Discrete-time distributed Kalman filter design for formations of autonomous vehicles

Daniel Viegas, Pedro Batista, Paulo Oliveira, and Carlos Silvestre

Control Engineering Practice, vol. 75, pp. 55-68, June 2018

https://doi.org/10.1016/j.conengprac.2018.03.014

Discrete-time distributed Kalman filter design for formations of autonomous vehicles[☆]

Daniel Viegas[a,*], Pedro Batista[b], Paulo Oliveira[b,c], Carlos Silvestre[a,1]

[a] *Department of Electrical and Computer Engineering of the Faculty of Science and Technology of the University of Macau, Macao, China*
[b] *Institute for Systems and Robotics (ISR), Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal*
[c] *IDMEC, Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal*

Abstract

This paper addresses the problem of distributed state estimation in a multi-vehicle framework. Each vehicle aims to estimate its own state relying on locally available measurements and limited communication with other vehicles in the vicinity. The dynamics of the problem are formulated as a discrete-time Kalman filtering problem with a sparsity constraint on the gain, and two different algorithms for computation of steady-state observer gains for arbitrary fixed measurement topologies are introduced. Their application to the practical problem of distributed localization in a formation of Autonomous Underwater Vehicles (AUVs) is detailed, supported by simulation results.

Keywords: Distributed estimation, Kalman filtering, Multi-vehicle systems, Autonomous underwater vehicles, State observer design

1. Introduction

Advances in science and technology in the last decades have led to the design of computers and autonomous vehicles that are increasingly powerful and versatile, and at the same time cheaper and more miniaturized. This has led to

[☆] Running head: Discrete-time distributed Kalman filter design

[*] Corresponding author. Fax: (+351) 218418291
Email address: dviegas@isr.ist.utl.pt (Daniel Viegas)
[1] Carlos Silvestre is on leave from Instituto Superior Técnico/University of Lisbon.

an interest in using multiple autonomous vehicles cooperatively to solve problems in a more time and resource efficient way, or even perform tasks beyond the capabilities of a single vehicle. In fact, there are several areas where the use of autonomous vehicles in formations offers attractive possibilities, such as unmanned formation flight (Giulietti et al. (2000); Wolfe et al. (1996)), underwater applications (Curtin et al. (1993); Healey (2001)), and automated highways systems (Bender (1991); Yanakiev and Kanellakopoulos (1996)). For more on applications and challenges involving multi-vehicle systems, see e.g. Kumar et al. (2008), Parker (2008), and references therein.

Naturally, the use of multi-agent systems has led to new challenges in the field of control and estimation. While the use of centralized solutions is attractive as the underlying theory is well understood, it may lead to severe problems during implementation due to the reliance on a central processing node that collects relevant data from all the vehicles, performs all the necessary computations, and relays the results back to the vehicles. As the number of vehicles increases, the computational load on the central node and the number of required communications will also increase, and may in fact render the centralized solution inviable. On the other hand, using distributed solutions, that is, breaking down the problems into smaller parts and making each vehicle responsible for a part of the computations can help bypass these scaling problems. However, there is a trade-off: while implementation is simpler and more efficient, design and analysis become much more complex, and problems that are handily solved within a centralized framework become difficult or even intractable, see e.g. Blondel and Tsitsiklis (2000) and Witsenhausen (1968).

There have been numerous interesting contributions in both distributed control (Antonelli et al. (2014); Cao et al. (2011); Cortés et al. (2006); Dunbar and Murray (2006); Olfati-Saber and Murray (2004)) and estimation (Liu et al. (2014); Olfati-Saber (2007); Pasqualetti et al. (2012); Sousa et al. (2009); Yuan and Tanner (2010)). Many more approaches and solutions can be found in recent surveys (Antonelli (2013); Bakule (2014); Cao et al. (2013); Oh et al. (2015)) and books (Bullo et al. (2009); Davison and Aghdam (2014); Lewis

