Distributed decentralized receding horizon control for very large-scale networks with application to satellite mega-constellations

Leonardo Pedroso[1] and Pedro Batista[1]

[1]*Institute for Systems and Robotics, Instituto Superior Técnico, Universidade de Lisboa, Portugal*

Published version[†]

Available online on 4 Oct 2023

Cite as

Distributed decentralized receding horizon control for very large-scale networks with application to satellite mega-constellations

Leonardo Pedroso *, Pedro Batista

Institute for Systems and Robotics, Instituto Superior Técnico, Universidade de Lisboa, Portugal

ARTICLE INFO

ABSTRACT

The implementation feasibility of control algorithms over very large-scale networks calls for hard constraints regarding communication, computational, and memory requirements. In this paper, the decentralized receding horizon control problem for very large-scale networks of dynamically decoupled systems with a common, possibly time-varying, control objective is addressed. Each system is assumed to be modeled by linear time-varying dynamics, which can be leveraged to approximate nonlinear systems about successive points of operation. A distributed and decentralized receding horizon control solution is put forward, which: (i) takes communication delays into account; (ii) allows local communication exclusively; and (iii) whose computational and memory requirements in each computational unit do not scale with the dimension of the network. The scalability of the proposed solution enables emerging very large-scale applications of swarm robotics and networked control. This approach is applied to the orbit control problem of low Earth orbit mega-constellations, featuring high-fidelity numerical simulations for the Starlink mega-constellation.

1. Introduction

1.1. Motivation

The interest in decentralized control and estimation has been growing over the past decades since it enables technology that relies on large-scale networks of coupled systems. The well-known classical control solutions are centralized, which rely on computations on a single node and all-to-all communication between systems. The sheer communication and computational requirements of these solutions render its implementation infeasible as the dimension of the network increases. The emergence of applications in swarm robotics, such as unmanned aircraft formation flight (Bereg et al., 2015; Thien & Kim, 2018), unmanned underwater formations (Viegas et al., 2012; Yuan et al., 2017) and satellite formation control (Ivanov et al., 2019; Russell Carpenter, 2002), as well as of large-scale networks of spatially scattered processes such as irrigation networks (Li, 2014; Nguyen et al., 2017) and power distribution networks (Sampathirao et al., 2021; Singh et al., 2016; Vlahakis et al., 2019) have accelerated the demand for efficient decentralized algorithms.

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 effort in this field, it remains an open problem even for linear time-invariant (LTI) systems due to its intractability (Blondel & Tsitsiklis,

2000). On top of that, a significant portion of the envisioned large-scale applications have 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 (Marinescu, 2010). This approach yields a more tractable linear time-varying (LTV) system for which decentralized control strategies have to be designed. On one hand, solutions for time-invariant dynamics can be synthesized offline. On the other hand, 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. In fact, the time-varying nature of the problem calls for real-time synthesis, which requires additional communication, computational, and memory requirements. Furthermore, these resources are often very limited, which poses heavy constraints to the design problem. In fact, the solutions must cope with these limiting factors by distributing the load of the overall algorithm evenly among the systems that make up the network. For these reasons, a distributed and decentralized solution ought to be sought.

1.2. Scope

The focus of this paper is to design a decentralized and distributed receding horizon control (RHC) solution for very large-scale networks

---

of dynamically decoupled systems with a common control objective. The problem is formulated considering decoupled LTV dynamics for each system and coupled LTV tracking output dynamics expressed in a generic, possibly time-varying, topology to model the network-wise control objective. Severe design constraints are levied to ensure the feasibility of its real-time implementation on a very large-scale.

1.3. Related literature

Note that, as aforementioned, a control solution for a network of systems with decoupled nonlinear dynamics and coupled nonlinear tracking outputs can be obtained from this formulation making use of linearization techniques. For a comprehensive overview of the approaches to distributed RHC, see Bemporad and Barcelli (2010). Farhood et al. (2015) 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 node. 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. Recently, Pedroso and Batista (2023a) address the problem of designing a decentralized control solution for a network of agents modeled by LTV dynamics. In this work, a well-performing decentralized solution leveraging convex relaxation is presented, which is validated in a network of nonlinear systems employing linearization techniques. Nevertheless, similarly to Farhood et al. (2015), the real-time controller synthesis cannot be computed distributively. Thus, albeit these two works propose decentralized strategies, their real-time synthesis is centralized, whose communication, computational, and memory requirements become unbearable as the dimension of the network increases. This challenge has been noted and addressed to a great extent for the extended Kalman filter (EKF) problem over very large-scale networks of mobile robots. In this context, partially distributed solutions have been proposed relying on centralized-equivalent frameworks (Dai et al., 2016), bookkeeping (Leung et al., 2009), and covariance intersection and split covariance intersection methods (Carrillo-Arce et al., 2013; Wanasinghe et al., 2014). 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 Luft et al. (2016, 2018). 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 every system does not grow with the dimension of the network, the memory requirements do. Recently, the authors proposed a distributed decentralized EKF (Pedroso & Batista, 2023b) whose communication, memory, and computational requirements of each computational unit do not grow with the dimension of the network. In this paper, an approach similar to Pedroso and Batista (2023b) is employed. Even though the control-estimation duality usually allows to establish analogous results, the intricacies of distributed and decentralized RHC do not allow for a straightforward extension. Unlike the estimation problem, research into distributed RHC schemes for decoupled LTV systems is rather limited and focuses mainly on particular control problems. For instance, in Bemporad and Rocchi (2011) a solution is presented for the particular case of a formation of unmanned aerial vehicles. 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 Dunbar and Murray (2004). A decentralized RHC scheme suitable for leader–follower topologies is presented in Richards and How (2004a, 2004b). Another very interesting distributed and decentralized approach to the RHC problem over networks of decoupled nonlinear systems is proposed in Keviczky et al. (2006). 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.

1.4. Proposed approach

In this paper, we employ a decentralized framework in which each system is associated with a computational unit that computes its own control input making use of local communication and local state feedback exclusively. To live up to the challenges of an implementation over a very large-scale network, heavy design constraints regarding communication, computational, and memory requirements are imposed. First, neither the data transmissions between systems nor floating-point computations are considered to be instantaneous, thus the distributed and decentralized solution must account for and cope with the inevitable communication and computation delays. Second, the number of communication links that a system establishes with other systems shall not grow with the dimension of the network, i.e., the communication complexity of each system must be $\mathcal{O}(1)$. Third, the quantity of data stored in each computational unit shall not grow with the number of systems in the network. Fourth, the overall computational demand of the control solution must be distributed across all computational units and the computational complexity of algorithms implemented in each one must grow with $\mathcal{O}(1)$ with the dimension of the network. To devise such a solution, the propagation of the contribution to the global tracking cost of the correlation between the states of the systems is decoupled. This is achieved as the result of an approximation that is introduced, similarly to Luft et al. (2016, 2018). A novel framework is proposed whereby each system S_i keeps an estimate of the contribution to the global tracking cost of the correlation between every pair of systems whose tracking output is coupled with the state of S_i. These are updated in the computational unit associated with S_i, neglecting the influence on the global tracking cost of the correlation between systems whose tracking outputs do not depend on a common system whose tracking output is coupled with the state of S_i. Although it may seem puzzling at first sight, in what follows, 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.

In contrast with the vast majority of the RHC solutions in the literature, the proposed approach abides by the aforementioned very large-scale feasibility constraints. Furthermore, contrarily to the innovative approach in Keviczky et al. (2006) that follows such constraints, the proposed solution does not assume knowledge about the global equilibrium of the network. This is key, for instance, for the application considered in this work, whereby the global equilibrium is unknown.

1.5. Motivation for application in LEO mega-constellations

To evaluate the effectiveness of this solution and show its scalability to very large-scale networks, it is applied to the orbit control problem of low Earth orbit (LEO) satellite mega-constellations.

Despite the failure of most large LEO constellation projects during the 1990s, the increasing technological advances and demand for broadband connectivity led to a reawakening of these projects over the past decade. LEO communication systems have plenty of advantages over medium Earth orbit (MEO) and geosynchronous Earth orbit (GEO) systems (Gavish, 1997). With the current technology, they are unarguably a solution to meet the soaring demand for reliable low-latency, high capacity, global broadband connectivity. It is possible to notice 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. A number of these projects have been under development, such as Telesat Lightspeed, OneWeb, Starlink, and Project Kuiper, some of which are starting to be deployed. Not surprisingly, the greatest concern is their economical viability.

Nevertheless, the aforementioned paradigm change has not yet been accompanied by a paradigm change from an operations standpoint. In fact, as detailed by Zhan et al. (2020), 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. As a result, the constellation monitoring systems are still constituted by a main mission control center (MCC) with several ground terminals scattered across the Earth. This framework is unsuitable for mega-constellations since it is very challenging to implement in practice and expensive to maintain. In fact, all-to-all communication is required between all the satellites via the MCC to transmit large quantities of data over long distances, which introduces delays and requires high power consumption. If, for several reasons, namely geo-political factors, it is not possible to establish a direct link with a ground station over some areas, it is necessary to retransmit TT&C data with inter-satellite links (ISL) via a path of satellites to an available ground station. This continuous flow of data introduces an uneven communication pressure on these links, which calls for complex protocols and introduces delays. These are only but a few reasons to illustrate the inadequacy of this classical paradigm, which wastes much needed resources such as power and bandwidth. Given the paramount importance of cutting costs for the viability of mega-constellations, there has to be a paradigm change as far as TT&C architecture is concerned so that it is efficient and cost effective (Zhan et al., 2020). The management of satellite mega-constellations could be carried out in a decentralized TT&C architecture. Orbit determination and control, as well as other low-level constellation operations, could be carried out cooperatively relying on local communication between satellites assured by ISL. The shift towards a decentralized cooperative paradigm does not require all-to-all communication, the computational load is shared among the satellites, and data is transmitted over much shorter distances. Such a paradigm shift has self-evident benefits in terms of increased efficiency and cost-effectiveness. Albeit well-performing, the current decentralized solutions either require all-to-all communication, neglect inevitable communication and computation delays, employ extensive measurement bookkeeping, or are not computationally scalable, which is unsuitable to the control problem of LEO mega-constellations. On the contrary, the distributed and decentralized algorithm put forward in this paper follows heavy constraints to cope with limited communication, computational, and memory resources of space systems, which is critical for the onboard implementation in a mega-constellation.

In this paper, it is applied to the decentralized orbit control problem, allowing for the cooperative maintenance of the constellation shape. A novel approach that relies on a set of relative orbital elements is devised aiming for efficiency and fuel saving. The source code of a MATLAB implementation of the algorithm and of the numerical simulations is available as an example in the DECENTER Toolbox at https://decenter2021.github.io/examples/DDRHCStarlink/.

1.6. Statement of contributions

The main contributions of this paper are threefold. First, severe constraints regarding communication, computational, and memory requirements for the implementation of a RHC solution in real-time in a very large-scale are rigorously defined. Second, a novel distributed decentralized RHC method that abides by the very large-scale feasibility constraints is devised. Third, the scalability of the proposed methods is illustrated on a high-fidelity simulation of a very large-scale pressing application, for which, to the best of the authors' knowledge, no other feasible solutions exist in the literature.

1.7. Paper outline

The structure of the paper is as follows. In Section 2, the decentralized control problem is formulated together with the communication, computational, and memory constraints that the control solution ought to follow to enable a real-time implementation in very large-scale systems. In Section 3, the proposed distributed and decentralized RHC algorithm is derived. In Section 4, the algorithm put forward in this paper is applied to the orbit control problem of LEO mega-constellations. In Section 5, the principal findings and conclusions drawn from this paper are presented.

