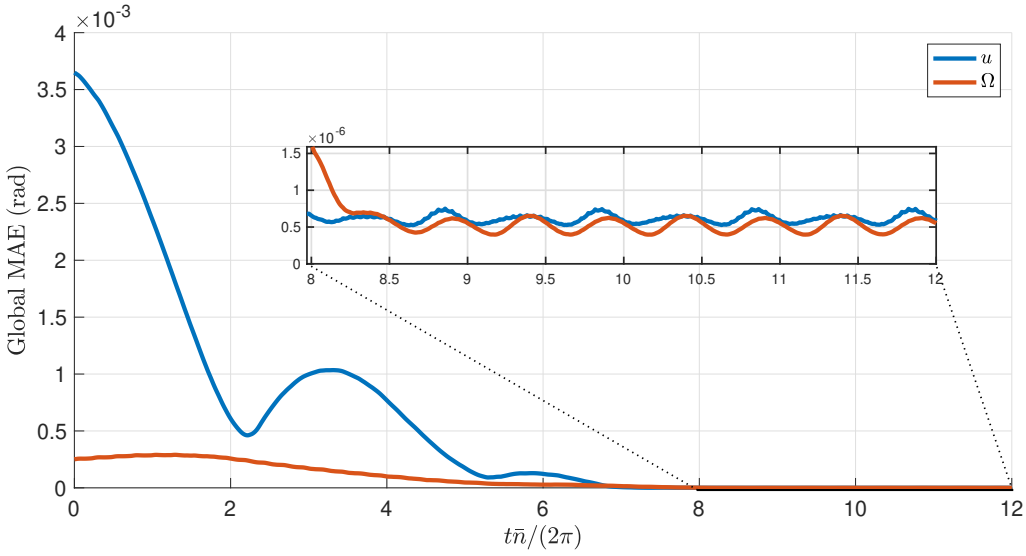
It is also interesting to analyze the evolution of a single satellite. Figures 5.6(a) and 5.6(b) show the evolution of the semi-major axis, eccentricity, and inclination tracking errors and Figure 5.6(c) depicts the evolution of the components of the control input, all for satellite 1. It is possible to notice that there is a steady-state error in the eccentricity and inclination tracking, but it is not significant. Figure 5.7 shows the trajectory of the mean argument of latitude and longitude of ascending node relative tracking errors. It is very interesting to remark that, for initial kilometer-level relative tracking errors, the proposed solution successfully drives and maintains the shape of the constellation with meter-level accuracy, as seen in Figure 5.7.



(a) Semi-major axis.

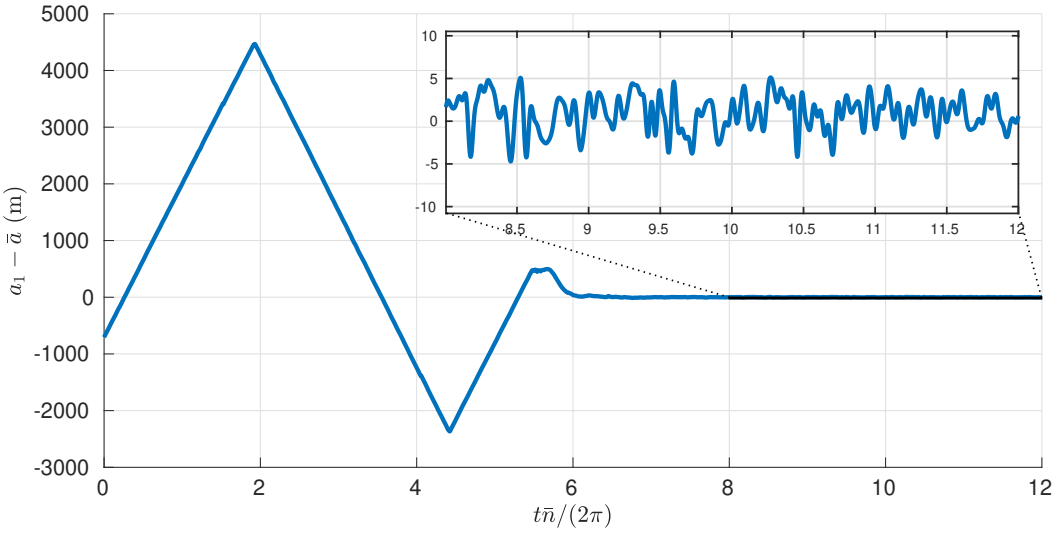


(b) Eccentricity and inclination.

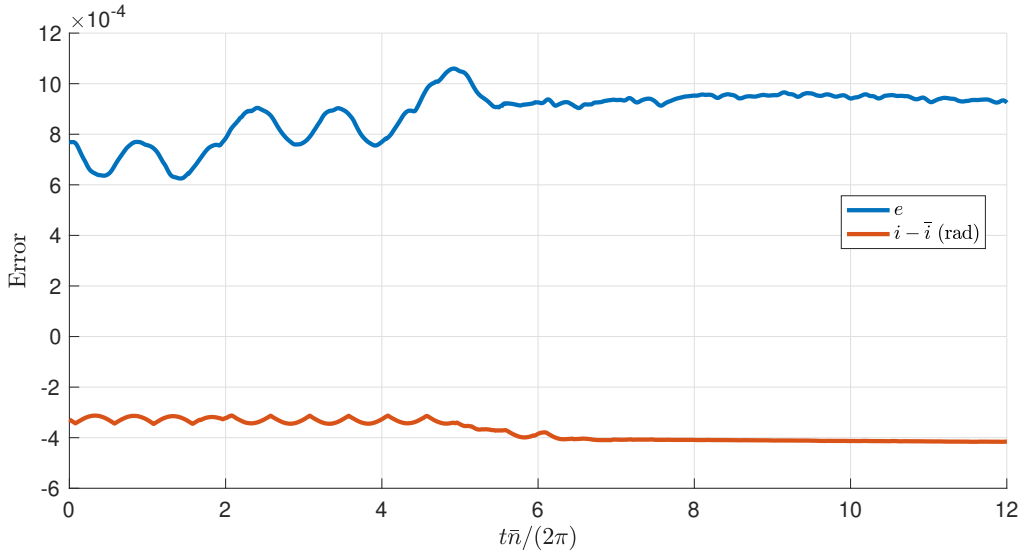


(c) Mean argument of latitude and longitude of ascending node.

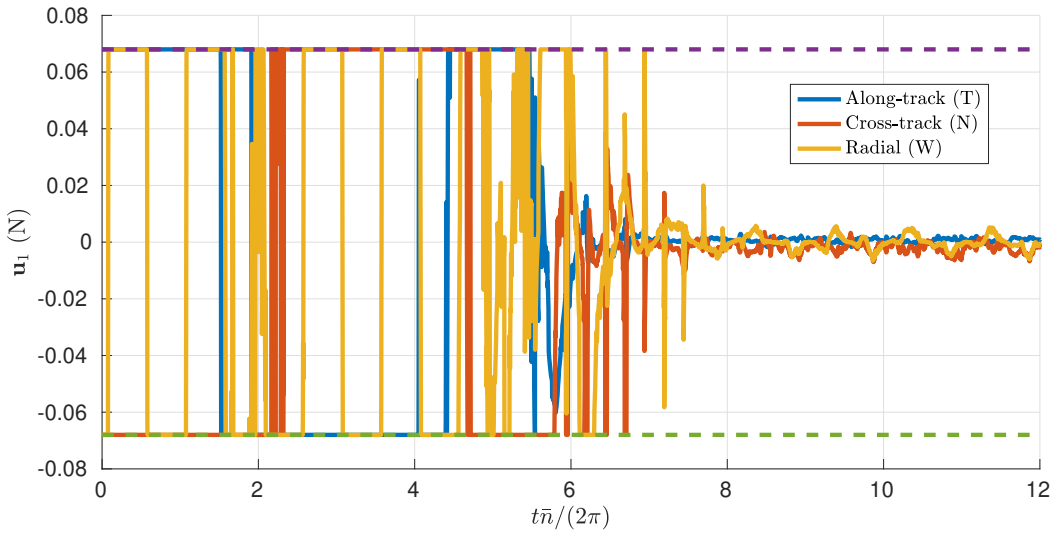
Figure 5.5: Evolution of the MAE.