et al. (2013); Mahmoud (2010)) on the subject. The specific case treated in this paper is part of the general framework of distributed controller and filter design subject to sparsity constraints. In it, the problems are modeled similarly to classical optimal control and filtering formulations, the key difference being that the controller or filter gain must verify a sparsity constraint or, in other words, certain entries of the gain matrix (or controller structure when designing dynamic controllers) are forced to zero to reflect the distributed nature of the problem. The most commonly seen formulations cast the infinite-horizon optimal control problem as optimization of feasibility problems involving bilinear matrix inequalities (BMIs), and existing approaches include suboptimal application of centralized design techniques (Bakule (2008); Siljak (2011)) and nonlinear optimization algorithms (Bompart et al. (2007); Dörfler et al. (2014); Lin et al. (2011); Zhai et al. (2001)). In the past few years, significant advances have been made for specific classes of problems, with solutions featuring global optimality guarantees. Regarding specific sparsity structures, key topics include quadratic invariance (Lamperski and Lessard (2015); Lessard and Lall (2016); Rotkowitz and Lall (2006)) and partially ordered sets (Shah and Parrilo (2008, 2013)). Optimal results also exist for certain classes of dynamic systems such as positive systems (Rantzer (2015); Tanaka and Langbort (2011)) and spatially invariant systems (Bamieh et al. (2002); Bamieh and Voulgaris (2005)).

Regarding the more specific problem of distributed position and velocity estimation in formations of vehicles, the most compelling contributions made in recent years consider range measurements as the basis for the positioning algorithms. In Antonelli et al. (2010), another EKF-based solution is presented to estimate the relative motion between two AUVs, featuring an observability analysis which results in weak local observability guarantees. Similarly, in Gadre and Stilwell (2005) the problem of localization based on range measurements to a single fixed beacon is addressed with weak local observability results based on the linearization of the dynamics of the problem and an EKF solution. Other approaches include the one proposed in Bahr et al. (2009), in which successive measurements at different points of the trajectory are used to recover the present

position, while Webster et al. (2012) presents a centralized EKF solution backed by extensive experimental results. Finally, Zhou and Roumeliotis (2008) studies the minimum number of distinct range measurements to compute both relative position and orientation in a 2-D setting, and proposes an algorithm to solve the problem, as well as a local observability result. Previous work in Viegas et al. (2015a) also addressed this topic, providing global observability and stability guarantees. The work presented in this paper eschews this framework in favor of a solution based in relative position measurements, allowing for a greater focus on filtering performance optimization as well as more general formulations which can be systematically applied to different vehicle models and arbitrary measurement topologies. Note that relative position measurements between vehicles can be obtained reliably in applications with AUVs as well as other types of vehicles, see e.g. Morgado et al. (2011) and Viegas et al. (2014).

This paper addresses the problem of distributed state estimation for formations of autonomous vehicles in a discrete-time framework. In the envisioned scenario, each vehicle in the formation aims to estimate its own state based on locally available measurements and communication with other vehicles in the vicinity. It is assumed that one or a few vehicles have access to measurements of their own state, while the rest must rely on measurements of their state relative to one or more other vehicles in the vicinity, as well as updated state estimates received from those vehicles through communication. Previous work by the authors in Viegas et al. (2012) tackled this problem in continuous-time, and Viegas et al. (2013) detailed an extension of these results to discrete-time. In the aforementioned works, linear matrix inequality (LMI) theory was used to derive algorithms to design distributed state observers with stability and performance guarantees. In this paper, another approach based on the discrete-time Kalman filter equations is considered to design computationally efficient solutions to obtain gains for the distributed state observer for arbitrary fixed measurement topologies. The problem is reformulated as the more general problem of Kalman filtering with a sparsity constraint on the gain, and two methods to compute distributed observer gains are introduced. The first proposed solution,

denoted as the one-step method in the sequel, consists in a direct adaptation of the Kalman filter iterations with the addition of the sparsity constraint on the gain. While its performance in terms of achieved steady-state error covariance is shown to be inferior to existing methods, it is nevertheless a very computationally efficient and scalable method to compute gains for distributed state observers and, unlike many existing solutions, it does not require an initial stabilizing gain for its execution. The second proposed solution, denoted as the finite-horizon method in the sequel, consists in an iterative algorithm that optimizes the observer gains over a finite-window to approximate optimal steady-state behavior. While the solution introduced in Viegas et al. (2012) requires solving an optimization problem subject to LMI constraints numerically in each step, the finite-horizon method bypasses this issue as the solution of the optimization problem solved in each iteration can be efficiently computed in closed-form. Moreover, as it is discussed in more detail in Section 5, while it requires a set of filter gains for its initialization, those initial gains do not need to be stabilizing or even verify the sparsity constraint. This framework is then particularized to the practical case of a formation of Autonomous Underwater Vehicles (AUVs), and an equivalent continuous-discrete formulation for local observer design is introduced to take advantage of the much faster sampling rate of on-board instrumentation in comparison with the positioning system. Finally, the performance of the proposed solutions is assessed in simulation for two cases: the aforementioned formation of AUVs, and a randomly generated system. Preliminary results on this subject have been presented in Viegas et al. (2017). The present manuscript details an additional method for efficient distributed gain computation, and detailed derivations are included for the closed-form solutions of the optimization problems solved in both proposed methods. Additionally, the sections on both methods feature a simple example to support the theoretical exposition. Moreover, the simulation results have been expanded to include an additional example.