1.8. Notation

Throughout this paper, 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 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 $\mathrm{diag}(\mathbf{A}_1, \ldots, \mathbf{A}_N)$ represents the block diagonal matrix whose diagonal blocks are matrices $\mathbf{A}_1, \ldots, \mathbf{A}_N$. The Kronecker delta is denoted by δ_{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. Problem statement

The problem statement is introduced in three stages. In a first instance, in Section 2.1, the models of all of the systems of the network are defined and concatenated to express a global model for the whole network. Second, in Section 2.2, the local RHC controllers are defined and the control problem is formulated for the global controller. Third, in Section 2.3, the communication, computational, and memory constraints are established. It is worth noting that this problem is formulated and tackled for a generic network of systems with LTV dynamics and LTV tracking output couplings that reflect a network-wide control objective. This approach does not rely on any further assumptions concerning the individual dynamics of each system, the nature of the tracking output function, or the topology of the tracking couplings.

2.1. Model of the network

Consider a network of N systems, S_i with $i = 1, \ldots, N$, each associated with a computational unit, $\mathcal{T}_i$. Each system is dynamically decoupled and modeled by LTV dynamics. Each system has also an

LTV tracking output, which is coupled with a set of other systems. The tracking outputs can express a control objective that is common to all the systems (for an example, see Section 4). Thus, the topology of the network is portrayed by the tracking output couplings between systems, which can be time-varying. It may be studied as a directed graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ is the set of directed edges. An edge e directed from vertex j towards vertex i is denoted by $e = (j, i)$. The in-degree of a vertex i is denoted by v_i^-, which is the number of edges directed towards it, and its in-neighborhood is denoted by $\mathcal{D}_i^-$, which is the set of indices of the vertices from which such edges originate. Likewise, the out-degree of a vertex i is denoted by v_i^+, which is the number of edges directed from it, and its out-neighborhood is denoted by $\mathcal{D}_i^+$, which is the set of indices of the vertices towards which such edges are directed. In this setting, each system is represented by a vertex, i.e., system S_i is represented by vertex i, and, if the tracking output of S_i is coupled with the state of S_j, this interconnection is expressed by an edge $e = (j, i)$. It is crucial to emphasize that the direction of the edge holds significant importance. For example, if the tracking output of S_i depends on the state of S_j it does not imply the converse necessarily. Moreover, since the local goal of each system S_i is to drive the tracking output to zero, henceforth it considered without any loss of generality that $i \in \mathcal{D}_i^-$, i.e., the tracking output of system S_i depends on its state. Thus, each vertex of the topology graph is assumed to have a self-loop.

The dynamics of system S_i are modeled by the discrete-time LTV system

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

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 S_i; matrices $\mathbf{A}_i(k)$, $\mathbf{B}_i(k)$, and $\mathbf{H}_{i,j}(k)$ with $j \in \mathcal{D}_i^-$ are time-varying matrices, known in $\mathcal{T}_i$, that model the dynamics of system 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 (1).

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

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) \\ \mathbf{z}(k) = \mathbf{H}(k)\mathbf{x}(k), \end{cases} \quad (2)$$

where $\mathbf{x}(k) := \mathrm{col}(\mathbf{x}_1(k), \dots, \mathbf{x}_N(k)) \in \mathbb{R}^n$ is the global state vector, $\mathbf{u}(k) := \mathrm{col}(\mathbf{u}_1(k), \dots, \mathbf{u}_N(k)) \in \mathbb{R}^m$ is the global input vector, and $\mathbf{z}(k) := \mathrm{col}(\mathbf{z}_1(k), \dots, \mathbf{z}_N(k)) \in \mathbb{R}^o$ is the global tracking output vector; $\mathbf{A}(k) := \mathrm{diag}(\mathbf{A}_1(k), \dots, \mathbf{A}_N(k))$ and $\mathbf{B}(k) := \mathrm{diag}(\mathbf{B}_1(k), \dots, \mathbf{B}_N(k))$ model the dynamics of the global network; and $\mathbf{H}(k)$ is a block matrix whose block of indices (i, j) is $\mathbf{H}_{i,j}(k)$, if $j \in \mathcal{D}_i^-$, 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, v_i^-, the number of tracking output couplings of a system S_i, is bounded and it is independent of N. The innovative solution put forward herein exploits the sparsity of these couplings to enable a distributed and decentralized RHC algorithm subject to severe communication, computational, and memory constraints.

2.2. Decentralized receding horizon control problem

The goal of the proposed RHC controller is to regulate the global tracking output making use of local linear state feedback. On one hand, in a centralized setting the global state of the network is accessible to every system at the cost of all-to-all communication via a central node. On the other hand, in a decentralized setting, it is not the case: each system S_i only has access to a fraction of the global state. Herein, a fully decentralized configuration is addressed, i.e., at every discrete-time instant k, only the states of the systems in the in-neighborhood of S_i are accessible to $\mathcal{T}_i$. Section 2.3 explores this property in greater depth. The control input of S_i is, thus, of the form

$$\mathbf{u}_i(k) = -\sum_{j \in \mathcal{D}_i^-} \mathbf{K}_{i,j}(k)\mathbf{x}_j(k), \quad (3)$$

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

The aim is to design optimal controller gains with respect to a performance metric. It is important to remark that the gains of each system cannot be independently designed due to the tracking couplings between systems. As a result, the local controllers are concatenated to define a global controller, which is used to formulate a global RHC problem. The global control input is given by

$$\mathbf{u}(k) = -\mathbf{K}(k)\mathbf{x}(k), \quad (4)$$

where $\mathbf{K}(k)$ is the global gain matrix. Note that the global control law (4) is equivalent to the concatenation of the local control laws (3) if and only if $\mathbf{K}(k)$ follows the sparsity pattern of block matrix $\mathbf{E}_D$, whose block of indices (i, j) is given by

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

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

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

If all-to-all communication were possible, then $\mathbf{E}_D = \mathbf{1}$, corresponding to a centralized configuration.

It is now possible to state the global RHC problem. The goal is to minimize an 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 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

$$J(k) := \sum_{i=1}^N J_i(k)$$
$$= \sum_{i=1}^N \Bigg(\mathbf{z}_i^T(k+H)\mathbf{Q}_i(k+H)\mathbf{z}_i(k+H) \quad (5)$$
$$+ \sum_{\tau=k}^{k+H-1} \left(\mathbf{z}_i^T(\tau)\mathbf{Q}_i(\tau)\mathbf{z}_i(\tau) + \mathbf{u}_i^T(\tau)\mathbf{R}_i(\tau)\mathbf{u}_i(\tau) \right) \Bigg),$$

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 a scheme similar to model predictive control (MPC). One considers a finite window $\{k, \dots, k+H\}$, 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. 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, (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 $\mathbf{Q}(\tau) := \mathrm{diag}\left(\mathbf{Q}_1(\tau), \dots, \mathbf{Q}_N(\tau)\right)$ and $\mathbf{R}(\tau) := \mathrm{diag}\left(\mathbf{R}_1(\tau), \dots, \mathbf{R}_N(\tau)\right)$. Remark that devising a decentralized controller for a network of systems, whose dynamics are governed by the LTV system (1), is analogous to devising a controller (4) for the global system (2) whereby the gain follows a sparsity pattern. The goal is to optimally design a sequence of gains that satisfy the sparsity pattern corresponding to a fully decentralized configuration. It results, for a finite-horizon, in the following optimization problem

$$
\begin{aligned}
&\underset{\substack{\mathbf{K}(\tau)\in\mathbb{R}^{m\times n} \\ \tau\in\{k,\dots,k+H-1\}}}{\text{minimize}} && J(k) \\
&\text{subject to} && \mathbf{K}(\tau) \in \mathrm{Sparse}(\mathbf{E}_D), \\
& && \mathbf{x}(\tau+1) = (\mathbf{A}(\tau)-\mathbf{B}(\tau)\mathbf{K}(\tau))\,\mathbf{x}(\tau), \\
& && \tau = k,\dots,k+H-1.
\end{aligned}
\tag{6}
$$

The optimization problem (6) poses a significant challenge since it is nonconvex even for networks with LTI dynamics. A viable solution is to relax the optimization problem, rendering it convex and making it possible to employ well-established optimization techniques. Nevertheless, the relaxed solution is merely an approximation to the original problem, despite being optimal for the relaxed version. Therefore, a cautious relaxation approach is essential to minimize the deviation between the two solutions. This approach is employed to devise the control solution proposed in this work.

2.3. Feasibility constraints

Naturally, the control solution developed to solve (6) ought to be feasible to be implemented in real-time in a decentralized configuration. Specifically, the computation of each gain $\mathbf{K}_{i,j}(k)$, $j \in \mathcal{D}_i^-$, in $\mathcal{T}_i$ ought to abide by severe constraints regarding communication, computational, and memory requirements. In what follows, these constraints are detailed rigorously.

First, a crucial point to consider 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 designated as hard real-time transmissions, which require complex synchronization procedures. On the other hand, in a soft 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. Therefore, soft real-time transmission are robust to communication delays up to the discretization time-step. One can readily point out that the definition of the local control input (3) requires a hard real-time transmission. In fact, $\mathbf{x}_j(k)$, with $j \in \mathcal{D}_i^-$, 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. As a result, henceforth, the communication requirements are focused on the gain computation, for which hard real-time transmissions are not allowed.

Second, each system can only establish a communication link with a limited number of other systems in the network. While some solutions in the literature rely on all-to-all communication, they are not scalable for very large-scale networks. Thus, it is necessary to restrict the number of communication links established with each systems to grow with $\mathcal{O}(1)$ with the dimension of the network, N, ensuring that the communication complexity is $\mathcal{O}(1)$. It is crucial to emphasize that the communication limitations are enforced at the protocol level, i.e., governing the exchange of data between systems, rather than limiting the physical communication links. Specifically, it is not permitted for $\mathcal{T}_i$ to access data from $\mathcal{T}_k$ via intermediary systems that could retransmit the information.

Third, each computational unit has limited memory. Therefore, the quantity of data stored in each one must not scale with the dimension of the network, i.e., the data storage complexity of each computational unit must grow with $\mathcal{O}(1)$ with N.

Fourth, the implementation of the local control solution requires computational resources in each computational unit, which are also limited. As such, the burden of the computation of the global control algorithm ought to be shared among all computational units, such that each performs computations related with its associated system exclusively. Hence, the computational complexity of each computational unit ought to grow with $\mathcal{O}(1)$ with N.

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 computational unit must grow with $\mathcal{O}(1)$ with N.

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

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

3. Distributed and decentralized RHC

The aim is to devise a decentralized control solution that solves the optimization problem (6) subject to the communication, memory, and computational Constraints 1–4 detailed in Section 2.3, which are critical for a feasible implementation to very large-scale systems. The derivation of the distributed and decentralized RHC solution is carried out in three stages: (i) convex relaxation of (6); (ii) decoupling of the gain synthesis procedure; and (iii) scheduling of the computations. In Section 3.4, the communication, memory, and computational requirements of the proposed solution are thoroughly analyzed in light of the constraints that were levied. First, this approach is devised for a time-invariant network topology and, in Section 3.5, it is extended to a time-varying topology.

3.1. Convex relaxation

A similar approach to the one used for the unconstrained RHC methodology is employed relying on the convex relaxation of (6). As pointed out in Section 1, (6) has already been addressed in Pedroso and Batista (2023a) without taking the communication, memory, and computational Constraints 1–4 put forward in Section 2.3 into account, which are not satisfied by the solution proposed therein. Nevertheless, some interesting results in Pedroso and Batista (2023a) can be leveraged to devise a distributed solution. Consider the following preliminary result.