(a) Semi-major axis tracking error.



(b) Eccentricity and inclination tracking errors.



(c) Components of the control input.

Figure 5.6: Simulation of satellite 1.

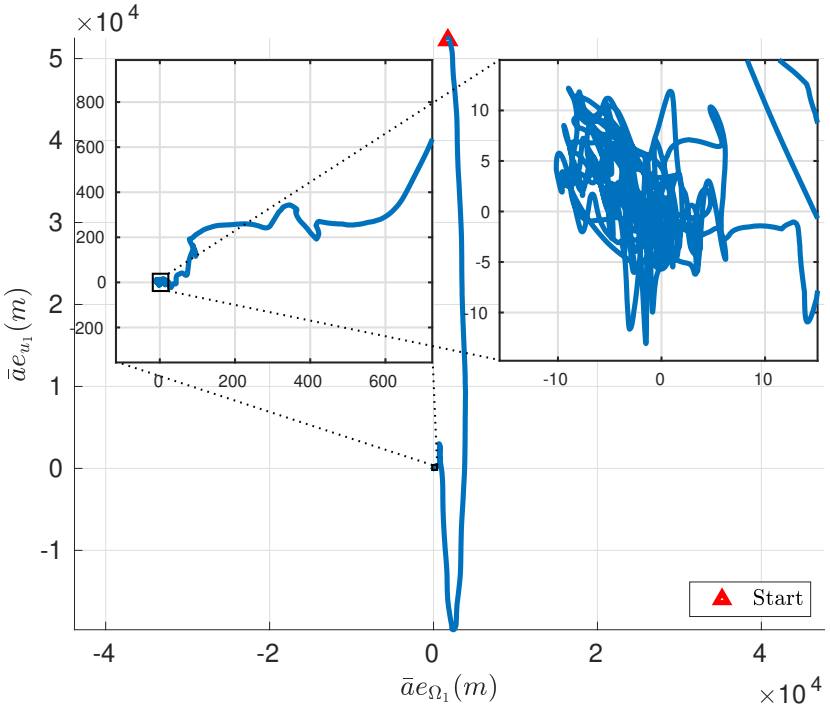


Figure 5.7: Trajectory of the mean argument of latitude and longitude of ascending node relative tracking errors, for satellite 1.

Chapter 6

Conclusions

Applications of very large-scale systems can no longer be ignored as a business opportunity by the industry nor as a natural shift towards efficiency, reliability, and scalability. They have potential for unprecedented societal impact in fields ranging from efficient large-scale distribution of electrical power to low-latency global broadband connectivity. Nevertheless, the transition from conceptualization to deployment of these emerging very large-scale tasks is hampered by severe technical challenges. Indeed, the economical viability of these ventures is doomed unless control algorithms that abide by these challenges are developed. Comparing the state-of-the-art solutions with the requirements befitting such very large-scale, it is clear that there is a void that needs to be addressed to enable these ground-breaking innovations.

First, it was possible to formulate the decentralized control problem globally in a RHC framework, modeling the limited communication of the decentralized scheme as a sparsity constraint on the global feedback gain. The implementation feasibility constraints on a very large-scale are modeled as bounds on the asymptotic growth rate of computational, memory, and communication resources with the dimension of the network.

Second, it was possible to conclude that the necessary conditions for a minimum of the decentralized RHC optimal synthesis problem, disregarding in a first instance the large-scale feasibility constraints, depend on the initial global state of the network, which is not available in a decentralized framework. A convex relaxation procedure is proposed to approximate the optimal solution of the synthesis problem without full knowledge of the state of the network. The control synthesis procedure that is obtained from the relaxation can be computed in closed-form. Moreover, a tracking solution is developed as well, building on the regulator results. Both regulator and tracker methods are validated resorting to a large-scale numeric simulation and experimental results of networks of interconnected tanks. Even though the system is nonlinear and the control inputs are subject to hard constraints, none of which were taken into account in the design of the decentralized controller, the proposed solution drives the output of the system to the reference signal, yielding very good performance.

Third, a distributed and decentralized control solution was successfully devised for the particular case of decoupled dynamics, building on the proposed convex relaxation procedure. It is shown that

a thoughtfully devised approximation allows to decouple the relaxed global synthesis procedure and to distribute the computations across the systems. A scheduling procedure for the computations is also put forward. This algorithm: i) is devised to account for communication and computation delays; ii) only requires local communication; and iii) ensures that the computational and memory requirements in each system do not scale with the dimension of the network. These requirements are in line with the large-scale implementation feasibility constraints. This is the main contribution of this thesis, since it enables the use of distributed control algorithms in applications of large-scale networks with unprecedented societal impact.

Fourth, the distributed decentralized control solution is applied to the orbit control problem of LEO mega-constellations. The LEO mega-constellation ventures that aim at providing low-latency global broadband connectivity are doomed to be economically inviable, unless suitable distributed and decentralized on-board TT&C solutions are employed. These are very large-scale networks of decoupled systems with nonlinear dynamics with a common control objective of maintaining the constellation shape. The control problem is formulated relying on a set of relative orbital elements, which allowed to write the shape-keeping task as a coupled sparse tracking output regulation problem. The proposed method shows promising performance for the orbit control problem of a shell of the Starlink mega-constellation.

Fifth, emphasis is put into the transparency and reproducibility of the results presented in this thesis. Indeed, the implementation of the methods put forward in this work, the source-code of all simulation examples, and the details of the experimental apparatus are available in well documented open-source repositories.

6.1 Future Work

Great strides still have to be made to further develop the subject of this thesis. Although a scalable well-performing distributed and decentralized solution has been developed, there is still a lot of room for improvement. Future research endeavors should address:

- The development of a distributed and decentralized solution for: i) coupled dynamics and decoupled tracking outputs and ii) coupled dynamics and tracking outputs, building on the one-step relaxation;
- An event-triggered actuation scheme to reduce the fuel consumption in the application to the LEO mega-constellation;
- The comparison in terms of tracking performance and fuel consumption with the bounding-box method;
- A LEO mega-constellation with multiple shells.

Bibliography

[1] C. R. Kube and H. Zhang. Collective robotics: From social insects to robots. *Adaptive behavior*, 2 (2):189–218, 1993.

[2] R. Arkin and G. Bekey. Robot colonies-editorial. *Autonomous Robots*, 4(1), 1997.

[3] A. Martinoli. *Swarm intelligence in autonomous collective robotics: From tools to the analysis and synthesis of distributed control strategies*. PhD thesis, École Polytechnique Fédérale de Lausanne, 1999.

[4] S. Bereg, J. M. Díaz-Báñez, M. A. Lopez, T. Rozario, and K. Valavanis. A decentralized geometric approach for the formation keeping in unmanned aircraft navigation. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 989–997, 2015.

[5] R. T. Thien and Y. Kim. Decentralized formation flight via pid and integral sliding mode control. *Aerospace Science and Technology*, 81:322–332, 2018.

[6] D. Viegas, P. Batista, P. Oliveira, and C. Silvestre. Decentralized observers for position and velocity estimation in vehicle formations with fixed topologies. *Systems & Control Letters*, 61(3):443–453, mar 2012. ISSN 01676911. doi: 10.1016/j.sysconle.2011.12.004.

[7] C. Yuan, S. Licht, and H. He. Formation learning control of multiple autonomous underwater vehicles with heterogeneous nonlinear uncertain dynamics. *IEEE Transactions on Cybernetics*, (99):1–15, 2017.

[8] J. Russell Carpenter. Decentralized control of satellite formations. *International Journal of Robust and Nonlinear Control*, 12(2-3):141–161, feb 2002. ISSN 1049-8923. doi: 10.1002/rnc.680.

[9] D. Ivanov, U. Monakhova, and M. Ovchinnikov. Nanosatellites swarm deployment using decentralized differential drag-based control with communicational constraints. *Acta Astronautica*, 159: 646–657, 2019.

[10] P. Radoglou-Grammatikis, P. Sarigiannidis, T. Lagkas, and I. Moscholios. A compilation of uav applications for precision agriculture. *Computer Networks*, 172:107148, 2020.