The rest of the paper is organized as follows. Section 2 details the dynamics of the vehicles as well as the structure of the local state observers, and Section

3 describes the more general problem of Kalman filtering with sparsity constraints. Sections 4 and 5 detail two different methods for computing gains for distributed state observers, and Section 6 details their application to the problem of position estimation in a formation of AUVs, with simulation results to assess their performance in simulation. To provide another application example, Section 7 presents further simulation results for a randomly generated system. Finally, Section 8 summarizes the main conclusions of the paper.

1.1. Notation

Throughout the paper, $\mathbf{I}$ and $\mathbf{0}$ denote the identity matrix and a vector or matrix of zeros, respectively, both of appropriate dimensions. When necessary, $\mathbf{I}_n$ is used to denote the n by n identity matrix. A block diagonal matrix $\mathbf{A}$ with n blocks $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n$ is denoted by $\mathbf{A} = \text{diag}(\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n)$. Similarly, a diagonal matrix $\mathbf{B}$ whose entries are given by the entries of a vector $\mathbf{b}$ is denoted by $\mathbf{B} = \text{diag}(\mathbf{b})$. The notation $\text{vec}(\mathbf{A})$ denotes the vectorization operator, which returns a vector composed of the columns of the matrix $\mathbf{A}$. For a symmetric matrix $\mathbf{P}$, $\mathbf{P} \succ \mathbf{0}$ and $\mathbf{P} \succeq \mathbf{0}$ indicate that $\mathbf{P}$ is positive definite or semidefinite, respectively. The Kronecker product of two matrices $\mathbf{A}$ and $\mathbf{B}$ is denoted by $\mathbf{A} \otimes \mathbf{B}$.

2. Problem Statement

Consider a formation composed of N autonomous vehicles, each indexed by a distinct integer $i \in \{1, 2, \ldots, N\}$. The problem considered in this paper is the design of a distributed state estimation solution such that each vehicle can estimate its own state, usually relevant variables such as position and linear velocity, using only measurements obtained from sensors mounted on-board and limited communication with other vehicles in the vicinity. The measurements available to each vehicle are divided into two categories: one or more vehicles have access to measurements of their own state, referred to as *inertial* measurements in the sequel, while the rest must rely instead on measurements of

their state *relative* to other vehicles in the vicinity. For the latter case, it is also assumed that they receive updated state estimates from the corresponding vehicles through communication, to serve as a basis for comparison with the relative measurements.

The solution proposed here consists in the implementation of a local state observer on-board each vehicle, which uses the aforementioned locally available measurements and communication to estimate the state of the vehicle in real time. This section briefly details the dynamic systems employed to model the vehicles, as well as the structure of the local observers. The system dynamics, as well as the structure of the local observers, are essentially a discrete-time version of those introduced in Viegas et al. (2012), as detailed in Viegas et al. (2013), and are hence briefly presented here only for the sake of completeness. The contribution of the paper then follows on how to design the observer gains.

2.1. Measurement graph

Before proceeding, it is convenient to introduce some concepts of graph theory, see e.g. Wallis (2007) and West (2001), as vehicle formations such as the one considered in this paper can be compactly described by directed graphs. A directed graph, or digraph, $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ is composed of a set $\mathcal{V}$ of vertices and a set of directed edges $\mathcal{E}$. An edge can be expressed as $e = (a, b)$, meaning that edge e is incident on vertices a and b, directed towards b. For a vertex i, its in-degree ν_i is the number of edges directed towards and incident on it, and its in-neighborhood $\mathcal{D}_i = \{d_i^1, d_i^2, \ldots, d_i^{\nu_i}\}$ is the set of corresponding vertices, that is, $j \in \mathcal{D}_i$ if and only if $(j, i) \in \mathcal{E}$. A digraph $\mathcal{G}$ with n_v vertices and n_e edges can be described by an incidence matrix $\boldsymbol{\mathcal{I}}_{\mathcal{G}} \in \mathbb{R}^{n_v \times n_e}$, whose individual entries follow

$$[\boldsymbol{\mathcal{I}}_{\mathcal{G}}]_{jk} = \begin{cases} 1, & \text{edge } k \text{ directed towards } j \\ -1, & \text{edge } k \text{ directed away from } j \\ 0, & \text{edge } k \text{ not incident on } j \end{cases} .$$