Lemma 3.1. *Let* $\mathbf{l}_j$ *denote a column vector whose entries are all set to zero except for the jth one, which is set to* 1. *The solutions to the necessary condition for a constrained minimum of the Hamiltonian of* (6) *follow*

$$
\begin{cases}
\mathbf{l}_j^T\left[\left(\mathbf{S}(\tau)\mathbf{K}(\tau)-\mathbf{B}^T(\tau)\mathbf{P}(\tau+1)\mathbf{A}(\tau)\right)\mathbf{x}(\tau)\mathbf{x}^T(\tau)\right]\mathbf{l}_i = 0 & ,[\mathbf{E}_D]_{ji}\neq 0 \\
\mathbf{l}_j^T\mathbf{K}(\tau)\mathbf{l}_i = 0 & ,[\mathbf{E}_D]_{ji}=0,
\end{cases}
\tag{7}
$$

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

$$
\begin{cases}
\mathbf{P}(k+H) = \mathbf{H}^T(k+H)\mathbf{Q}(k+H)\mathbf{H}(k+H) \\
\mathbf{P}(\tau) = \mathbf{H}(\tau)^T\mathbf{Q}(\tau)\mathbf{H}(\tau)+\mathbf{K}^T(\tau)\mathbf{R}(\tau)\mathbf{K}(\tau) \\
\qquad +(\mathbf{A}(\tau)-\mathbf{B}(\tau)\mathbf{K}(\tau))^T\mathbf{P}(\tau+1)(\mathbf{A}(\tau)-\mathbf{B}(\tau)\mathbf{K}(\tau)),
\end{cases}
\tag{8}
$$

and

$$
\mathbf{S}(\tau) := \mathbf{B}^T(\tau)\mathbf{P}(\tau+1)\mathbf{B}(\tau)+\mathbf{R}(\tau).
\tag{9}
$$

Furthermore, it follows that

$$\mathbf{x}^T(\tau)\mathbf{P}(\tau)\mathbf{x}(\tau) = \mathbf{z}^T(k+H)\mathbf{Q}(k+H)\mathbf{z}(k+H)$$
$$+ \sum_{s=\tau}^{k+H-1} \left(\mathbf{z}^T(s)\mathbf{Q}(s)\mathbf{z}(s) + \mathbf{u}^T(s)\mathbf{R}(s)\mathbf{u}(s) \right). \tag{10}$$

Proof. See Pedroso and Batista (2023a, Appendix A) . □

Lemma 3.1 puts emphasis on the nonconvexity of (6) since there are, in general, multiple solutions to the necessary condition of a constrained minimum. Herein, the convex relaxation approach is to approximate a single solution to (7), not necessarily optimal, which follows

$$\begin{cases} \mathbf{l}_j^T \left[\mathbf{S}(\tau)\mathbf{K}(\tau) - \mathbf{B}^T(\tau)\mathbf{P}(\tau+1)\mathbf{A}(\tau) \right] \mathbf{l}_i = 0 & , [\mathbf{E}_D]_{ji} \neq 0 \\ \mathbf{l}_j^T \mathbf{K}(\tau)\mathbf{l}_i = 0 & , [\mathbf{E}_D]_{ji} = 0. \end{cases} \tag{11}$$

This convex relaxation procedure, denoted as one-step relaxation, is analyzed with detail in Pedroso and Batista (2023a), where it is shown to be equivalent to three different relaxation hypothesis.

3.2. 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 (8) 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

$$\mathbf{P}_{p,q}(\tau) = \sum_{r \in \mathcal{D}_p^+ \cap \mathcal{D}_q^+} \mathbf{H}_{r,i}^T(\tau)\mathbf{Q}_r(\tau)\mathbf{H}_{r,j}(\tau) + \sum_{r \in \mathcal{D}_p^+ \cap \mathcal{D}_q^+} \mathbf{K}_{r,i}^T(\tau)\mathbf{R}_r(\tau)\mathbf{K}_{r,j}(\tau)$$
$$+ \sum_{r \in \mathcal{D}_p^+} \sum_{s \in \mathcal{D}_q^+} \left(\mathbf{A}_p(\tau)\delta_{pr} - \mathbf{B}_r(\tau)\mathbf{K}_{r,p}(\tau) \right)^T \tag{12}$$
$$\mathbf{P}_{r,s}(\tau+1) \left(\mathbf{A}_q(\tau)\delta_{qs} - \mathbf{B}_s(\tau)\mathbf{K}_{s,q}(\tau) \right).$$

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

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

which follows immediately from (9). Moreover, leveraging the aforementioned block decomposition, the relaxed conditions (11) 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

$$\begin{cases} \sum_{p \in \mathcal{D}_i^+} \mathbf{S}_{j,p}(\tau)\mathbf{K}_{p,i}(\tau) - \mathbf{B}_j^T(\tau)\mathbf{P}_{j,i}(\tau+1)\mathbf{A}_i(\tau) = \mathbf{0}, & j \in \mathcal{D}_i^+ \\ \mathbf{K}_{j,i}(\tau) = \mathbf{0}, & j \notin \mathcal{D}_i^+. \end{cases} \tag{13}$$

For each set $\mathcal{D}_i^+$, let $\mathcal{D}_i^+ = \{p_1^i, \ldots, p_{|\mathcal{D}_i^+|}^i\}$. 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 (13), it follows that

$$\tilde{\mathbf{K}}_i(\tau) = \tilde{\mathbf{S}}_i(\tau)^{-1}\tilde{\mathbf{P}}_i(\tau+1), \tag{14}$$

where

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

$$\tilde{\mathbf{S}}_i(\tau) := \begin{bmatrix} \mathbf{S}_{p_1^i,p_1^i} & \cdots & \mathbf{S}_{p_1^i,p_{|\mathcal{D}_i^+|}^i} \\ \vdots & \ddots & \vdots \\ \mathbf{S}_{p_{|\mathcal{D}_i^+|}^i,p_1^i} & \cdots & \mathbf{S}_{p_{|\mathcal{D}_i^+|}^i,p_{|\mathcal{D}_i^+|}^i} \end{bmatrix}, \tag{15}$$

and

$$\tilde{\mathbf{P}}_i(\tau+1) := \begin{bmatrix} \mathbf{B}_{p_1^i}^T(\tau)\mathbf{P}_{p_1^i,i}(\tau+1)\mathbf{A}_i(\tau) \\ \vdots \\ \mathbf{B}_{p_{|\mathcal{D}_i^+|}^i}^T(\tau)\mathbf{P}_{p_{|\mathcal{D}_i^+|}^i,i}(\tau+1)\mathbf{A}_i(\tau) \end{bmatrix}. \tag{16}$$

Note that, although (14) provides an expression to compute the gains of systems S_j with $j \in \mathcal{D}_i^+$ in relation to the state of system 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 (12) 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 S_i, and so on. Hence, this result is unfit to be implemented in a distributed framework. In what follows, under a thoughtful approximation, the computation of these gains is decoupled. Then, an algorithm is proposed to enable their distributed computation across the computational units of each system such that the communication, memory, and computational constraints detailed in Section 2.3 are followed.

Approximation 3.1. 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 paper, $\mathbf{P}_{r,s}(\tau+1)$ 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

$$\psi_i = \bigcup_{j \in \mathcal{D}_i^+} \phi_j, \tag{17}$$

with

$$\phi_i := \mathcal{D}_i^+ \times \mathcal{D}_i^+ = \{(p,q) \in \mathbb{N}^2 : p \in \mathcal{D}_i^+ \wedge q \in \mathcal{D}_i^+\}. \tag{18}$$

The main result of this paper follows from Approximation 3.1. Next, it is argued that this approximation makes sense in the context of RHC of a large-scale network. Note that, from (10) in Lemma 3.1, $\mathbf{P}_{r,s}(\tau)$ is a measure of the contribution of the correlation between the states of systems S_r and S_s to the global cost. Fig. 1 depicts the topology of Approximation 3.1 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 S_r and S_s are coupled with the output of a system S_k that is coupled with the output of S_i. As a result, to enable a decoupled computation of the synthesis of each local controller, each computational unit $\mathcal{T}_i$ keeps in memory 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 is stored and updated in $\mathcal{T}_i$ is denoted by $\mathbf{P}_{i,(p,q)}(\tau)$. Thus, making use of Approximation 3.1, (12) becomes

$$\mathbf{P}_{i,(p,q)}(\tau) = \sum_{r \in \mathcal{D}_p^+ \cap \mathcal{D}_q^+} \mathbf{H}_{r,i}^T(\tau)\mathbf{Q}_r(\tau)\mathbf{H}_{r,j}(\tau) + \sum_{r \in \mathcal{D}_p^+ \cap \mathcal{D}_q^+} \mathbf{K}_{r,i}^T(\tau)\mathbf{R}_r(\tau)\mathbf{K}_{r,j}(\tau)$$
$$+ \sum_{r \in \mathcal{D}_p^+} \sum_{\substack{s \in \mathcal{D}_q^+ \\ (r,s) \in \psi_i}} \left(\mathbf{A}_p(\tau)\delta_{pr} - \mathbf{B}_r(\tau)\mathbf{K}_{r,p}(\tau) \right)^T \tag{19}$$
$$\mathbf{P}_{\mathcal{D}_i^+,(r,s)}(\tau+1) \left(\mathbf{A}_q(\tau)\delta_{qs} - \mathbf{B}_s(\tau)\mathbf{K}_{s,q}(\tau) \right),$$

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 3.1, the propagation of $\mathbf{P}(\tau)$ in (19) can be computed in a distributed manner. Note that $\mathbf{P}_{\mathcal{D}_i^+,(r,s)}(\tau+1)$, inside the summation in (19), 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, S_i has to receive $\mathbf{P}_{k,(r,s)}(\tau+1)$ through communication from a system S_k, with $k \in \mathcal{D}_i^+$.

Similarly to the analogous approximation for the estimation problem, it can be shown analogously that the approximation induced by Approximation 3.1 is exact for certain topologies such as string, tree, and ring configurations (Pedroso & Batista, 2023b, Lemma 2). In such topologies, it is possible to distribute the computations of the globally synthesized controller among the computational units without
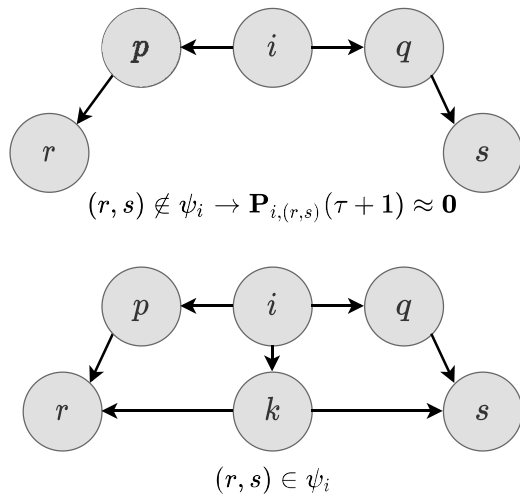
$$(r, s) \notin \psi_i \rightarrow \mathbf{P}_{i,(r,s)}(\tau + 1) \approx \mathbf{0}$$

$$(r, s) \in \psi_i$$

Fig. 1. Graphic illustration of Approximation 3.1.

disregarding any components of $\mathbf{P}(\tau+1)$. The proposed RHC algorithm, which follows from Approximation 3.1, is presented in the following result.

Theorem 3.1 (*Distributed and Decentralized RHC*). *The solution of the optimization problem* (6) *under the one-step convex relaxation* (11), *under Approximation* 3.1, *and subject to the communication, computational, and memory constraints detailed in Section* 2.3 *is given, for a time-invariant coupling topology, by the local gain computations presented in Algorithm 1.*

Proof. The decoupled computation of each local gain is portrayed by (14). Taking into account Approximation 3.1, the distributed computation of $\mathbf{P}(\tau)$ is carried out according to (19). All variables that are required for these computations in each computational unit $\mathcal{T}_i$ are available through local communication with computational units in its out-neighborhood, i.e., $\mathcal{T}_j$ for $j \in \mathcal{D}_i^+$. Under the scheduling of the computations presented in Section 3.3, it is shown in Section 3.4 that computational, and memory requirements of this algorithm satisfy the very large-scale feasibility constraints. □

Unsurprisingly, the inner loop of Algorithm 1 shares some similarities with the solution to the analogous distributed decentralized estimation problem recently proposed by the authors in Pedroso and Batista (2023b). Although this work addresses a completely different problem with intricacies that far exceed the control-estimation duality, in Pedroso and Batista (2023b, Remarks 2–4) noteworthy remarks are raised which are applicable to some implementation aspects of Algorithm 1.