[11] K. A. Ghamry, M. A. Kamel, and Y. Zhang. Multiple uavs in forest fire fighting mission using particle swarm optimization. In *2017 International conference on unmanned aircraft systems (ICUAS)*, pages 1404–1409. IEEE, 2017.

[12] S. Awasthi, B. Balusamy, and V. Porkodi. Artificial intelligence supervised swarm uavs for reconnaissance. In *international conference on recent developments in science, engineering and technology*, pages 375–388. Springer, 2019.

[13] K. Z. Ang, X. Dong, W. Liu, G. Qin, S. Lai, K. Wang, D. Wei, S. Zhang, S. K. Phang, X. Chen, et al. High-precision multi-uav teaming for the first outdoor night show in singapore. *Unmanned Systems*, 6(01):39–65, 2018.

[14] A. Petrovsky, I. Kalinov, P. Karpyshev, D. Tsetserukou, A. Ivanov, and A. Golkar. The two-wheeled robotic swarm concept for mars exploration. *Acta Astronautica*, 194:1–8, 2022.

[15] Y. Li. Offtake feedforward compensation for irrigation channels with distributed control. *IEEE Transactions on Control Systems Technology*, 22(5):1991–1998, 2014.

[16] L.-D.-L. Nguyen, I. Prodan, L. Lefevre, and D. Genon-Catalot. Distributed model predictive control of irrigation systems using cooperative controllers. *IFAC-PapersOnLine*, 50(1):6564–6569, 2017.

[17] K. Aboudolas, M. Papageorgiou, and E. Kosmatopoulos. Store-and-forward based methods for the signal control problem in large-scale congested urban road networks. *Transportation Research Part C: Emerging Technologies*, 17(2):163–174, 2009.

[18] L. Pedroso and P. Batista. Decentralized store-and-forward based strategies for the signal control problem in large-scale congested urban road networks. *Transportation Research Part C: Emerging Technologies*, 132:103412, 2021. doi: 10.1016/j.trc.2021.103412.

[19] V. P. Singh, N. Kishor, and P. Samuel. Load frequency control with communication topology changes in smart grid. *IEEE Transactions on Industrial Informatics*, 12(5):1943–1952, 2016.

[20] E. Vlahakis, L. Dritsas, and G. Halikias. Distributed lqr design for a class of large-scale multi-area power systems. *Energies*, 12(14):2664, 2019.

[21] H. Farhangi. The path of the smart grid. *IEEE power and energy magazine*, 8(1):18–28, 2009.

[22] V. D. Blondel and J. N. Tsitsiklis. A survey of computational complexity results in systems and control. *Automatica*, 36(9):1249–1274, 2000.

[23] B. Marinescu. Output feedback pole placement for linear time-varying systems with application to the control of nonlinear systems. *Automatica*, 46(9):1524–1530, 2010.

[24] C. Daehnick, I. Klinghoffer, B. Maritz, and B. Wiseman. Large leo satellite constellations: Will it be different this time? Aerospace & Defense Insights - McKinsey & Company, 2020.

[25] B. Gavish. Leo/meo systems–global mobile communication systems. *Telecommunication Systems*, 8(2):99–141, 1997.

[26] V. Bonneau, B. Carle, L. Probst, and B. Pedersen. Low-earth orbit satellites: Spectrum access. Digital Transformation Monitor - European Commission, 2017.

[27] Y. Zhan, P. Wan, C. Jiang, X. Pan, X. Chen, and S. Guo. Challenges and solutions for the satellite tracking, telemetry, and command system. *IEEE Wireless Communications*, 27(6):12–18, 2020.

[28] I. Del Portillo, B. G. Cameron, and E. F. Crawley. A technical comparison of three low earth orbit satellite constellation systems to provide global broadband. *Acta Astronautica*, 159:123–135, 2019.

[29] N. Pachler, I. del Portillo, E. F. Crawley, and B. G. Cameron. An updated comparison of four low earth orbit satellite constellation systems to provide global broadband. In *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–7. IEEE, 2021.

[30] L. Bakule. Decentralized control: Status and outlook. *Annual Reviews in Control*, 38(1):71–80, 2014.

[31] K.-K. Oh, M.-C. Park, and H.-S. Ahn. A survey of multi-agent formation control. *Automatica*, 53: 424–440, 2015.

[32] D. Ding, Q.-L. Han, Z. Wang, and X. Ge. A survey on model-based distributed control and filtering for industrial cyber-physical systems. *IEEE Transactions on Industrial Informatics*, 15(5):2483–2499, 2019.

[33] E. Espina, J. Llanos, C. Burgos-Mellado, R. Cardenas-Dobson, M. Martinez-Gomez, and D. Sáez. Distributed control strategies for microgrids: An overview. *IEEE Access*, 8:193412–193448, 2020.

[34] M. Kordestani, A. A. Safavi, and M. Saif. Recent survey of large-scale systems: architectures, controller strategies, and industrial applications. *IEEE Systems Journal*, 15(4):5440–5453, 2021.

[35] H. S. Witsenhausen. A counterexample in stochastic optimum control. *SIAM Journal on Control*, 6(1):131–147, 1968.

[36] L. Lessard and S. Lall. Internal quadratic invariance and decentralized control. In *Proceedings of the 2010 American Control Conference*, pages 5596–5601. IEEE, 2010.

[37] L. Lessard and S. Lall. Convexity of decentralized controller synthesis. *IEEE Transactions on Automatic Control*, 61(10):3122–3127, 2015.

[38] J. Kulkarni and M. Campbell. An approach to magnetic torque attitude control of satellites via h-infinity control for ltv systems. In *2004 43rd IEEE Conference on Decision and Control (CDC)*, volume 1, pages 273–277. IEEE, 2004.

[39] R. Vazquez, F. Gavilan, and E. F. Camacho. Pulse-width predictive control for ltv systems with application to spacecraft rendezvous. *Control Engineering Practice*, 60:199–210, 2017.

[40] L. Zuo and S. A. Nayfeh. Structured h2 optimization of vehicle suspensions based on multi-wheel models. *Vehicle System Dynamics*, 40(5):351–371, 2003.

[41] G. Zhai, M. Yoshida, J. Imae, and T. Kobayashi. Decentralized h2 controller design for descriptor systems: An lmi approach. *Nonlinear Dynamics and Systems Theory*, 6(1):98–108, 2006.

[42] P. Shah and P. A. Parrilo. H2-optimal decentralized control over posets: A state-space solution for state-feedback. *IEEE Transactions on Automatic Control*, 58(12):3084–3096, 2013.

[43] M. Farhood, Z. Di, and G. E. Dullerud. Distributed control of linear time-varying systems interconnected over arbitrary graphs. *International Journal of Robust and Nonlinear Control*, 25(2): 179–206, 2015.

[44] D. Viegas, P. Batista, P. Oliveira, C. Silvestre, and C. P. Chen. Distributed state estimation for linear multi-agent systems with time-varying measurement topology. *Automatica*, 54:72–79, 2015.

[45] P. Jain, P. Kar, et al. Non-convex optimization for machine learning. *Foundations and Trends® in Machine Learning*, 10(3-4):142–363, 2017.

[46] S. H. Low. Convex relaxation of optimal power flow—part i: Formulations and equivalence. *IEEE Transactions on Control of Network Systems*, 1(1):15–27, 2014.

[47] S. H. Low. Convex relaxation of optimal power flow—part ii: Exactness. *IEEE Transactions on Control of Network Systems*, 1(2):177–189, 2014.

[48] D. Viegas, P. Batista, P. Oliveira, and C. Silvestre. Distributed controller design and performance optimization for discrete-time linear systems. *Optimal Control Applications and Methods*, pages 1–18, 2020. doi: 10.1002/oca.2669.

[49] A. Lamperski and L. Lessard. Optimal decentralized state-feedback control with sparsity and delays. *Automatica*, 58:143–151, 2015.

[50] C. W. Scherer. Structured hinf-optimal control for nested interconnections: A state-space solution. *Systems & Control Letters*, 62(12):1105–1113, 2013.

[51] T. Tanaka and C. Langbort. The bounded real lemma for internally positive systems and h-infinity structured static state feedback. *IEEE transactions on automatic control*, 56(9):2218–2223, 2011.