The ordering of the edges in $\boldsymbol{\mathcal{I}}_{\mathcal{G}}$ is not relevant in most applications, however in this paper the following convention is chosen, without loss of generality, to

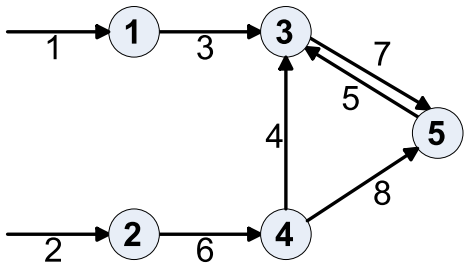

Figure 1: Measurement graph for a sample formation composed of 5 vehicles

simplify the notation: first, all edges of the form $(i, 1)$, then all edges of the form $(i, 2)$, and so on, finishing with the (i, N) edges.

Now, consider the vehicle formation described at the beginning of this section. The measurement scheme of the formation can be described by a digraph $\mathcal{G}_M$. In the measurement graph $\mathcal{G}_M$ each vertex represents a distinct vehicle, and an edge (a, b) signifies that vehicle b has access to a measurement relative to vehicle a. Finally, define a special set of edges of the form $(0, i)$, connected to only one vertex, which represent the absolute state measurements available to some of the vehicles. Figure 1 depicts an example of such a measurement graph, whose associated incidence matrix is given by

$$\mathcal{I}_{\mathcal{G}_M} = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 1 \end{bmatrix}.$$

The numbers next to each edge in the figure denote the corresponding column in $\mathcal{I}_{\mathcal{G}_M}$.

Throughout the paper, it is assumed that the measurement graph $\mathcal{G}_M$ associated with the vehicle formation is fixed, that is, the edges in $\mathcal{G}_M$ do not change over the course of the mission. Previous work by the authors in Viegas et al. (2015b) addressed the problem of a time-varying measurement graph, however it is not considered in the present paper as the main focus is on developing novel

methods of distributed filter design for a given measurement topology.

2.2. Local state observers

For a vehicle i which has access to inertial measurements, its dynamics are described by the linear time-invariant (LTI) system

$$\begin{cases} \mathbf{x}_i(k+1) = \mathbf{A}_L\mathbf{x}_i(k) + \mathbf{B}_L\mathbf{u}_i(k) + \mathbf{w}_i(k) \\ \mathbf{y}_i(k) = \mathbf{C}_L\mathbf{x}_i(k) + \mathbf{v}_i(k) \end{cases}, \tag{1}$$

where the state $\mathbf{x}_i(k) \in \mathbb{R}^{n_L}$ is the the quantity to be estimated, the input $\mathbf{u}_i(k) \in \mathbb{R}^{m_L}$ is assumed to be known, and the measured output $\mathbf{y}_i(k) \in \mathbb{R}^{o_L}$ is the inertial measurement at time k. $\mathbf{A}_L$, $\mathbf{B}_L$, and $\mathbf{C}_L$ are given constant matrices of appropriate dimensions. $\mathbf{w}_i(k) \in \mathbb{R}^{n_L}$ and $\mathbf{v}_i(k) \in \mathbb{R}^{o_L}$ represent process and observation noise, and are assumed to be zero-mean white Gaussian processes with associated covariance matrices $\mathbf{Q}_i \succeq \mathbf{0} \in \mathbb{R}^{n_L \times n_L}$ and $\mathbf{R}_i \succ \mathbf{0} \in \mathbb{R}^{o_L \times o_L}$, respectively. It is assumed that the pair $(\mathbf{A}_L, \mathbf{C}_L)$ is observable. To estimate the state, the prediction-filtering scheme used in the Kalman filter is followed. Denote the predicted state estimate at time k by $\hat{\mathbf{x}}_i(k|k-1) \in \mathbb{R}^{n_L}$ and the filtered state estimate by $\hat{\mathbf{x}}_i(k|k) \in \mathbb{R}^{n_L}$. The predicted estimate is given by