3.3. Scheduling

In Section 3.2, under appropriate approximations, an algorithm was devised for the decoupled synthesis of the RHC gains over a finite-window $\{k, \dots, k+H-1\}$. 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, 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 (3). Furthermore, the decoupled gain computation in Algorithm 1 is carried out backwards 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 2.3. This issue is addressed in this section.

Denote the control discretization time by T_c, which is the sampling time of the LTV system (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 $\mathbf{K}_{i,p}(\tau), \forall p \in \mathcal{D}_i^-, \tau = k, \dots, k+d-1$ are available to $\mathcal{T}_i$. To compute the control feedback according to (3), these gains need to be available in 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 $\{k, \dots, k+H-1\}$ must start at $t = kT_c - \Delta^-$. A scheme of the proposed scheduling of Algorithm 1 over a timeline is depicted in Fig. 2, for the illustrative time interval $[kT_c; (k+d)T_c[$. In this time interval, the control input is computed according to (3), making use of the gains in the window $\{k, \dots, k+H-1\}$, which is represented in the line adjacent to the timeline that comprises $kT_c \le t < (k+d)T_c$. 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 $kT_c \le t < (k+d)T_c$, 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.

First, it is worth remarking that the proposed scheduling requires that the dynamics of a system S_i over the window $\{k, \dots, k+H-1\}$ 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 Fig. 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.

3.4. Communication, computational, and memory requirements

To lighten the notation and enable a clearer analysis of the communication, computational, and memory requirements of Algorithm 1, an homogeneous network is considered. Specifically, consider systems with the same order n_1, output dimension o_1, dimension of in-neighborhood v_1^-, and dimension of out-neighborhood v_1^+.

First, making use of the scheduling of the computations in Algorithm 1 proposed in Section 3.3, only soft real-time transmissions are required for the distributed computation of the controller gains. Second, the communication graph $\mathcal{G}_c$, i.e., the graph representation of the required directed communication links, corresponds to the tracking output coupling graph with undirected edges. Thus, system S_i can only receive information from system S_j if the tracking output of S_j is coupled with the state of S_i or the converse, i.e., $j \in \mathcal{D}_i^+ \cup \mathcal{D}_i^-$. In fact, system S_i requires the exchange of data through communication with at most $\max(v_i^-, v_i^+) - 1$ systems at each iteration. The communication complexity of system S_i is $\mathcal{O}(\max(v_i^-, v_i^+))$. Since neither v_i^- nor v_i^+ increase

---

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

Step 1: Predict: $\mathbf{A}_i(\tau), \mathbf{B}_i(\tau), \mathbf{R}_i(\tau), \tau = k, \dots, k + H - 1;$

$\qquad\qquad \mathbf{H}_{i,p}(\tau), \forall p \in \mathcal{D}_i^-, \tau = k + 1, \dots, k + H;$

$\qquad\qquad \mathbf{Q}_i(\tau), \tau = k + 1, \dots, k + H.$

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: $\tau = k + H - 1, \dots, k$

$\qquad$**Step 4.1: Transmit:** $\mathbf{R}_i(\tau), \mathbf{B}_i(\tau)$ to $\forall p \in \mathcal{D}_i^- \setminus \{i\};$

$\qquad\qquad\qquad\qquad \mathbf{Q}_p(\tau+1)^{1/2} \mathbf{H}_{p,i}(\tau+1), \forall p \in \mathcal{D}_i^+$ to $\forall q \in \mathcal{D}_i^- \setminus \{i\};$

$\qquad\qquad\qquad\quad$ **If:** $\tau \neq k$

$\qquad\qquad\qquad\qquad\quad \mathbf{Q}_i(\tau)^{1/2} \mathbf{H}_{i,p}(\tau), \forall p \in \mathcal{D}_i^-$ to $\forall p \in \mathcal{D}_i^- \setminus \{i\};$

$\qquad\qquad\qquad\quad$ **End if**

$\qquad\qquad\qquad\quad$ **If:** $\tau \neq k + H - 1$

$\qquad\qquad\qquad\qquad\quad \mathbf{R}_p(\tau+1), \mathbf{B}_p(\tau+1), \forall p \in \mathcal{D}_i^+$ to $\forall q \in \mathcal{D}_i^- \setminus \{i\};$

$\qquad\qquad\qquad\qquad\quad \mathbf{A}_i(\tau+1)$ to $\forall p \in \mathcal{D}_i^-;$

$\qquad\qquad\qquad\qquad\quad \mathbf{K}_{p,i}(\tau+1), \forall p \in \mathcal{D}_i^+$ to $\forall q \in \mathcal{D}_i^- \setminus \{i\};$

$\qquad\qquad\qquad\qquad\quad \mathbf{P}_{i,(p,q)}(\tau+1)$ for some $(p, q) \in \phi_i$ to $\forall l \in \mathcal{D}_i^- \setminus \{i\}.$

$\qquad\qquad\qquad\quad$ **End if**

$\qquad$**Step 4.2: Receive:** $\mathbf{R}_p(\tau), \mathbf{B}_p(\tau)$ from $\forall p \in \mathcal{D}_i^+ \setminus \{i\};$

$\qquad\qquad\qquad\qquad \mathbf{Q}_p(\tau+1)^{1/2} \mathbf{H}_{r,p}(\tau+1), \forall r \in \mathcal{D}_i^+$ from $\forall p \in \mathcal{D}_i^+ \setminus \{i\};$

$\qquad\qquad\qquad\quad$ **If:** $\tau \neq k$

$\qquad\qquad\qquad\qquad\quad \mathbf{Q}_p(\tau)^{1/2} \mathbf{H}_{p,i}(\tau), \forall p \in \mathcal{D}_i^+$ from $p \in \mathcal{D}_i^+ \setminus \{i\};$

$\qquad\qquad\qquad\quad$ **End if**

$\qquad\qquad\qquad\quad$ **If:** $\tau \neq k + H - 1$

$\qquad\qquad\qquad\qquad\quad \mathbf{R}_r(\tau+1), \mathbf{B}_r(\tau+1), \forall r \in \mathcal{D}_p^+$ from $\forall p \in \mathcal{D}_i^+ \setminus \{i\};$

$\qquad\qquad\qquad\qquad\quad \mathbf{A}_p(\tau+1)$ from $p \in \mathcal{D}_i^-;$

$\qquad\qquad\qquad\qquad\quad \mathbf{K}_{r,p}(\tau+1), \forall r \in \mathcal{D}_p^+$ from $\forall p \in \mathcal{D}_i^+ \setminus \{i\};$

$\qquad\qquad\qquad\qquad\quad \mathbf{P}_{p,(r,s)}(\tau+1)$ for some $(r, s) \in \phi_p$ from $\forall p \in \mathcal{D}_i^+ \setminus \{i\}.$

$\qquad\qquad\qquad\quad$ **End if**

$\qquad$**Step 4.3: Compute:**

$\qquad\qquad\qquad\quad$ **If:** $\tau = k + H - 1$

$\qquad\qquad\qquad\qquad\quad \mathbf{P}_{i,(p,q)}(\tau+1) \leftarrow \sum_{r \in \mathcal{D}_p^+ \cap \mathcal{D}_q^+} \mathbf{H}_{r,p}^T(\tau+1) \mathbf{Q}_r(\tau+1) \mathbf{H}_{r,q}(\tau+1), \forall (p, q) \in \phi_i;$

$\qquad\qquad\qquad\quad$ **Else**

$\qquad\qquad\qquad\qquad\quad$ Compute $\mathbf{P}_{i,(p,q)}(\tau+1) \forall (p, q) \in \phi_i$ making use of (19).

$\qquad\qquad\qquad\quad$ **End if**

$\qquad$**Step 4.4: Compute:**

$\qquad\qquad\qquad\quad \mathbf{S}_{p,q}(\tau) \leftarrow \mathbf{B}_p^T(\tau) \mathbf{P}_{i,(p,q)}(\tau+1) \mathbf{B}_q(\tau) + \delta_{pq} \mathbf{R}_q(\tau), \forall (p, q) \in \phi_i;$

$\qquad\qquad\qquad\quad$ Compute $\tilde{\mathbf{S}}_i(\tau)$ and $\bar{\mathbf{P}}_i(\tau+1)$ making use of (15) and (16);

$\qquad\qquad\qquad\quad \bar{\mathbf{K}}_i(\tau) \leftarrow \tilde{\mathbf{S}}_i(\tau)^{-1} \bar{\mathbf{P}}_i(\tau+1);$

$\qquad$**End for**

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

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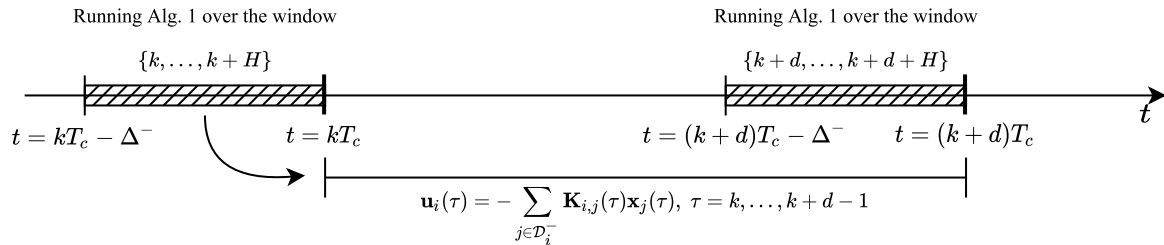

Fig. 2. Illustration of scheduling of Algorithm 1 over a timeline.

with the number of systems in the whole network, the communication complexity of each system grows with $\mathcal{O}(1)$ with N.

Third, according to Algorithm 1, each system S_i has to store in memory: (i) the matrices that model the dynamics of S_i over the window $\{k, \dots, k + H - 1\}$; (ii) the tracking output and control input weighting matrices of S_i; (iii) $\mathbf{P}_{i,(p,q)}(\tau+1)$, with $(p, q) \in \phi_i$; and (iv) the sequences of controller gains $\mathbf{K}_{i,p}(\tau), \forall p \in \mathcal{D}_i^-, \tau = k, \dots, k + d - 1$ and

$\mathbf{K}_{p,i}(\tau), \forall p \in \mathcal{D}_i^+, \tau = k, \dots, k + H - 1$. The data storage complexity of these is: (i) $\mathcal{O}\left(\max\left(n_1, m_1, v_1^- o_1\right) nH\right)$; (ii) $\mathcal{O}\left(\max\left(m_1^2, o_1^2\right) H\right)$; (iii) $\mathcal{O}((v_1^+)^2 n_1^2)$; and (iv) $\mathcal{O}(m_1 n_1 \max(v_1^+, v_1^-) H)$, respectively. Even though some auxiliary variables are required to be handled at each iteration, their memory footprint is lower than the previously mentioned variables. Hence, the data storage complexity of each system grows with $\mathcal{O}(1)$ with N.

Fourth, 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 $\mathcal{O}((v_1^{-})^4 n_1^3)$ floating-point operations in total, which grows with $\mathcal{O}(1)$ with N.

To conclude, Algorithm 1 abides by the communication, computational, and memory constraints detailed in Section 2.3. In Pedroso and Batista (2023a) a decentralized RHC solution is also devised for networks of systems with LTV dynamics. However, using the method proposed therein, the global synthesis would have to be replicated in each computational unit, which violates Constraints 2, 3, and 4 put forward in Section 2.3. Therefore, the proposed solution makes great strides towards the implementation feasibility of decentralized control solutions over very large-scale networks.