[52] C. Lidström and A. Rantzer. Optimal distributed h-infinity state feedback for systems with symmetric and hurwitz state matrix. *arXiv preprint arXiv:1510.00070*, 2015.

[53] P. Chanfreut, J. M. Maestre, and E. F. Camacho. A survey on clustering methods for distributed and networked control systems. *Annual Reviews in Control*, 52:75–90, 2021.

[54] A. H. Chow, R. Sha, and Y. Li. Adaptive control strategies for urban network traffic via a decentralized approach with user-optimal routing. *IEEE Transactions on Intelligent Transportation Systems*, 21(4):1697–1704, 2019.

[55] M. Bahramipanah, R. Cherkaoui, and M. Paolone. Decentralized voltage control of clustered active distribution network by means of energy storage systems. *Electric Power Systems Research*, 136: 370–382, 2016.

[56] M. Dai, H. Mu, and M. Wu. Junction-tree-based cooperative orbit determination for crosslink-augmented satellite constellations. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 230(10):1848–1859, 2016.

[57] K. Y. Leung, T. D. Barfoot, and H. H. Liu. Decentralized localization of sparsely-communicating robot networks: A centralized-equivalent approach. *IEEE Transactions on Robotics*, 26(1):62–77, 2009.

[58] L. C. Carrillo-Arce, E. D. Nerurkar, J. L. Gordillo, and S. I. Roumeliotis. Decentralized multi-robot cooperative localization using covariance intersection. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1412–1417. IEEE, 2013.

[59] T. R. Wanasinghe, G. K. Mann, and R. G. Gosine. Decentralized cooperative localization for heterogeneous multi-robot system using split covariance intersection filter. In *2014 Canadian Conference on Computer and Robot Vision*, pages 167–174. IEEE, 2014.

[60] L. Luft, T. Schubert, S. I. Roumeliotis, and W. Burgard. Recursive decentralized collaborative localization for sparsely communicating robots. In *Robotics: Science and Systems*. New York, 2016.

[61] L. Luft, T. Schubert, S. I. Roumeliotis, and W. Burgard. Recursive decentralized localization for multi-robot systems with asynchronous pairwise communication. *The International Journal of Robotics Research*, 37(10):1152–1167, 2018.

[62] A. Bemporad and C. Rocchi. Decentralized linear time-varying model predictive control of a formation of unmanned aerial vehicles. In *2011 50th IEEE conference on decision and control and European control conference*, pages 7488–7493. IEEE, 2011.

[63] A. Bemporad and D. Barcelli. Decentralized model predictive control. *Networked control systems*, pages 149–178, 2010.

[64] W. B. Dunbar and R. M. Murray. Distributed receding horizon control with application to multi-vehicle formation stabilization. Technical report, California Institute of Technology Pasadena,Control and Dynamical Systems, 2004.

[65] A. Richards and J. How. A decentralized algorithm for robust constrained model predictive control. In *Proceedings of the 2004 American control conference*, volume 5, pages 4261–4266. IEEE, 2004.

[66] A. Richards and J. How. Decentralized model predictive control of cooperating uavs. In *2004 43rd IEEE Conference on Decision and Control (CDC)*, volume 4, pages 4286–4291. IEEE, 2004.

[67] T. Keviczky, F. Borrelli, and G. J. Balas. Decentralized receding horizon control for large scale dynamically decoupled systems. *Automatica*, 42(12):2105–2115, 2006.

[68] C.-W. Park, P. Ferguson, N. Pohlman, and J. P. How. Decentralized relative navigation for formation flying spacecraft using augmented cdgps. In *Proceedings of the 14th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2001)*, pages 2304–2315, 2001.

[69] F. Busse and J. How. Real-time experimental demonstration of precise decentralized relative navigation for formation flying spacecraft. In *AIAA guidance, navigation, and control conference and exhibit*, page 5003, 2002.

[70] P. Ferguson and J. How. Decentralized estimation algorithms for formation flying spacecraft. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 5442, 2003.

[71] M. Mandic, L. Breger, and J. How. Analysis of decentralized estimation filters for formation flying spacecraft. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 5135, 2004.

[72] X. Wang, W. Qin, Y. Bai, and N. Cui. A novel decentralized relative navigation algorithm for spacecraft formation flying. *Aerospace Science and Technology*, 48:28–36, 2016.

[73] J. Collins, S. Dawson, and J. Wertz. Autonomous constellation maintenance system. In *10th Annual AIAA/USU Conference on Small Satellites*. AIAA, 1996.

[74] N. H. Shah. *Automated station-keeping for satellite constellations*. PhD thesis, Massachusetts Institute of Technology, 1997.

[75] B. L. Kantsiper. *A systematic approach to station-keeping of constellations of satellites*. PhD thesis, Massachusetts Institute of Technology, 1998.

[76] M. Tillerson, G. Inalhan, and J. P. How. Co-ordination and control of distributed spacecraft systems using convex optimization techniques. *International Journal of Robust and Nonlinear Control*, 12 (2-3):207–242, 2002.

[77] E. Sin, H. Yin, and M. Arcak. Passivity-based distributed acquisition and station-keeping control of a satellite constellation in areostationary orbit. In *Dynamic Systems and Control Conference*, volume 84287. American Society of Mechanical Engineers, 2020.

[78] S. D'Amico. *Autonomous formation flying in low earth orbit*. PhD thesis, TU Delft, 2010.

[79] L. Pedroso and P. Batista. Discrete-time decentralized linear quadratic control for linear time-varying systems. *International Journal of Robust and Nonlinear Control*, 2021. doi: 10.1002/rnc.5772.

[80] L. Pedroso and P. Batista. Reproducible low-cost flexible quadruple-tank process experimental setup for control educators, practitioners, and researchers. *Journal of Process Control*, 118:82–94, 2022. doi: 10.1016/j.jprocont.2022.08.010.

[81] L. Pedroso and P. Batista. Distributed decentralized receding horizon control for very large-scale networks with application to leo satellite mega-constellations, 2022. arXiv. Preprint. doi: 10.48550/arXiv.2209.14951.

[82] D. B. West. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, NJ, 1996.

[83] W. D. Wallis. *A beginner's guide to graph theory*. Springer Science & Business Media, 2010.

[84] L. Pedroso and P. Batista. Efficient algorithm for the computation of the solution to a sparse matrix equation in distributed control theory. *Mathematics*, 9(13), 2021. doi: 10.3390/math9131497.

[85] B. Anderson and J. B. Moore. Optimal control: linear quadratic methods. 1990.

[86] H. Kwakernaak and R. Sivan. *Linear optimal control systems*, volume 1. Wiley-interscience New York, 1972.

[87] I. R. Shafarevich and A. O. Remizov. *Linear algebra and geometry*. Springer Science & Business Media, 2012.

[88] D. Heller. Some aspects of the cyclic reduction algorithm for block tridiagonal linear systems. *SIAM Journal on Numerical Analysis*, 13(4):484–496, 1976.

[89] S. P. Hirshman, K. S. Perumalla, V. E. Lynch, and R. Sanchez. Bcyclic: A parallel block tridiagonal matrix cyclic solver. *Journal of Computational Physics*, 229(18):6392–6404, 2010.

[90] K. H. Johansson. The quadruple-tank process: A multivariable laboratory process with an adjustable zero. *IEEE Transactions on Control Systems Technology*, 2000. ISSN 10636536. doi: 10.1109/87.845876.

[91] W. Li and E. Todorov. Iterative linear quadratic regulator design for nonlinear biological movement systems. In *ICINCO (1)*, pages 222–229, 2004.

[92] E. Todorov and W. Li. A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 300–306. IEEE, 2005.

[93] D. A. Vallado. *Fundamentals of Astrodynamics and Applications*. McGraw-Hill Companies, Inc. College Custom Series, 1997.

[94] D. Jakovetić, J. Xavier, and J. M. Moura. Fast distributed gradient methods. *IEEE Transactions on Automatic Control*, 59(5):1131–1146, 2014.

[95] J. Sullivan, A. Koenig, and S. D'Amico. Improved maneuver-free approach to angles-only navigation for space rendezvous. In *26th AAS/AIAA Space Flight Mechanics Meeting, Napa, CA*, 2016.