$$\hat{\mathbf{x}}_i(k+1|k) = \mathbf{A}_L\hat{\mathbf{x}}_i(k|k) + \mathbf{B}_L\mathbf{u}_i(k), \tag{2}$$

and the filtered estimate follows

$$\hat{\mathbf{x}}_i(k+1|k+1) = \hat{\mathbf{x}}_i(k+1|k) + \mathbf{K}_i(\mathbf{y}_i(k+1) - \mathbf{C}_L\hat{\mathbf{x}}_i(k+1|k)), \tag{3}$$

in which $\mathbf{K}_i \in \mathbb{R}^{n_L \times o_L}$ is a constant matrix of observer gains. For this case, it is straightforward to compute $\mathbf{K}_i$ such that the observer features globally exponentially stable error dynamics, see e.g. Anderson and Moore (2012).

On the other hand, if vehicle i has access to relative measurements, its dynamics follow

$$\begin{cases} \mathbf{x}_i(k+1) = \mathbf{A}_L\mathbf{x}_i(k) + \mathbf{B}_L\mathbf{u}_i(k) + \mathbf{w}_i(k) \\ \mathbf{y}_i(k) = \mathbf{C}_i\Delta\mathbf{x}_i(k) + \mathbf{v}_i(k) \end{cases}, \tag{4}$$

in which $\mathbf{y}_i(k) \in \mathbb{R}^{o_L \nu_i}$, $\mathbf{C}_i = \mathbf{I}_{\nu_i} \otimes \mathbf{C}_L$, and

$$\Delta\mathbf{x}_i(k) := \begin{bmatrix} \mathbf{x}_i(k) - \mathbf{x}_{d_i^1}(k) \\ \mathbf{x}_i(k) - \mathbf{x}_{d_i^2}(k) \\ \vdots \\ \mathbf{x}_i(k) - \mathbf{x}_{d_i^{\nu_i}}(k) \end{bmatrix} \in \mathbb{R}^{n_L \nu_i},\ d_i^j \in \mathcal{D}_i,$$

where ν_i and $\mathcal{D}_i$ are respectively the in-degree and the in-neighborhood of the vertex associated with vehicle i in the measurement graph $\mathcal{G}_M$ of the formation. The remaining variables are defined as in (1). For this case, the following prediction-filtering structure for the system (4) can be implemented. The predicted estimate is given by (2), and the filtered estimate follows

$$\hat{\mathbf{x}}_i(k+1|k+1) = \hat{\mathbf{x}}_i(k+1|k) + \mathbf{K}_i(\mathbf{y}_i(k+1) - \mathbf{C}_i\Delta\hat{\mathbf{x}}_i(k+1|k)), \qquad (5)$$

where $\mathbf{K}_i \in \mathbb{R}^{n_L \times o_L \nu_i}$ is a constant matrix of observer gains, to be computed, and $\Delta\hat{\mathbf{x}}_i(k|k-1)$ is constructed using the state estimates received through communication, in addition to the estimate of the state of vehicle i itself:

$$\Delta\hat{\mathbf{x}}_i(k|k-1) := \begin{bmatrix} \hat{\mathbf{x}}_i(k|k-1) - \hat{\mathbf{x}}_{d_i^1}(k|k-1) \\ \hat{\mathbf{x}}_i(k|k-1) - \hat{\mathbf{x}}_{d_i^2}(k|k-1) \\ \vdots \\ \hat{\mathbf{x}}_i(k|k-1) - \hat{\mathbf{x}}_{d_i^{\nu_i}}(k|k-1) \end{bmatrix} \in \mathbb{R}^{n_L \nu_i},\ d_i^j \in \mathcal{D}_i.$$

Note that, as the evolution of the state estimate depends not only on the local dynamics of the vehicle but also on the state estimates of other vehicles, even though the pair $(\mathbf{A}_L, \mathbf{C}_i)$ is assumed to be observable, the matrix $\mathbf{K}_i$ of observer gains cannot be computed as straightforwardly as in the previous case. To do so, it is necessary to consider the dynamics of the formation as a whole in order to take into account the interactions between the dynamics of the different local observers.