3.5. 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. The distributed and decentralized control solution is now extended to accommodate a time-varying tracking output coupling topology. In that regard, consider a time-varying directed graph $\mathcal{G}(k)$, with time-varying in-degree $v_i^{-}(k)$, in-neighborhood $\mathcal{D}_i^{-}(k)$, out-degree $v_i^{+}(k)$, out-neighborhood $\mathcal{D}_i^{+}(k)$, and define $\phi_i(k)$ and $\psi_i(k)$ analogously to (17) and (18), respectively, for system S_i.

Algorithm 2, proposed in Appendix, is the extension of Algorithm 1 to a time-varying tracking output coupling topology. Even though such extension is quite straightforward as far as distributing the gain synthesis across the systems is concerned, that is not the case for the scheduling of the computations over time. Recall the scheduling procedure proposed in Section 3.3. 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 S_i receives data through communication from S_p, with $p \in \mathcal{D}_i^{+}(k + H)$. That is, at $t = kT_c - \Delta^{-}$ communication between S_i and a system with which 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 $HT_c + \Delta^{-}$ may later 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 3.3 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.

4. Application to onboard orbit control of LEO mega-constellations

In this section, the distributed decentralized RHC algorithm developed in Section 3 is applied to the cooperative orbit control problem of LEO mega-constellations. The scheme presented in this section is novel and it is developed aiming for efficiency and fuel saving in a distributed and decentralized framework. Unlike other approaches in the literature, it follows the crucial constraints detailed in Section 2.3 for a feasible real-time implementation in a very large-scale. The source code of the numerical simulations is available in the DECENTER Toolbox at https://decenter2021.github.io/examples/DDRHCStarlink/.

4.1. Mega-constellation model

Consider a Walker constellation $\bar{i} : T/P/F$. It has a total of T satellites, evenly distributed over P orbital planes at a nominal inclination $\bar{i}$, whose nominal relative phasing between adjacent planes is $\bar{\beta} = 2\pi F/T$, where F is the phasing parameter. 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, S_j, each associated with a computational unit $\mathcal{T}_j$, with $j = 1, \ldots, T$. Let $\mathbf{p}_i \in \mathbb{R}^3$ and $\mathbf{v}_i \in \mathbb{R}^3$ denote the position and velocity vectors, respectively, of 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 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 S_i is, thus, given by

$$\begin{cases} \dot{\mathbf{p}}_i = \mathbf{v}_i \\ \dot{\mathbf{v}}_i = -\mu \mathbf{p}_i / \|\mathbf{p}_i\|_2^3 + \mathbf{a}_i^{J_2} + \mathbf{a}_i^P + \mathbf{R}_{TNW}^{ECI} \mathbf{u}_i / m_i \\ \dot{m}_i = -\|\mathbf{u}_i\|_1 / (I_i^{sp} g_0), \end{cases} \quad (20)$$

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 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 S_i, made up of the aforementioned six non-singular mean orbital elements, by

$$\mathbf{x}_i(t) = \begin{bmatrix} a_i(t) & u_i(t) & e_{xi}(t) & e_{yi}(t) & i_i(t) & \Omega_i(t) \end{bmatrix}^T.$$

The satellite orbital mechanics (20) are nonlinear and, thus, have to be linearized to employ the distributed and decentralized method put forward in this paper. 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 4.2.

The nominal state of S_i at time instant t, $\bar{\mathbf{x}}_i(t) = \begin{bmatrix} \bar{a}_i(t) & \bar{u}_i(t) & \bar{e}_{xi}(t) & \bar{e}_{yi}(t) & \bar{i}_i(t) & \bar{\Omega}_i(t) \end{bmatrix}^T$, is defined by

$$
\begin{cases}
\bar{a}_i(t) = \bar{a} \\
\bar{u}_i(t) = \bar{u}_{t_0} + ((i-1) \bmod T/P) 2\pi P/T \\
\qquad\quad + \lfloor (i-1)P/T \rfloor 2\pi F/T + (\dot{M} + \dot{\omega})(t - t_0) \\
\bar{e}_{x,i}(t) = 0 \\
\bar{e}_{y,i}(t) = 0 \\
\bar{i}_i(t) = \bar{i} \\
\bar{\Omega}_i(t) = \bar{\Omega}_{t_0} + \lfloor (i-1)P/T \rfloor 2\pi/P + \dot{\Omega}(t - t_0).
\end{cases}
\tag{21}
$$

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 Vallado (1997, 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 (21) 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 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

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

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

$$
\begin{cases}
\bar{u}_{t_0} = \sum_{i=1}^{T} \left(u_i(t_0) - ((i-1) \bmod T/P) 2\pi P/T \right. \\
\qquad\qquad \left. - \lfloor (i-1)P/T \rfloor 2\pi F/T \right) \\
\bar{\Omega}_{t_0} = \sum_{i=1}^{T} \left(\Omega_i(t_0) - \lfloor (i-1)P/T \rfloor 2\pi/P \right).
\end{cases}
\tag{22}
$$

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 (Jakovetić et al., 2014). 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 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 D'Amico (2010), which is defined as

$$
\delta\mathbf{x}_i(t) := \begin{bmatrix}
a_i(t)/\bar{a}_i(t) - 1 \\
u_i(t) - \bar{u}_i(t) + \left(\Omega_i(t) - \bar{\Omega}_i(t) \right) \cos \bar{i}_i(t) \\
e_{x,i}(t) - \bar{e}_{x,i}(t) \\
e_{y,i}(t) - \bar{e}_{y,i}(t) \\
i_i(t) - \bar{i}_i(t) \\
\left(\Omega_i(t) - \bar{\Omega}_i(t) \right) \sin \bar{i}_i(t)
\end{bmatrix}.
\tag{23}
$$

This set parameterizes the position of the satellite, $\mathbf{x}_i(t)$, in relation to its nominal position, $\bar{\mathbf{x}}_i(t)$. In Di Mauro et al. (2018), Koenig et al. (2016) and Sullivan et al. (2016), the dynamics (20), 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

$$
\delta\mathbf{x}_i((k+1)T_c) = \mathbf{A}_i(k)\delta\mathbf{x}_i(kT_c) + \mathbf{B}_i(k)\mathbf{u}_i(kT_c)/m_i(kT_c)
\tag{24}
$$

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 Di Mauro et al. (2018), by

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

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

$$
\mathbf{B}_i(k) = \begin{bmatrix}
\frac{2\Delta u}{\bar{n}\bar{a}\bar{W}} & 0 & 0 \\
-\frac{\bar{A}\Delta u^2}{\bar{n}\bar{a}\bar{W}^2} & \frac{2\Delta u}{\bar{n}\bar{a}\bar{W}} & \Psi_{2,3} \\
2\Psi_{4,2} & \Psi_{3,2} & 0 \\
-2\Psi_{3,2} & \Psi_{4,2} & 0 \\
0 & 0 & \Psi_{5,3} \\
\frac{7}{2} \frac{\bar{K}\Delta u^2 \sin 2\bar{i}}{\bar{n}\bar{a}\bar{W}^2} & 0 & \Psi_{6,3}
\end{bmatrix},
\tag{25}
$$

where

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

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

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

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

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

$$\Delta u := \bar{W} T_c,$$
$$\Delta \omega := \bar{K}(5 \cos^2 \bar{\imath} - 1) T_c,$$
$$\bar{K} := (3/4) \bar{n} R_\oplus^2 J_2 / \bar{a}^2,$$
$$\bar{\Lambda} := (3/2) \bar{n} + (7/2) \bar{K}(3 \cos^2 \bar{\imath} - 1),$$
$$\bar{W} := \bar{n} + \bar{K}(8 \cos^2 \bar{\imath} - 2), \quad \text{and}$$
$$\bar{C} := \bar{K}(5 \cos^2 \bar{\imath} - 1) / \bar{W}.$$

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.

4.2. Controller implementation

Now that the constellation has been modeled as a LTV system, the constellation orbit control problem has to be formulated as a RHC problem. A common approach is the bounding box method. 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.

In this paper, in an attempt to reduce fuel consumption and to follow the communication, computational, and memory constraints detailed in Section 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

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

is considered for each satellite 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 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., $\|\mathbf{p}_i - \mathbf{p}_j\| \le R$, up to a maximum of $|D^-|_{\max}$ satellites in D_i^-. If more than $|D^-|_{\max} - 1$ satellites other than S_i are within a tracking range of S_i, only the $|D^-|_{\max} - 1$ 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 $D_i^- \setminus \{i\} = \left\{ j_1^i, \dots, j_{|D_i^-|-1}^i \right\}$. Then the relative tracking output component is given by

$$\mathbf{z}_{i,rel}(k) := \text{col}\left(\mathbf{z}_{i,j_1^i}^{ref}(k), \dots, \mathbf{z}_{i,j_{|D_i^-|-1}^i}^{ref}(k) \right)$$

with

$$\mathbf{z}_{i,j}^{ref}(k) := \begin{bmatrix} u_i(k) - u_j(k) - \left(\bar{u}_i(k) - \bar{u}_j(k) \right) \\ \Omega_i(k) - \Omega_j(k) - \left(\bar{\Omega}_i(k) - \bar{\Omega}_j(k) \right) \end{bmatrix}$$
$$= \begin{bmatrix} \delta u_i(k) - \delta u_j(k) - \left(\delta \Omega_i(k) - \delta \Omega_j(k) \right) / \tan \bar{\imath} \\ \left(\delta \Omega_i(k) - \delta \Omega_j(k) \right) / \sin \bar{\imath} \end{bmatrix}. \tag{26}$$

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 (26) makes use of the nominal constellation just to retrieve the nominal angular spacing in u and Ω between S_i and 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 S_i as $\mathbf{z}_i(k) := \text{col}(\mathbf{z}_{i,rel}(k), \mathbf{z}_{i,in}(k))$, it can be written as

$$\mathbf{z}_i(k) = \sum_{p \in D_i^-} \mathbf{H}_{i,p}(k) \delta \mathbf{x}_p(k) \tag{27}$$

with

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

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

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

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

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

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 (24) and the constellation orbit control problem is formulated as the regulation of the tracking output (27). Thus, we are in the conditions of applying the distributed and decentralized RHC method put forward in Section 3, which abides by the communication, computational, and memory feasibility constraints detailed in Section 2.3. The tracking output coupling graph $\mathcal{G}(k)$ is time-varying, thus one has to follow Algorithm 2. Considering the convolution matrix as defined in (25), one obtains the feedback law

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

However, more often than not, the relative mean orbital elements are not readily available to each satellite from onboard 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 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 (23), 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

Table 1
Parameters of the constellation.

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

Fig. 3. Number of satellites within ISL range at 0 TDB seconds since J2000.

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 Spiridonova et al. (2014) 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 available in the *osculating2mean* toolbox at https://github.com/decenter2021/osculating2mean, which is based on Eckstein and Hechler (1970), Hwang (2001), Hwang and Hwang (2002) and Kaula (1966). In the simulation results in the sequel, the Earth's gravity field EGM96 spherical harmonic expansion (Lemoine et al., 1998) up to degree and order 12 is employed.

4.3. 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 paper. 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 deg : 1584/72/17. The phasing parameter of this Starlink shell is chosen so that the minimum distance between two satellites is maximized (Liang et al., 2021). Table 1 presents the parameters that characterize this shell. All satellites are assumed to be identical.