[96] A. W. Koenig, T. Guffanti, and S. D'Amico. New state transition matrices for relative motion of spacecraft formations in perturbed orbits. In *AIAA/AAS Astrodynamics Specialist Conference*, page 5635, 2016.

[97] G. Di Mauro, R. Bevilacqua, D. Spiller, J. Sullivan, and S. D'Amico. Continuous maneuvers for spacecraft formation flying reconfiguration using relative orbit elements. *Acta Astronautica*, 153: 311–326, 2018.

[98] S. Spiridonova, M. Kirschner, and U. Hugentobler. Precise mean orbital elements determination for leo monitoring and maintenance. In *24th International Symposium on Space Flight Dynamics, Laurel, MD*, 2014.

[99] M. Eckstein and F. Hechler. *A reliable derivation of the perturbations due to any zonal and tesseral harmonics of the geopotential for nearly-circular satellite orbits*. ESRO sr-13, ESOC, 1970.

[100] W. M. Kaula. *Theory of satellite geodesy: applications of satellites to geodesy*. Blaisdell Publ. Company, 1966.

[101] C. Hwang. Gravity recovery using cosmic gps data: application of orbital perturbation theory. *Journal of Geodesy*, 75(2):117–136, 2001.

[102] C. Hwang and L.-S. Hwang. Satellite orbit error due to geopotential model error using perturbation theory: applications to rocsat-2 and cosmic missions. *Computers & geosciences*, 28(3):357–367, 2002.

[103] F. Lemoine, J. Factor, and S. Kenyon. *The development of the joint NASA GSFC and the National Imagery and Mapping Agency (NIMA) geopotential model EGM96*, volume 206861. National Aeronautics and Space Administration, Goddard Space Flight Center, 1998.

[104] J. Liang, A. U. Chaudhry, and H. Yanikomeroglu. Phasing parameter analysis for satellite collision avoidance in starlink and kuiper constellations. In *2021 IEEE 4th 5G World Forum (5GWF)*, pages 493–498. IEEE, 2021.

[105] D. Bhattacherjee and A. Singla. Network topology design at 27,000 km/hour. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, pages 341–354, 2019.

[106] Z. Sodnik, B. Furch, and H. Lutz. Optical intersatellite communication. *IEEE journal of selected topics in quantum electronics*, 16(5):1051–1057, 2010.

[107] K. Kumar, P. van Barneveld, D. Dirkx, J. Melman, E. Mooij, and R. Noomen. Tudat: a modular and robust astrodynamics toolbox. In *5th ICATT Conference*, pages 1–8. ESA, 2012.

[108] J. Picone, A. Hedin, D. P. Drob, and A. Aikin. Nrlmsise-00 empirical model of the atmosphere: Statistical comparisons and scientific issues. *Journal of Geophysical Research: Space Physics*, 107(A12):SIA–15, 2002.

[109] F. L. Lewis, D. Vrabie, and V. L. Syrmos. *Optimal control*. John Wiley & Sons, 2012.

[110] D. P. Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA, 3 edition, 2005. ISBN 1-886529-26-4.

[111] D. Viegas, P. Batista, P. Oliveira, and C. Silvestre. Discrete-time distributed Kalman filter design for formations of autonomous vehicles. *Control Engineering Practice*, 75:55–68, jun 2018. ISSN 09670661. doi: 10.1016/j.conengprac.2018.03.014.

Appendix A

Proofs

A.1 Proof of Theorem 3.1

The proof of the result follows the Lagrange-multiplier approach detailed, for instance, in [109]. The goal of using this approach is to ease the inclusion of the constraints of (2.6), which allows to write (2.6) as an unconstrained optimization problem. Writing an augmented performance index, $J'(k)$, that takes into account the linear feedback action (2.4), as well as the state equation (2.2), yields

$$\begin{aligned} J'(k) = \quad & \mathbf{x}^T(k+H)\mathbf{H}^T(k+H)\mathbf{Q}(k+H)\mathbf{H}(k+H)\mathbf{x}(k+H) \\ & + \sum_{\tau=k}^{k+H-1} \mathbf{x}^T(\tau)\left(\mathbf{H}^T(\tau)\mathbf{Q}(\tau)\mathbf{H}(\tau) + \mathbf{K}^T(\tau)\mathbf{R}(\tau)\mathbf{K}(\tau)\right)\mathbf{x}(\tau) \\ & + \sum_{\tau=k}^{k+H-1} \boldsymbol{\lambda}^T(\tau+1)\left[\left(\mathbf{A}(\tau) - \mathbf{B}(\tau)\mathbf{K}(\tau)\right)\mathbf{x}(\tau) - \mathbf{x}(\tau+1)\right] , \end{aligned} \tag{A.1}$$

where $\boldsymbol{\lambda}(\tau+1) \in \mathbb{R}^n$ is the Lagrange-multiplier associated with each of the constraints that arise from the state equation. The augmented performance index (A.1) is often written, for convenience, as a function of the Hamiltonian, defined, in this case, as

$$H'(\tau) = \mathbf{x}^T(\tau)\left(\mathbf{H}^T(\tau)\mathbf{Q}(\tau)\mathbf{H}(\tau) + \mathbf{K}^T(\tau)\mathbf{R}(\tau)\mathbf{K}(\tau)\right)\mathbf{x}(\tau) + \boldsymbol{\lambda}^T(\tau+1)\left(\mathbf{A}(\tau) - \mathbf{B}(\tau)\mathbf{K}(\tau)\right)\mathbf{x}(\tau) ,$$

which yields

$$\begin{aligned} J'(k) = \quad & \mathbf{x}^T(k+H)\mathbf{H}^T(k+H)\mathbf{Q}(k+H)\mathbf{H}(k+H)\mathbf{x}(k+H) - \boldsymbol{\lambda}^T(k+H)\mathbf{x}(k+H) \\ & + H'(k) + \sum_{\tau=k+1}^{k+H-1} \Big(H'(\tau) - \boldsymbol{\lambda}^T(\tau)\mathbf{x}(\tau)\Big) . \end{aligned} \tag{A.2}$$

Taking the differential of the augmented performance index (A.2), one obtains

$$\begin{aligned} dJ'(k) = \quad & \left(2\mathbf{H}^T(k+H)\mathbf{Q}(k+H)\mathbf{H}(k+H)\mathbf{x}(k+H) - \boldsymbol{\lambda}(k+H)\right)^T d\mathbf{x}(k+H) \\ & + \sum_{\tau=k+1}^{k+H} \left(\frac{\partial H'(\tau-1)}{\partial \boldsymbol{\lambda}(\tau)} - \mathbf{x}(\tau)\right)^T d\boldsymbol{\lambda}(\tau) + \left(\frac{\partial H'(k)}{\partial \mathrm{vec}\left(\mathbf{K}(k)\right)}\right)^T d\mathrm{vec}\left(\mathbf{K}(k)\right) \\ & + \left(\frac{\partial H'(k)}{\partial \mathbf{x}(k)}\right)^T d\mathbf{x}(k) + \sum_{\tau=k+1}^{k+H-1} \left[\left(\frac{\partial H'(\tau)}{\partial \mathrm{vec}\left(\mathbf{K}(\tau)\right)}\right)^T d\mathrm{vec}\left(\mathbf{K}(\tau)\right) + \left(\frac{\partial H'(\tau)}{\partial \mathbf{x}(\tau)} - \boldsymbol{\lambda}(\tau)\right)^T d\mathbf{x}(\tau)\right]. \end{aligned} \tag{A.3}$$

Define the set χ of integer pairs of the form (j,i) to index the non-zero entries of $\mathbf{K}(k)$ as

$$\begin{cases} (j,i) \in \chi & \text{if } [\mathbf{E}_{\mathcal{G}_c}]_{ji} \neq 0 \\ (j,i) \notin \chi & \text{otherwise} \end{cases}, j = 1, ..., m,\ i = 1, ..., n\,. \tag{A.4}$$