2.3. Global dynamics

The global dynamics of the formation can be represented by the LTI system

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}_g \mathbf{x}(k) + \mathbf{B}_g \mathbf{u}(k) + \mathbf{w}(k) \\ \mathbf{y}(k) = \mathbf{C}_g \mathbf{x}(k) + \mathbf{v}(k) \end{cases}, \tag{6}$$

where $\mathbf{x}(k) := \begin{bmatrix} \mathbf{x}_1^\mathsf{T}(k) & \dots & \mathbf{x}_N^\mathsf{T}(k) \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{n_L N}$ is the concatenation of the states of each vehicle in the formation, $\mathbf{y}(k) := \begin{bmatrix} \mathbf{y}_1^\mathsf{T}(k) & \dots & \mathbf{y}_N^\mathsf{T}(k) \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{o_L M}$ is the concatenation of the measurements available to each vehicle, M being the number of edges in $\mathcal{G}_M$, and $\mathbf{u}(k) := \begin{bmatrix} \mathbf{u}_1^\mathsf{T}(k) & \dots & \mathbf{u}_N^\mathsf{T}(k) \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{m_L N}$ is the concatenation of the inputs of each vehicle. In the same way, $\mathbf{w}(k) := \begin{bmatrix} \mathbf{w}_1^\mathsf{T}(k) & \dots & \mathbf{w}_N^\mathsf{T}(k) \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{n_L N}$ and $\mathbf{v}(k) := \begin{bmatrix} \mathbf{v}_1^\mathsf{T}(k) & \dots & \mathbf{v}_N^\mathsf{T}(k) \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{o_L M}$ are the state and measurement noise vectors, respectively. The matrices $\mathbf{A}_g$, $\mathbf{B}_g$, and $\mathbf{C}_g$ are built from the dynamics of the individual agents, following

$$\begin{cases} \mathbf{A}_g = \mathbf{I}_N \otimes \mathbf{A}_L \\ \mathbf{B}_g = \mathbf{I}_N \otimes \mathbf{B}_L \\ \mathbf{C}_g = \boldsymbol{\mathcal{I}}_{\mathcal{G}_M}^\mathsf{T} \otimes \mathbf{C}_L \end{cases},$$

where $\boldsymbol{\mathcal{I}}_{\mathcal{G}_M}$ is the incidence matrix of $\mathcal{G}_M$, constructed using the specific ordering detailed in Section 2.1. The local state observers can also be grouped in a similar way. The predicted estimate $\hat{\mathbf{x}}(k|k-1) := \begin{bmatrix} \hat{\mathbf{x}}_1^\mathsf{T}(k|k-1) & \dots & \hat{\mathbf{x}}_N^\mathsf{T}(k|k-1) \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{n_L N}$ is updated following

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{A}_g \hat{\mathbf{x}}(k|k) + \mathbf{B}_g \mathbf{u}(k), \tag{7}$$

and the filtered state estimate $\hat{\mathbf{x}}(k|k) := \begin{bmatrix} \hat{\mathbf{x}}_1^\mathsf{T}(k|k) & \dots & \hat{\mathbf{x}}_N^\mathsf{T}(k|k) \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{n_L N}$ follows

$$\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{K}_g(\mathbf{y}(k+1) - \mathbf{C}_g \hat{\mathbf{x}}(k+1|k)), \tag{8}$$

where $\mathbf{K}_g \in \mathbb{R}^{n_L N \times o_L M}$ is the matrix of observer gains. To account for the fact that each local observer only has access a limited set of measurements, $\mathbf{K}_g$ must follow a block diagonal structure:

$$\mathbf{K}_g = \operatorname{diag}(\mathbf{K}_1, \mathbf{K}_2, \dots, \mathbf{K}_N). \tag{9}$$

Due to this structural constraint on the observer gain, classical observer design methods cannot be applied. However, as it will be shown in the sequel, the Kalman filter iterations can be used as a starting point to design methods to compute observer gains for such a distributed state observer.

Throughout the paper it is assumed that the distributed state observer is stabilizable, that is, there exists a gain of the form (9) such that, in the absence of noise, the global state estimate $\hat{\mathbf{x}}(k|k)$ converges globally exponentially fast to the state $\mathbf{x}(k)$ of (6). For an easily verifiable sufficient condition for stabilizability of this type of distributed state observer, see e.g. (Viegas et al., 2012, Theorem 1 and Remark 3).