Fig. 3 depicts the minimum, maximum, and average number of satellites within tracking output coupling range among all satellites as a function of R, at 0 Dynamical Barycentric Time (TDB) seconds since J2000. A tracking output coupling range of $R = 750$ km is considered, which enables all satellites to establish a tracking output coupling with, at least, another satellite at any time. The maximum in-neighborhood cardinality is set to $|D^-|_{max} = 6$, which allows for each satellite to establish tracking output couplings with up to 5 other satellites. In Fig. 4, 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 $T_c = 10$ 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
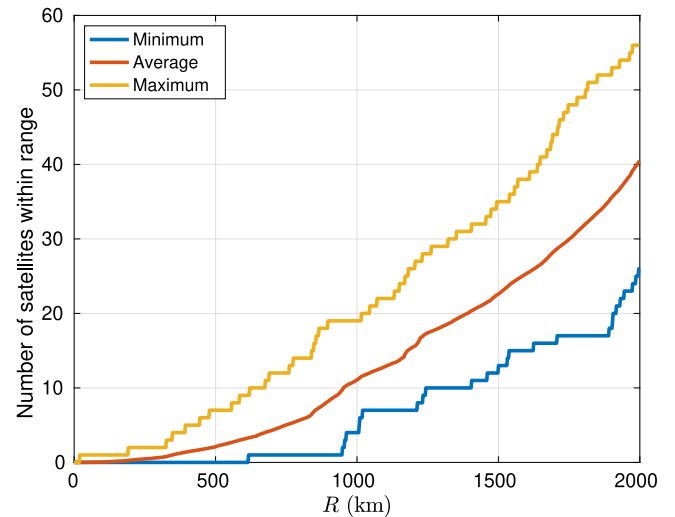
update frequency is attainable by the Hall effect thrusters. The tracking output coupling topology varies greatly with time, thus the parameters H and d of the scheduling of the RHC distributed and decentralized algorithm have to be tuned thoughtfully, according to the limitations and communication requirements pointed out in Section 3.5. The interval of time between allowed communications is set to $T_t = 1$ 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 $R_{LOS} = 2\sqrt{\bar{a}^2 - (R_\oplus + 80 \text{ km})^2} = 5{,}014$ km (Bhattacherjee & Singla, 2019). This range is supported by an ISL system, since multiple laser ISLs of up to 4900 km have been reported between LEO satellites since 2008 (Sodnik et al., 2010). 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. These procedure is repeated several times. Fig. 5 depicts an histogram of the interval of time that the pairs of satellites remained in line-of-sight range.

Consider the computation of the RHC gains over a generic window $\{k, \dots, k + H - 1\}$, according to the scheduling solution proposed in Section 3.5. 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 (A.1) of each satellite would not contain any other satellites due to communication constraints. Notice that, in Fig. 5, no pair of satellites remained in line-of-sight range for more than $\Delta t_{max} = 1320$ s. Therefore, the constraint

$$(H + 2)T_t + HT_c < \Delta t_{max} \tag{28}$$

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 3.5. 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
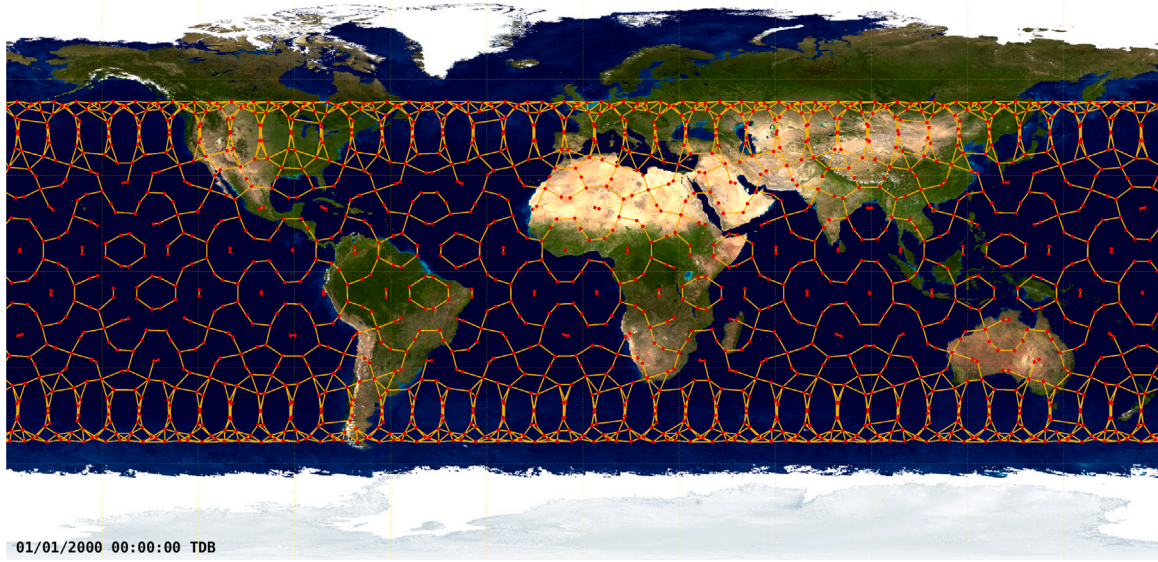
Fig. 4. Snapshot of ground track and ISL of the simulated constellation at 0 TDB seconds since J2000.
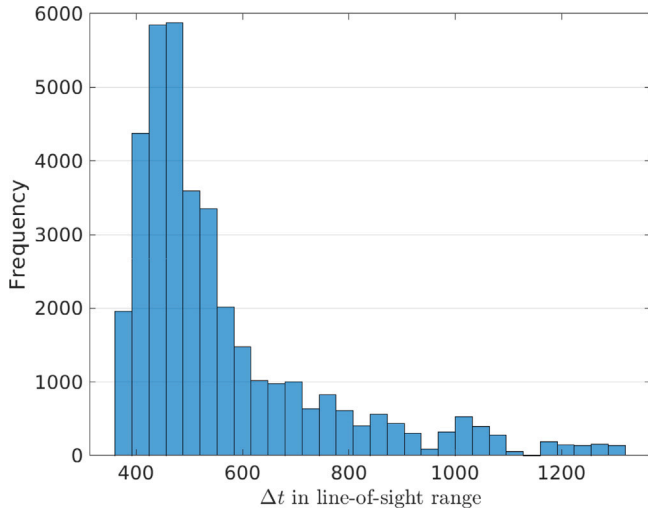

Fig. 5. Intervals of time two satellites are in line-of-sight range before establishing a tracking coupling.

least, since t_s. From Fig. 5, the minimum time for the maintenance of line-of-sight is $\Delta t_{min} = 360$ s. Thus, the constraint

$$(d+1)T_t + T_c d < \Delta t_{min}. \tag{29}$$

arises from this observation. After algebraic manipulation of (28) and (29) and considering non-overlapping computation windows, as described in Section 3.3, 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 to physically meaningful orders of magnitude and set to $\mathbf{Q}_i^{rel}(k) = (1/10^{-4})^2 \mathbf{I}$, $\mathbf{Q}_i^{in}(k) = \text{diag}(1/(\bar{a}10^{-4})^2, 1/(0.5 \times 10^{-2})^2 \mathbf{I}_2, 1/(10^{-2})^2)$, and $\mathbf{R}_i(k) = (1/C_{t1})^2 \mathbf{I}_2$.

4.4. Simulation results

In this section, the simulation results are presented for the aforementioned illustrative mega-constellation. The numerical simulation was carried out employing the high-fidelity TU Delft's Astrodynamic Toolbox[1] (TUDAT) (Kumar et al., 2012). NASA's SPICE ephemerides are used for the orbit propagation and the following perturbations are taken into account:

1. Earth's gravity field EGM96 spherical harmonic expansion (Lemoine et al., 1998) up to degree and order 24;
2. Atmospheric drag NRLMSISE-00 model (Picone et al., 2002), assuming constant drag coefficient and section area;
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. The *tudat-matlab-thrust-feedback* package, available at https://github.com/decenter2021/tudat-matlab-thrust-feedback, is employed to implement the thrust feedback interface between TUDAT and MATLAB. A scheme of the simulation environment is depicted in Fig. 6.

The total number of states in this shell of the mega-constellation amounts to $6 \times 1584 = 9504$, which is the dimension of the global system of an equivalent centralized framework. Hence, implementing a centralized control algorithm in real-time would require the manipulation of very high-dimensional matrices. For example, $\mathbf{P}(\tau)$, which is not typically sparse, would occupy 722.6×10^6 bytes in double precision. Furthermore, it would have to feature all-to-all communication of substantial data volumes over great distances via the MCC, which requires several ground stations scattered throughout the planet. Therefore, the extreme computational demands in real-time and vast communication requirements make an equivalent centralized framework entirely unfeasible. On top of that, it is also insightful to compare
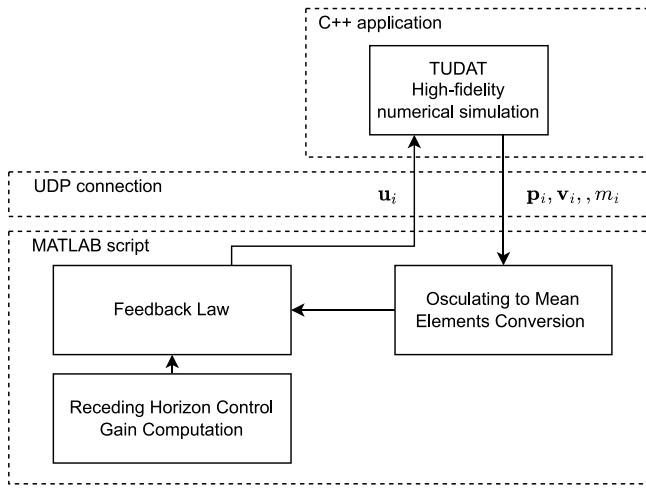
Fig. 6. Scheme of the simulation environment.
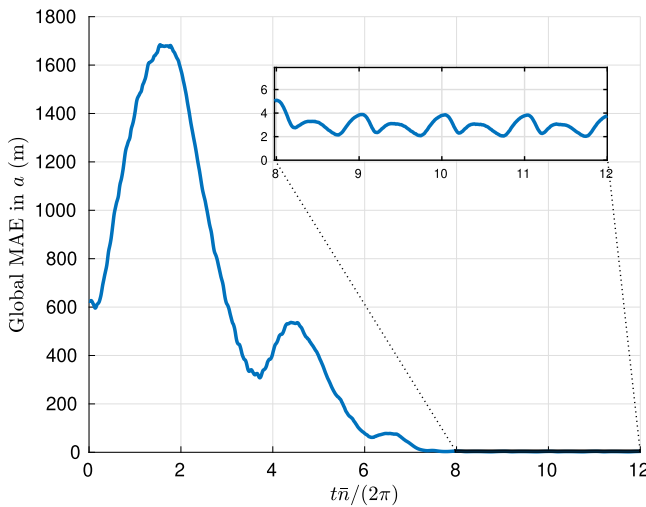

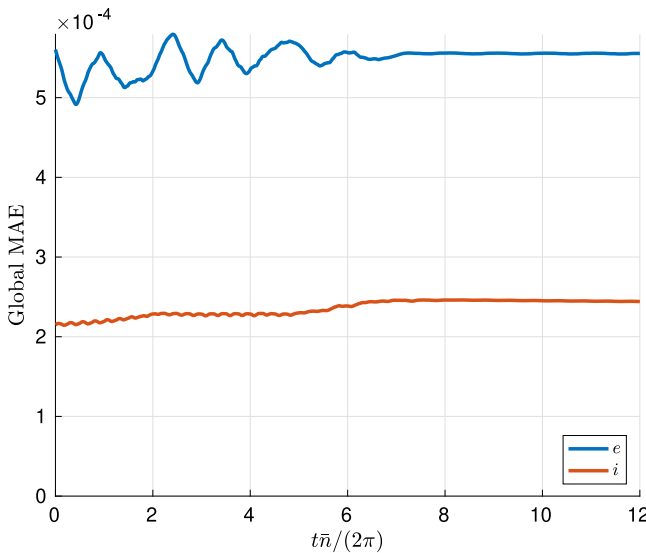**Fig. 7.** Evolution of the MAE in the semi-major axis.