The necessary conditions for the constrained minimum follow from (A.3) and from the sparsity constraint. For a fixed initial state $\mathbf{x}(k)$ and free final state $\mathbf{x}(k+H)$, the constrained minimum requires that $dJ'(k) = 0$ holds for any: i) $d\mathbf{x}(\tau)$, with $\tau = k+1, ..., k+H$; ii) $d\boldsymbol{\lambda}(\tau)$, with $\tau = k+1, ..., k+H$; and iii) $[d\mathbf{K}(\tau)]_{ji}$, with $\tau = k, ..., k+H-1$ and $(j,i) \in \chi$. Hence, it follows that

$$\mathbf{x}(\tau+1) = \frac{\partial H'(\tau)}{\partial \boldsymbol{\lambda}(\tau+1)},\ \tau = k, ..., k+H-1\,, \tag{A.5a}$$

$$\boldsymbol{\lambda}(\tau) = \frac{\partial H'(\tau)}{\partial \mathbf{x}(\tau)},\ \tau = k+1, ..., k+H-1\,, \tag{A.5b}$$

$$\left[\frac{\partial H'(\tau)}{\partial \mathbf{K}(\tau)}\right]_{ji} = 0\,, \tau = k, ..., k+H-1,\ (j,i) \in \chi\,, \tag{A.5c}$$

$$[\mathbf{K}(\tau)]_{ji} = 0\,, \tau = k, ..., k+H-1,\ (j,i) \notin \chi\,, \tag{A.5d}$$

and

$$\boldsymbol{\lambda}(k+H) = 2\mathbf{H}^T(k+H)\mathbf{Q}(k+H)\mathbf{H}(k+H)\mathbf{x}(k+H). \tag{A.5e}$$

Above, (A.5a) is the state equation, (A.5b) is the costate equation, (A.5c) is the stationary condition, (A.5d) is the sparsity constraint, and (A.5e) is the boundary condition. It is interesting to remark the usefulness of the Hamiltonian function, which allows to write the constraints of the optimization problem as neat identities involving its partial derivatives. As the form of the boundary condition suggests, the Lagrage-multipliers can possibly be written as $\boldsymbol{\lambda}(\tau) = 2\mathbf{P}(\tau)\mathbf{x}(\tau)$, where $\mathbf{P}(\tau)$ is a symmetric positive semidefinite matrix. In that case, from the boundary condition (A.5e), it follows that

$$\mathbf{P}(k+H) = \mathbf{H}^T(k+H)\mathbf{Q}(k+H)\mathbf{H}(k+H). \tag{A.6}$$

In fact, making use of the costate equation (A.5b), this hypothesis yields

$$\mathbf{P}(\tau)\mathbf{x}(\tau) = \left(\mathbf{H}^T(\tau)\mathbf{Q}(\tau)\mathbf{H}(\tau) + \mathbf{K}^T(\tau)\mathbf{R}(\tau)\mathbf{K}(\tau)\right)\mathbf{x}(\tau) + \left(\mathbf{A}(\tau) - \mathbf{B}(\tau)\mathbf{K}(\tau)\right)^T \mathbf{P}(\tau+1)\mathbf{x}(\tau+1),$$

$\tau = k, \ldots, k+H-1$, which holds for every $\mathbf{x}(\tau)$ if and only if

$$\mathbf{P}(\tau) = \mathbf{H}^T(\tau)\mathbf{Q}(\tau)\mathbf{H}(\tau) + \mathbf{K}^T(\tau)\mathbf{R}(\tau)\mathbf{K}(\tau) + (\mathbf{A}(\tau) - \mathbf{B}(\tau)\mathbf{K}(\tau))^T \mathbf{P}(\tau+1)\,(\mathbf{A}(\tau) - \mathbf{B}(\tau)\mathbf{K}(\tau))\,. \quad \text{(A.7)}$$

For this reason, the hypothesis (3.4) on the form of the Lagrange multipliers, $\boldsymbol{\lambda}(\tau) = 2\mathbf{P}(\tau)\mathbf{x}(\tau)$, is valid, and $\mathbf{P}(\tau)$ is given by the recursive closed-form expression (3.2). Making use of (A.5c), and using also the closed-loop system dynamics

$$\mathbf{x}(\tau+1) = (\mathbf{A}(\tau) - \mathbf{B}(\tau)\mathbf{K}(\tau))\mathbf{x}(\tau)\,, \quad \text{(A.8)}$$

one can write

$$\left[\mathbf{R}(\tau)\mathbf{K}(\tau)\mathbf{x}(\tau)\mathbf{x}^T(\tau) - \mathbf{B}^T(\tau)\mathbf{P}(\tau+1)\,(\mathbf{A}(\tau) - \mathbf{B}(\tau)\mathbf{K}(\tau))\,\mathbf{x}(\tau)\mathbf{x}^T(\tau)\right]_{ji} = 0\,, \quad \text{(A.9)}$$

for all $(j,i) \in \chi$ and $\tau = k, \ldots k+H-1$. Finally, (3.1) follows immediately from (A.9) and (A.5d). One can also prove (3.4) by induction. Consider $\tau = k+H$. Note that $V_k(k+H) = \mathbf{x}^T(k+H)\mathbf{P}(k+H)\mathbf{x}(k+H)$, which follows directly from (A.6). Now consider a generic τ. It follows from (3.5), the command action (2.4), and the global dynamics (2.2) that

$$V_k(\tau) = V_k(\tau+1) + \mathbf{x}^T(\tau)\left(\mathbf{H}^T(\tau)\mathbf{Q}(\tau)\mathbf{H}(\tau) + \mathbf{K}^T(\tau)\mathbf{R}(\tau)\mathbf{K}(\tau)\right)\mathbf{x}(\tau). \quad \text{(A.10)}$$

Substituting the inductive hypothesis (3.4) in (A.10) and making use of the closed-loop system dynamics (A.8) yields

$$\begin{aligned} V_k(\tau) = \quad & \mathbf{x}^T(\tau)\left(\mathbf{H}^T(\tau)\mathbf{Q}(\tau)\mathbf{H}(\tau) + \mathbf{K}^T(\tau)\mathbf{R}(\tau)\mathbf{K}(\tau)\right)\mathbf{x}(\tau) \\ & + \mathbf{x}^T(\tau)\left((\mathbf{A}(\tau) - \mathbf{B}(\tau)\mathbf{K}(\tau))^T \mathbf{P}(\tau+1)\,(\mathbf{A}(\tau) - \mathbf{B}(\tau)\mathbf{K}(\tau))\right)\mathbf{x}(\tau)\,, \end{aligned}$$

which by comparison with (A.7) concludes the proof by induction.

A.2 Alternative proof of Theorem 3.1

Consider the cost-to-go defined in (3.5). To make use of the optimality principle, on which dynamic programming is based, assume that the optimal cost-to-go at time instant $\tau+1$, $V_k^\star(\tau+1)$, is known. Note that such optimal cost-to-go depends on the state of the system at time instant $\tau+1$, $\mathbf{x}(\tau+1)$. Therefore to put emphasis on this dependence, it is represented henceforth by $V_k^\star(\tau+1, \mathbf{x}(\tau+1))$. By the definition of the cost-to-go in (3.5), one can write

$$V_k(\tau, \mathbf{x}(\tau)) = \mathbf{x}^T(\tau)\mathbf{H}^T(\tau)\mathbf{Q}(\tau)\mathbf{H}(\tau)\mathbf{x}(\tau) + \mathbf{u}^T(\tau)\mathbf{R}(\tau)\mathbf{u}(\tau) + V_k\left(\tau+1, \mathbf{A}(\tau)\mathbf{x}(\tau) + \mathbf{B}(\tau)\mathbf{u}(\tau)\right). \quad \text{(A.11)}$$