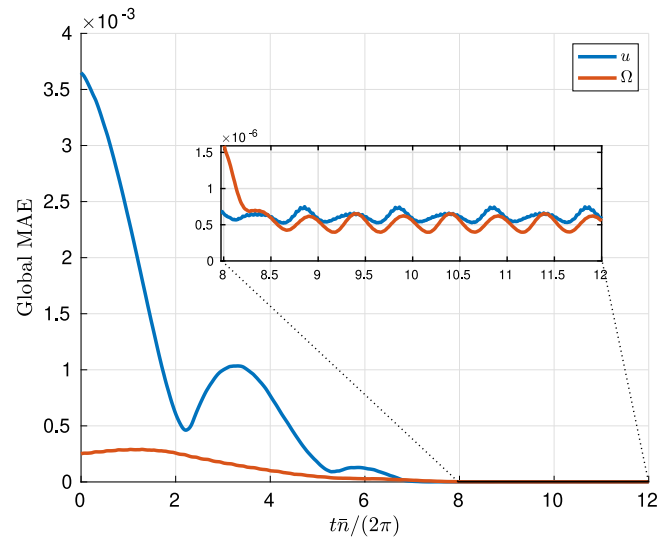**Fig. 8.** Evolution of the MAE in eccentricity and inclination.

Fig. 9. Evolution of the MAE in mean argument of latitude and longitude of ascending node.

the resource usage of the proposed solution in relation to state-of-the-art decentralized solutions that could, in principle, be used to solve the mega-constellation maintenance problem. First, the decentralized methods that rely on a global synthesis procedure such as Farhood et al. (2015) and Pedroso and Batista (2023a) are required to solve LMIs and perform several algebraic manipulations, respectively, with global matrices of the system dynamics at each discrete-time control instant. However, due to the sheer size of this system, that is unfeasible in a standard computational server. In fact, from the computational analysis of an efficient implementation of the latter method, performed in Pedroso and Batista (2021), one would estimate that each RHC window would take $H \times 3623.7 \times 10^{-7} \times 9504^{2.770} \approx 3.8 \times 10^9$ s to be computed. Second, a control design procedure analogous to Luft et al. (2018), whose memory requirements grow with the dimension of the system, would require each individual system to store, just in components of $\mathbf{P}(\tau)$, $1584 \times 6^2 \times 8 \approx 45.6 \times 10^4$ bytes in double precision. The proposed solution only requires $|D^-|^2_{\max} \times 6^2 \times 8 \approx 1.04 \times 10^4$ bytes in double precision. This reduction of one order of magnitude is especially relevant in stringent settings such satellite design. Moreover, using the design method of Luft et al. (2018), this gap increases significantly as one considers more shells and the memory requirements of each single satellite increase as the number of satellites in the network increases.

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 (22). 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 Figs. 7 and 8. 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 (22), 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 (21) 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 Fig. 9. The steady-state MAE, obtained by averaging the MAE of the last three orbital periods of the simulation, is depicted in Table 2.
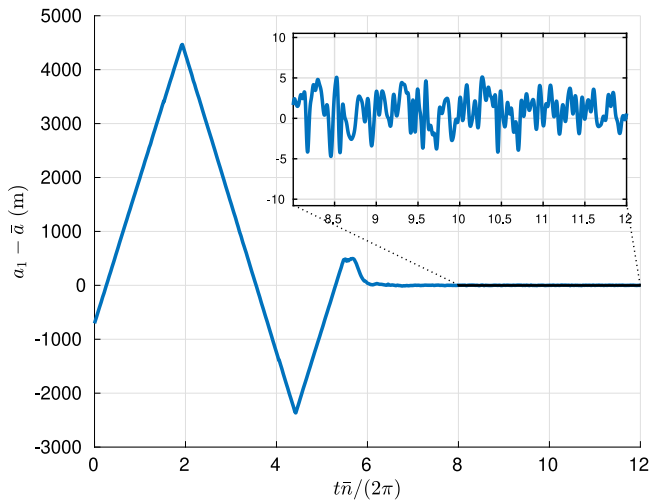
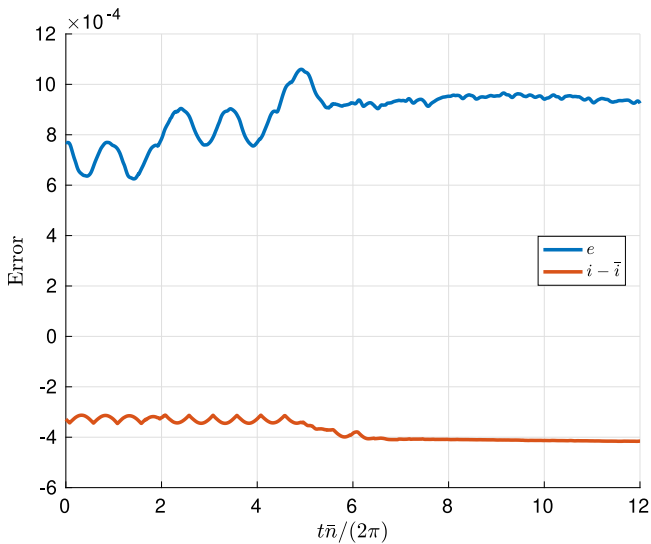**Fig. 10.** Evolution of the semi-major axis tracking error, for satellite 1.

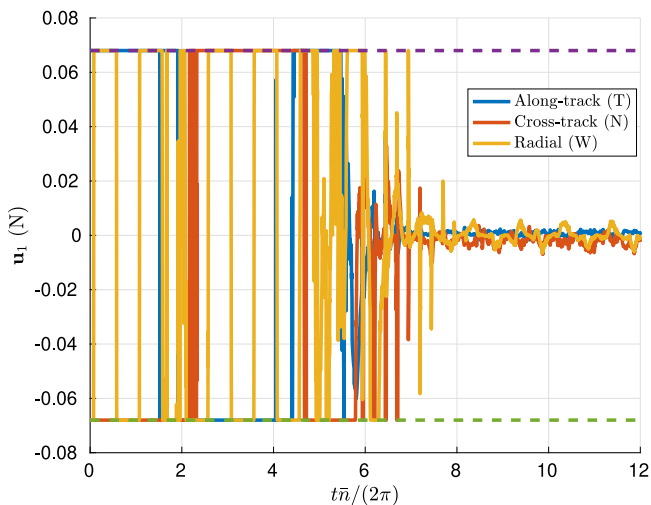**Fig. 11.** Evolution of the eccentricity and inclination tracking errors, for satellite 1.

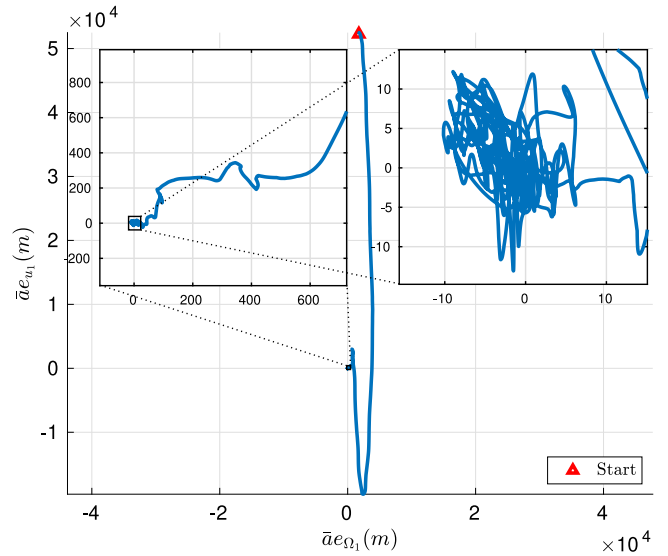**Fig. 12.** Evolution of the components of the control input, for satellite 1.

Fig. 13. Trajectory of the mean argument of latitude and longitude of ascending node relative tracking errors, for satellite 1.

Table 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 m	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 paper 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 highly nonlinear dynamics and realistic disturbances. Indeed, the proposed solution is robust to uncertainty, disturbances, and model-reality mismatch. 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 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. Figs. 10 and 11 show the evolution of the semi-major axis, eccentricity, and inclination tracking errors and Fig. 12 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. Fig. 13 shows the trajectory of the mean argument of latitude and longitude of ascending node relative tracking errors. It is very interesting to remark that, even 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 Fig. 13. Given the very large initial error, the control input is initially saturated, as depicted in Fig. 12. Afterwards, when the relative error is driven to close to zero, the control input is far from the saturation limits. Despite the fact that the proposed method does not have any stability guarantees regarding the saturated inputs, no stability issues arise in the numeric simulation.

5. Conclusion

Applications of very large-scale networks of dynamically decoupled systems have been emerging in the fields of swarm robotics and networked control, many of which have nonlinear dynamics. The feasible implementation of control algorithms over these networks comes with

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

Step 1: Predict: $\tilde{D}_i^-(\tau), \tilde{D}_i^-(\tau), \tau = k, \ldots, k + H$, according to (A.1);

$\qquad\qquad$ $\mathbf{A}_i(\tau), \mathbf{B}_i(\tau), \mathbf{R}_i(\tau), \tau = k, \ldots, k + H - 1$;

$\qquad\qquad$ $\mathbf{H}_{i,p}(\tau), \forall p \in \tilde{D}_i^-(\tau), \tau = k + 1, \ldots, k + H$;

$\qquad\qquad$ $\mathbf{Q}_i(\tau), \tau = k + 1, \ldots, k + H$.

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

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

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

$\qquad$ **Step 4.1: Transmit:** $\mathbf{R}_i(\tau), \mathbf{B}_i(\tau)$ to $\forall p \in \tilde{D}_i^-(\tau) \setminus \{i\}$;

$\qquad\qquad\qquad\qquad$ $\mathbf{Q}_p(\tau + 1)^{1/2}\mathbf{H}_{p,i}(\tau + 1), \forall p \in \tilde{D}_i^+(\tau + 1)$ to $\forall q \in \tilde{D}_i^-(\tau) \setminus \{i\}$;

$\qquad\qquad\qquad\qquad$ **If:** $\tau \neq k$

$\qquad\qquad\qquad\qquad\qquad$ $\mathbf{Q}_i(\tau)^{1/2}\mathbf{H}_{i,p}(\tau), \forall p \in \tilde{D}_i^-(\tau)$ to $\forall p \in \tilde{D}_i^-(\tau) \setminus \{i\}$;

$\qquad\qquad\qquad\qquad$ **End if**

$\qquad\qquad\qquad\qquad$ **If:** $\tau \neq k + H - 1$

$\qquad\qquad\qquad\qquad\qquad$ $\mathbf{R}_p(\tau + 1), \mathbf{B}_p(\tau + 1), \forall p \in \tilde{D}_i^+(\tau + 1)$ to $\forall q \in \tilde{D}_i^-(\tau) \setminus \{i\}$;

$\qquad\qquad\qquad\qquad\qquad$ $\mathbf{A}_i(\tau + 1)$ to $\forall p \in \tilde{D}_i^-(\tau)$;

$\qquad\qquad\qquad\qquad\qquad$ $\mathbf{K}_{p,i}(\tau + 1), \forall p \in \tilde{D}_i^+(\tau + 1)$ to $\forall q \in \tilde{D}_i^-(\tau) \setminus \{i\}$;

$\qquad\qquad\qquad\qquad\qquad$ $\mathbf{P}_{i,(p,q)}(\tau + 1)$ for some $(p, q) \in \phi_i(\tau + 1)$ to $\forall l \in \tilde{D}_i^-(\tau) \setminus \{i\}$.

$\qquad\qquad\qquad\qquad$ **End if**

$\qquad$ **Step 4.2: Receive:** $\mathbf{R}_p(\tau), \mathbf{B}_p(\tau)$ from $\forall p \in \tilde{D}_i^+(\tau) \setminus \{i\}$;

$\qquad\qquad\qquad\qquad$ $\mathbf{Q}_p(\tau + 1)^{1/2}\mathbf{H}_{r,p}(\tau + 1), \forall r \in \tilde{D}_p^+(\tau + 1)$ from $\forall p \in \tilde{D}_i^+(\tau) \setminus \{i\}$;

$\qquad\qquad\qquad\qquad$ **If:** $\tau \neq k$

$\qquad\qquad\qquad\qquad\qquad$ $\mathbf{Q}_p(\tau)^{1/2}\mathbf{H}_{p,i}(\tau), \forall p \in \tilde{D}_i^+(\tau)$ from $p \in \tilde{D}_i^+(\tau) \setminus \{i\}$;

$\qquad\qquad\qquad\qquad$ **End if**

$\qquad\qquad\qquad\qquad$ **If:** $\tau \neq k + H - 1$

$\qquad\qquad\qquad\qquad\qquad$ $\mathbf{R}_r(\tau + 1), \mathbf{B}_r(\tau + 1), \forall r \in \tilde{D}_p^+(\tau + 1)$ from $\forall p \in \tilde{D}_i^+(\tau) \setminus \{i\}$;

$\qquad\qquad\qquad\qquad\qquad$ $\mathbf{A}_p(\tau + 1)$ from $p \in \tilde{D}_i^-(\tau)$;

$\qquad\qquad\qquad\qquad\qquad$ $\mathbf{K}_{r,p}(\tau + 1), \forall r \in \tilde{D}_p^+(\tau + 1)$ from $\forall p \in \tilde{D}_i^+(\tau) \setminus \{i\}$;

$\qquad\qquad\qquad\qquad\qquad$ $\mathbf{P}_{p,(r,s)}(\tau + 1)$ for some $(r, s) \in \tilde{\phi}_p(\tau + 1)$ from $\forall p \in \tilde{D}_i^+(\tau) \setminus \{i\}$.

$\qquad\qquad\qquad\qquad$ **End if**

$\qquad$ **Step 4.3: Compute:**

$\qquad\qquad\qquad$ **If:** $\tau = k + H - 1$

$$\mathbf{P}_{i,(p,q)}(\tau + 1) \leftarrow \sum_{r \in \tilde{D}_p^+(\tau+1) \cap \tilde{D}_q^+(\tau+1)} \mathbf{H}_{r,p}^T(\tau + 1)\mathbf{Q}_r(\tau + 1)\mathbf{H}_{r,q}(\tau + 1), \ \forall(p, q) \in \tilde{\phi}_i(\tau);$$

$\qquad\qquad\qquad$ **Else**

$$\mathbf{P}_{i,(p,q)}(\tau + 1) \leftarrow \sum_{r \in \tilde{D}_p^+(\tau+1) \cap \tilde{D}_q^+(\tau+1)} \left(\mathbf{H}_{r,i}^T(\tau + 1)\mathbf{Q}_r(\tau + 1)\mathbf{H}_{r,j}(\tau + 1) + \mathbf{K}_{r,i}^T(\tau + 1)\mathbf{R}_r(\tau + 1)\mathbf{K}_{r,j}(\tau + 1) \right)$$
$$+ \tilde{\mathbf{W}}_p(k + 1)\tilde{\mathbf{P}}_{\tilde{D}_i^+(\tau+1)}^{(p,q)}(\tau + 2)\tilde{\mathbf{W}}_q^T(k), \ \forall(p, q) \in \tilde{\phi}_i(\tau).$$

$\qquad\qquad\qquad$ **End if**

$\qquad$ **Step 4.4: Compute:**

$\qquad\qquad\qquad$ $\mathbf{S}_{p,q}(\tau) \leftarrow \mathbf{B}_p^T(\tau)\mathbf{P}_{i,(p,q)}(\tau + 1)\mathbf{B}_q(\tau) + \delta_{pq}\mathbf{R}_p(\tau), \forall(p, q) \in \tilde{\phi}_i(\tau)$;

$\qquad\qquad\qquad$ Compute $\tilde{\mathbf{S}}_i(\tau)$ and $\tilde{\mathbf{P}}_i(\tau + 1)$ making use of (15) and (16);

$\qquad\qquad\qquad$ $\tilde{\mathbf{K}}_i(\tau) \leftarrow \tilde{\mathbf{S}}_i(\tau)^{-1}\tilde{\mathbf{P}}_i(\tau + 1)$;

$\qquad$ **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 $\forall p \in \mathcal{D}_i^-(\tau) \setminus \{i\}$.

---

pressing challenges that arise from the scale of the network, which impose limiting constraints regarding communication, computational power, and memory. State-of-the-art receding horizon control solutions fail to meet such constraints. In this paper, a decentralized receding horizon control solution befitting these challenges is proposed. This algorithm is devised to account for communication and computation delays, only requires local communication and ensures that the computational and memory requirements in each system do not scale with the dimension of the network. As a means of assessing the performance of the algorithm, it is applied to the orbit control problem of low Earth orbit mega-constellations. The proposed method shows promising performance for the orbit control problem of a shell of the Starlink mega-constellation. Future work should focus on devising an approximation to enable a distributed gain synthesis that preserves consistency and on obtaining stability guarantees.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Code availability statement

The data that support the findings of this study are openly available in the DECENTER toolbox at https://decenter2021.github.io/examples/DDRHCStarlink.

Appendix

The extension of Algorithm 1 to a time-varying tracking output coupling topology is shown in Algorithm 2. 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 $C_i(\tau)$ as the set of systems with which system 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.,

$$\tilde{\mathcal{D}}_i^{\pm}(\tau) := \mathcal{D}_i^{\pm}(\tau) \cap C_i(\tau), \tag{A.1}$$

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 (3), no restrictions can be enforced, i.e., $\mathcal{D}_i^{\pm}(\tau) \subseteq C_i(\tau)$. Therefore, d should be sufficiently small to allow for that. In Section 4, 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.

References

Bemporad, A., & Barcelli, D. (2010). Decentralized model predictive control. *Networked Control Systems*, 149–178.

Bemporad, A., & Rocchi, C. (2011). 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* (pp. 7488–7493). IEEE.

Bereg, S., Díaz-Báñez, J. M., Lopez, M. A., Rozario, T., & Valavanis, K. (2015). A decentralized geometric approach for the formation keeping in unmanned aircraft navigation. In *2015 international conference on unmanned aircraft systems (ICUAS)* (pp. 989–997).

Bhattacherjee, D., & Singla, A. (2019). Network topology design at 27,000 km/hour. In *Proceedings of the 15th international conference on emerging networking experiments and technologies* (pp. 341–354).

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

Carrillo-Arce, L. C., Nerurkar, E. D., Gordillo, J. L., & Roumeliotis, S. I. (2013). Decentralized multi-robot cooperative localization using covariance intersection. In *2013 IEEE/RSJ international conference on intelligent robots and systems* (pp. 1412–1417). IEEE.

Dai, M., Mu, H., & Wu, M. (2016). 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.

D'Amico, S. (2010). *Autonomous formation flying in low earth orbit* (Ph.D. thesis), TU Delft.

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

Dunbar, W. B., & Murray, R. M. (2004). *Distributed receding horizon control with application to multi-vehicle formation stabilization*: *Technical Report*, California Institute of Technology Passadena,Control and Dynamical Systems.

Eckstein, M., & Hechler, F. (1970). A reliable derivation of the perturbations due to any zonal and tesseral harmonics of the geopotential for nearly-circular satellite orbits.

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

Gavish, B. (1997). LEO/MEO systems–Global mobile communication systems. *Telecommunication Systems*, *8*(2), 99–141.

Hwang, C. (2001). Gravity recovery using COSMIC GPS data: application of orbital perturbation theory. *Journal of Geodesy*, *75*(2), 117–136.

Hwang, C., & Hwang, L.-S. (2002). Satellite orbit error due to geopotential model error using perturbation theory: applications to ROCSAT-2 and COSMIC missions. *Computers & Geosciences*, *28*(3), 357–367.

Ivanov, D., Monakhova, U., & Ovchinnikov, M. (2019). Nanosatellites swarm deployment using decentralized differential drag-based control with communicational constraints. *Acta Astronautica*, *159*, 646–657.

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

Kaula, W. M. (1966). *Theory of satellite geodesy: Applications of satellites to geodesy*. Waltham, MA: Blaisdell Publ. Company.

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

Koenig, A. W., Guffanti, T., & D'Amico, S. (2016). New state transition matrices for relative motion of spacecraft formations in perturbed orbits. In *AIAA/AAS astrodynamics specialist conference* (p. 5635).

Kumar, K., van Barneveld, P., Dirkx, D., Melman, J., Mooij, E., & Noomen, R. (2012). Tudat: a modular and robust astrodynamics toolbox. In *5th ICATT conference* (pp. 1–8). ESA.

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

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

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

Liang, J., Chaudhry, A. U., & Yanikomeroglu, H. (2021). Phasing parameter analysis for satellite collision avoidance in starlink and kuiper constellations. In *2021 IEEE 4th 5G world forum (5GWF)* (pp. 493–498). IEEE.

Luft, L., Schubert, T., Roumeliotis, S. I., & Burgard, W. (2016). Recursive decentralized collaborative localization for sparsely communicating robots. In *Robotics: Science and systems*. New York.

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

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

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

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

Pedroso, L., & Batista, P. (2023a). Discrete-time decentralized linear quadratic control for linear time-varying systems. *International Journal of Robust and Nonlinear Control*, *33*(1), 67–101. http://dx.doi.org/10.1002/rnc.5772.

Pedroso, L., & Batista, P. (2023b). Distributed decentralized EKF for very large-scale networks with application to satellite mega-constellations navigation. *Control Engineering Practice*, *135*, Article 105509. http://dx.doi.org/10.1016/j.conengprac.2023.105509.

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

Richards, A., & How, J. (2004a). A decentralized algorithm for robust constrained model predictive control. In *Proceedings of the 2004 American control conference, Vol. 5* (pp. 4261–4266). IEEE.

Richards, A., & How, J. (2004b). Decentralized model predictive control of cooperating UAVs. In *2004 43rd IEEE conference on decision and control (CDC)(IEEE Cat. No. 04CH37601), Vol. 4* (pp. 4286–4291). IEEE.

Russell Carpenter, J. (2002). Decentralized control of satellite formations. *International Journal of Robust and Nonlinear Control*, *12*(2–3), 141–161. http://dx.doi.org/10.1002/rnc.680.

Sampathirao, A. K., Hofmann, S., Raisch, J., & Hans, C. A. (2021). Distributed conditional cooperation model predictive control of interconnected microgrids. arXiv:1810.03361.

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

Sodnik, Z., Furch, B., & Lutz, H. (2010). Optical intersatellite communication. *IEEE Journal of Selected Topics in Quantum Electronics*, *16*(5), 1051–1057.

Spiridonova, S., Kirschner, M., & Hugentobler, U. (2014). Precise mean orbital elements determination for LEO monitoring and maintenance.

Sullivan, J., Koenig, A., & D'Amico, S. (2016). Improved maneuver-free approach to angles-only navigation for space rendezvous. In *26th AAS/AIAA space flight mechanics meeting, Napa, CA*.

Thien, R. T., & Kim, Y. (2018). Decentralized formation flight via PID and integral sliding mode control. *Aerospace Science and Technology*, *81*, 322–332.

Vallado, D. A. (1997). *Fundamentals of astrodynamics and applications*. McGraw-Hill Companies, Inc. College Custom Series.

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

Vlahakis, E., Dritsas, L., & Halikias, G. (2019). Distributed LQR design for a class of large-scale multi-area power systems. *Energies*, *12*(14), 2664.

Wanasinghe, T. R., Mann, G. K., & Gosine, R. G. (2014). Decentralized cooperative localization for heterogeneous multi-robot system using split covariance intersection filter. In *2014 Canadian conference on computer and robot vision* (pp. 167–174). IEEE.

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

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