Applying the principle of optimality [110, Proposition 1.3.1] to (A.11) yields

$$\begin{aligned} V_k^\star(\tau,\mathbf{x}(\tau)) = & \min_{\substack{\mathbf{K}(s)\in \mathrm{Sparse}(\mathbf{E}_{\mathcal{G}_c}) \\ s=\tau,\ldots,k+H-1}} V_k(\tau,\mathbf{x}(\tau)) \\ = & \min_{\mathbf{K}(\tau)\in \mathrm{Sparse}(\mathbf{E}_{\mathcal{G}_c})} \left(\mathbf{u}^T(\tau)\mathbf{R}(\tau)\mathbf{u}(\tau) + \min_{\substack{\mathbf{K}(s)\in \mathrm{Sparse}(\mathbf{E}_{\mathcal{G}_c}) \\ s=\tau+1,\ldots,k+H-1}} V_k\left(\tau+1, \mathbf{A}(\tau)\mathbf{x}(\tau)+\mathbf{B}(\tau)\mathbf{u}(\tau)\right) \right) \\ & + \mathbf{x}^T(\tau)\mathbf{H}^T(\tau)\mathbf{Q}(\tau)\mathbf{H}(\tau)\mathbf{x}(\tau). \end{aligned}$$

Given that $J(k) = V_k(k)$, the decentralized problem (2.6) can be written as H optimization problems, one for each gain of the finite window,

$$\min_{\mathbf{K}(\tau)\in \mathrm{Sparse}(\mathbf{E}_{\mathcal{G}_c})} \left(\mathbf{u}^T(\tau)\mathbf{R}(\tau)\mathbf{u}(\tau) + V_k^\star\left(\tau+1, \mathbf{A}(\tau)\mathbf{x}(\tau)+\mathbf{B}(\tau)\mathbf{u}(\tau)\right)\right), \tag{A.12}$$

$\tau = k, \ldots, k+H-1$. Making use of the closed-loop system dynamics (A.8), as well as of (3.4), which was proved in Section A.1 independently of the Lagrange multiplier approach, one can rewrite (A.12) as

$$\min_{\mathbf{K}(\tau)\in \mathrm{Sparse}(\mathbf{E}_{\mathcal{G}_c})} \left[\mathbf{x}^T(\tau)\left(\mathbf{K}^T(\tau)\mathbf{R}(\tau)\mathbf{K}(\tau) + (\mathbf{A}(\tau)-\mathbf{B}(\tau)\mathbf{K}(\tau))^T\mathbf{P}(\tau+1)(\mathbf{A}(\tau)-\mathbf{B}(\tau)\mathbf{K}(\tau))\right)\mathbf{x}(\tau)\right]. \tag{A.13}$$

Note that $\mathbf{P}(\tau+1)$ does not depend on $\mathbf{K}(\tau)$, thus, it is possible to solve (A.13) for each $\mathbf{K}(\tau)$ backward in time. Taking the derivative of the objective function of (A.13) w.r.t. the unconstrained entries of $\mathbf{K}(\tau)$ yields

$$\left[\mathbf{R}(\tau)\mathbf{K}(\tau)\mathbf{x}(\tau)\mathbf{x}^T(\tau) - \mathbf{B}^T(\tau)\mathbf{P}(\tau+1)\left(\mathbf{A}(\tau)-\mathbf{B}(\tau)\mathbf{K}(\tau)\right)\mathbf{x}(\tau)\mathbf{x}^T(\tau)\right]_{ji} = 0, \tag{A.14}$$

for all $(j,i) \in \chi$, where the set χ is defined as in (A.4), and $\tau = k, \ldots k+H-1$. Note that (A.14) is analogous to the constrained condition on each gain in (A.9) that arises using the Lagrange multiplier approach. Finally, (3.1) follows immediately from (A.14) and the sparsity constraint $\mathbf{K}(k) \in \mathrm{Sparse}(\mathbf{E}_{\mathcal{G}_c})$.

A.3 Proof of Theorem 3.2

The result is proved by induction. Consider $\tau = k$. Note that at the beginning of the finite window $\mathbf{x}(k) = \mathbf{x_c}(k)$. Consider, now, a generic time instant $\tau+1$. The input $\mathbf{u}(\tau)$ follows $\mathbf{u}(\tau) = \mathbf{K}(\tau)\mathbf{x}(\tau)$, where $\mathbf{K}(\tau) \in \mathrm{Sparse}(\mathbf{E}_{\mathcal{G}_c})$. By induction hypothesis, it follows that it can be written as $\mathbf{u}(\tau) = \mathbf{K}(\tau)\mathbf{x_c}(\tau)$ in vector form, and as

$$[\mathbf{u}(\tau)]_i = [\mathbf{K}(\tau)\mathbf{x_c}(\tau)]_i = \sum_{j\in {}^c\mathcal{D}_i^-} [\mathbf{K}(\tau)]_{ij}[\mathbf{x_c}(\tau)]_j,$$

for each component. Consider, now, each component of index i. If $[\mathbf{u_c}(\tau)]_i = 0$, then, by appropriate choice of $[\mathbf{K}(\tau)]_{ij}$ with $j \in {}^c\mathcal{D}_i^-$, it is possible to attain $[\mathbf{u}(\tau)]_i = [\mathbf{u_c}(\tau)]_i$. Due to the hypothesis that $i \in {}^c\mathcal{D}_i^- \ \forall i \in \mathcal{V}_c$, then ${}^c\mathcal{D}_i^- \neq \emptyset$. Furthermore, due to the hypothesis (3.6), if $[\mathbf{u_c}(\tau)]_i \neq 0$, it follows that at least one component $[\mathbf{x_c}(\tau)]_j$, with $j \in {}^c\mathcal{D}_i^-$, is non-null. Thus, by appropriate choice of $[\mathbf{K}(\tau)]_{ij}$ with $j \in {}^c\mathcal{D}_i^-$, it is possible to attain $[\mathbf{u}(\tau)]_i = [\mathbf{u_c}(\tau)]_i$. Given that by the inductive hypothesis $\mathbf{x}(\tau) = \mathbf{x_c}(\tau)$

and that there exists $\mathbf{K}(\tau) \in \mathrm{Sparse}(\mathbf{E}_{\mathcal{G}_c})$ so that $\mathbf{u}(\tau) = \mathbf{u_c}(\tau)$, it follows that $\mathbf{x}(\tau+1) = \mathbf{x_c}(\tau+1)$, thus completing the proof.

A.4 Proof of Theorem 3.3

Applying the one-step relaxation to the necessary conditions of a constrained minimum of (2.6) given by (3.1) yields (3.7). Taking the transpose of (3.7) yields

$$\begin{cases} \left[\mathbf{K}^T(\tau)\mathbf{S}(\tau) - \mathbf{A}^T(\tau)\mathbf{P}(\tau+1)\mathbf{B}(\tau)\right]_{ij} = 0, & [\mathbf{E}^T_{\mathcal{G}_c}]_{ij} \neq 0 \\ [\mathbf{K}^T(\tau)]_{ij} = 0, & [\mathbf{E}^T_{\mathcal{G}_c}]_{ij} = 0 \end{cases}, \ \tau = k, \ldots, k+T-1. \tag{A.15}$$

Note that (A.15) has the same form, for the transpose of the gain, as the equation that arises in the LTI formulation of the one-step method for the decentralized estimation problem, put forward in [111, Theorem 4.1]. The closed-form solution (3.8) follows from that result.

A.5 Proof of Theorem 3.4

The optimization problems (3.9) are solved backward in time making use of the recurrence (3.2) to compute $\mathbf{P}(\tau)$, similarly to the procedure to compute the one-step gains according to Theorem 3.3, which is outlined in Remark 3.3. Thus, the result is proved by showing that, for a known $\mathbf{P}(\tau+1)$, (3.9) and (3.8) are equivalent. For a time instant τ, $\mathbf{P}(\tau)$ is given by (3.2). Taking the derivative of its trace with respect to $\mathbf{K}(\tau)$ yields

$$\frac{\partial}{\partial \mathbf{K}(\tau)} \mathrm{tr}(\mathbf{P}(\tau)) = -2\mathbf{B}^T(\tau)\mathbf{P}(\tau+1)\mathbf{A}(\tau) + 2\mathbf{S}(\tau)\mathbf{K}(\tau)\,. \tag{A.16}$$

Define χ to index the non-zero entries of $\mathbf{E}_{\mathcal{G}_c}$ as in (A.4). Equaling the nonzero entries $(i,j) \in \chi$ of (A.16) to zero and introducing the sparsity constraint on the gain matrix yields (3.7). The solution to optimization problem (3.9) is, thus, given by (3.8).

A.6 Proof of Theorem 3.5

The following derivation makes use of the Lagrange multiplier approach. The Lagrangian function of the optimization problem can be written as

$$\begin{aligned} \mathcal{L}(k) = \frac{1}{2} \sum_{\tau=k}^{k+H-1} & \left(\bar{\mathbf{x}}(\tau+1) - \bar{\mathbf{x}}(\tau) - \mathbf{B}(\tau)\mathbf{u_a}(\tau)\right)^T \mathbf{H}^T(\tau+1)\mathbf{H}(\tau+1) \left(\bar{\mathbf{x}}(\tau+1) - \bar{\mathbf{x}}(\tau) - \mathbf{B}(\tau)\mathbf{u_a}(\tau)\right) \\ + \sum_{\tau=k}^{k+H} & \Big(\boldsymbol{\lambda}^T(\tau)\left(\left(\mathbf{A}(\tau) - \mathbf{I}\right)\bar{\mathbf{x}}(\tau) + \mathbf{B}(\tau)\bar{\mathbf{u}}(\tau)\right) + \boldsymbol{\gamma}^T(\tau)\left(\mathbf{H}(\tau)\bar{\mathbf{x}}(\tau) - \mathbf{r}(\tau)\right)\Big)\,. \end{aligned} \tag{A.17}$$

The necessary conditions for the constrained minimum follow from (A.17)

$$
\begin{aligned}
\frac{\partial \mathcal{L}(k)}{\partial \boldsymbol{\lambda}(\tau)} &= \left(\mathbf{A}(\tau)-\mathbf{I}\right)\bar{\mathbf{x}}(\tau)+\mathbf{B}(\tau)\bar{\mathbf{u}}(\tau)=\mathbf{0}\,,\tau=k,\ldots,k+H\,,\\
\frac{\partial \mathcal{L}(k)}{\partial \boldsymbol{\gamma}(\tau)} &= \mathbf{H}(\tau)\bar{\mathbf{x}}(\tau)-\mathbf{r}(\tau)=\mathbf{0}\,,\tau=k,\ldots,k+H\,,\\
\frac{\partial \mathcal{L}(k)}{\partial \bar{\mathbf{x}}(k)} &= \mathbf{H}^T(k+1)\mathbf{H}(k+1)\bar{\mathbf{x}}(k)-\mathbf{H}^T(k+1)\mathbf{H}(k+1)\bar{\mathbf{x}}(k+1)+\mathbf{H}^T(k+1)\mathbf{H}(k+1)\mathbf{B}(k)\mathbf{u_a}(k)\\
&\quad+\left(\mathbf{A}(k)-\mathbf{I}\right)^T\boldsymbol{\lambda}(k)+\mathbf{H}^T(k)\boldsymbol{\gamma}(k)=\mathbf{0},\\
\frac{\partial \mathcal{L}(k)}{\partial \bar{\mathbf{x}}(\tau)} &= \mathbf{H}^T(\tau)\mathbf{H}(\tau)\bar{\mathbf{x}}(\tau)+\mathbf{H}^T(\tau+1)\mathbf{H}(\tau+1)\bar{\mathbf{x}}(\tau)-\mathbf{H}^T(\tau+1)\mathbf{H}(\tau+1)\bar{\mathbf{x}}(\tau+1)-\mathbf{H}^T(\tau)\mathbf{H}(\tau)\bar{\mathbf{x}}(\tau-1)\\
&\quad+\mathbf{H}^T(\tau+1)\mathbf{H}(\tau+1)\mathbf{B}(\tau)\mathbf{u_a}(\tau)-\mathbf{H}^T(\tau)\mathbf{H}(\tau)\mathbf{B}(\tau-1)\mathbf{u_a}(\tau-1)+\left(\mathbf{A}(\tau)-\mathbf{I}\right)^T\boldsymbol{\lambda}(\tau)\\
&\quad+\mathbf{H}^T(\tau)\boldsymbol{\gamma}(\tau)=\mathbf{0},\ \tau=k+1,\ldots,k+H-1\,,\\
\frac{\partial \mathcal{L}(k)}{\partial \bar{\mathbf{x}}(k+H)} &= \mathbf{H}^T(k+H)\mathbf{H}(k+H)\bar{\mathbf{x}}(k+H)-\mathbf{H}^T(k+H)\mathbf{H}(k+H)\bar{\mathbf{x}}(k+H-1)\\
&\quad-\mathbf{H}^T(k+H)\mathbf{H}(k+H)\mathbf{B}(k+H-1)\mathbf{u_a}(k+H-1)+\left(\mathbf{A}(k+H)-\mathbf{I}\right)^T\boldsymbol{\lambda}(k+H)\\
&\quad+\mathbf{H}^T(k+H)\boldsymbol{\gamma}(k+H)=\mathbf{0},\\
\frac{\partial \mathcal{L}(k)}{\partial \bar{\mathbf{u}}(\tau)} &= \mathbf{B}^T(\tau)\boldsymbol{\lambda}(\tau)=\mathbf{0}\,,\tau=k,\ldots,k+H\,,\\
\frac{\partial \mathcal{L}(k)}{\partial \mathbf{u_a}(\tau)} &= \mathbf{B}^T(\tau)\mathbf{H}^T(\tau+1)\mathbf{H}(\tau+1)\mathbf{B}(\tau)\mathbf{u_a}(\tau)-\mathbf{B}^T(\tau)\mathbf{H}^T(\tau+1)\mathbf{H}(\tau+1)\bar{\mathbf{x}}(\tau+1)\\
&\quad+\mathbf{B}^T(\tau)\mathbf{H}^T(\tau+1)\mathbf{H}(\tau+1)\bar{\mathbf{x}}(\tau)=\mathbf{0}\,,\tau=k,\ldots,k+H-1\,,
\end{aligned}
$$

which can be written has the system of linear equations (3.16). As shown in Section 3.2, the linear equality constraint (3.10) has, at least, one solution $(\bar{\mathbf{x}}(\tau),\bar{\mathbf{u}}(\tau))$. Furthermore, let f_k denote the objective function of the optimization problem, *i.e.*, $f_k := \sum_{\tau=k}^{k+H-1}\mathbf{d}(\tau)^T\mathbf{H}^T(\tau+1)\mathbf{H}(\tau+1)\mathbf{d}(\tau)$. Note that f_k features positive semidefinite matrices $\mathbf{H}^T(\tau)\mathbf{H}(\tau)$, for $\tau=k+1,\ldots,k+H$, since $\mathbf{H}(\tau)$ has full rank for $\tau\in\mathbb{N}_0$. For that reason, there is at least one global constrained minimum for the minimization problem (3.15). Thus, at least one of the solutions to (3.16) corresponds to the global minimum.

Let p be the dimension and $\{\mathbf{v}_1,\ldots,\mathbf{v}_p\}$ a basis of the null space of matrix $\mathbf{G}$. The solutions to (3.16) are of the form

$$\bar{\boldsymbol{\chi}}^\star=\bar{\boldsymbol{\chi}}^*+\sum_{i=1}^{p}t_i\mathbf{v}_i\,,$$

where $\bar{\boldsymbol{\chi}}^*$ is a particular solution and $t_1,\ldots,t_p\in\mathbb{R}$. All the solutions to (3.16) are critical points, *i.e.*,

$$\left.\frac{\partial f_k}{\partial\bar{\boldsymbol{\chi}}}\right|_{\bar{\boldsymbol{\chi}}=\bar{\boldsymbol{\chi}}^\star}=\mathbf{0}\,.$$

Therefore, using the chain rule,

$$\frac{\partial f_k}{\partial t_i}=\left.\frac{\partial f_k}{\partial\bar{\boldsymbol{\chi}}}\right|_{\bar{\boldsymbol{\chi}}=\bar{\boldsymbol{\chi}}^\star}\cdot\frac{\partial\bar{\boldsymbol{\chi}}}{\partial t_i}=\mathbf{0}\cdot\mathbf{v}_i=0\,,$$

for $i=1,\ldots,p$. Hold every t_j constant, with $j\in\{1,\ldots,p\}\backslash\{i\}$. Then, given that f_k is continuous and

differentiable for $t_i \in \mathbb{R}$, f_k is constant over the set of solutions to (3.16), one of which is the global minimum. Therefore all of the solutions to (3.16) achieve global optimality.