3. Kalman filter with sparsity constraints

The block diagonal structure of the gain of the distributed state observer (7)-(8) belongs to a more general category of constraints, commonly denoted as sparsity constraints. In this paper, the following notation is used to describe sparsity patterns. For $\mathbf{E} \in \mathbb{R}^{n\times o}$, the set of matrices which follow the sparsity pattern of $\mathbf{E}$ is defined as

$$\text{Sparse}(\mathbf{E}) = \left\{\mathbf{K} \in \mathbb{R}^{n\times o} \colon [\mathbf{E}]_{ij} = 0 \implies [\mathbf{K}]_{ij} = 0,\ i = 1,\dots,n,\ j = 1,\dots,o\right\}.$$

This section details the problem of Kalman filtering with sparsity constraints on the filter gain, as well as its relevance towards solving the main problem of this paper, that is, computing a constant gain of the form (9) for the distributed state observer (7)-(8).

Consider the general discrete time LTI system

$$\begin{cases} \boldsymbol{x}(k+1) = \mathbf{A}\boldsymbol{x}(k) + \mathbf{B}\boldsymbol{u}(k) + \boldsymbol{w}(k) \\ \boldsymbol{y}(k) = \mathbf{C}\boldsymbol{x}(k) + \boldsymbol{v}(k) \end{cases}, \tag{10}$$

where $\boldsymbol{x}(k) \in \mathbb{R}^n$, $\boldsymbol{u}(k) \in \mathbb{R}^m$, and $\boldsymbol{y}(k) \in \mathbb{R}^o$ are the state, input, and output of the system, respectively, $\boldsymbol{w}(k) \in \mathbb{R}^n$ and $\boldsymbol{v}(k) \in \mathbb{R}^o$ are the process and observation noises, which are assumed to be independent zero-mean white

Gaussian processes with associated covariance matrices $\mathbf{Q} \succeq \mathbf{0} \in \mathbb{R}^{n\times n}$ and $\mathbf{R} \succ \mathbf{0} \in \mathbb{R}^{o\times o}$, respectively, and $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are given constant matrices of appropriate dimensions.

As in the centralized Kalman filter, the state estimate is updated in two steps. First, a prediction step which updates the state estimate given the knowledge of the process model, followed by a filtering step using the measured output. The usual notation is followed: $\hat{\boldsymbol{x}}(k|k-1) \in \mathbb{R}^n$ denotes the state estimate after the prediction at step k, with associated estimation error covariance $\mathbf{P}(k|k-1) \succeq \mathbf{0} \in \mathbb{R}^{n\times n}$, and $\hat{\boldsymbol{x}}(k|k) \in \mathbb{R}^n$ denotes the state estimate after the correction at the same step k, with associated estimation error covariance $\mathbf{P}(k|k) \succeq \mathbf{0} \in \mathbb{R}^{n\times n}$. The prediction step remains the same, that is,

$$\hat{\boldsymbol{x}}(k+1|k) = \mathbf{A}\hat{\boldsymbol{x}}(k|k) + \mathbf{B}\boldsymbol{u}(k), \tag{11}$$

and the estimation error covariance is updated accordingly:

$$\mathbf{P}(k+1|k) = \mathbf{A}\mathbf{P}(k|k)\mathbf{A}^{\mathsf{T}} + \mathbf{Q}. \tag{12}$$

Then, the a priori estimate (11) for $\boldsymbol{x}(k+1)$ is corrected using the observation $\boldsymbol{y}(k+1)$. The filtered state estimate is computed following

$$\hat{\boldsymbol{x}}(k+1|k+1) = \hat{\boldsymbol{x}}(k+1|k) + \mathbf{K}(k+1)(\boldsymbol{y}(k+1) - \mathbf{C}\hat{\boldsymbol{x}}(k+1|k)), \tag{13}$$

where $\mathbf{K}(k+1) \in \mathbb{R}^{n\times o}$ is the filter gain, and the estimation error covariance follows

$$\begin{aligned}&\mathbf{P}(k+1|k+1) = \\ &\mathbf{K}(k+1)\mathbf{R}\mathbf{K}^{\mathsf{T}}(k+1) + (\mathbf{I} - \mathbf{K}(k+1)\mathbf{C})\mathbf{P}(k+1|k)(\mathbf{I} - \mathbf{K}(k+1)\mathbf{C})^{\mathsf{T}}.\end{aligned} \tag{14}$$

For a finite time window $W \in \mathbb{N}$, the problem of designing a Kalman filter subject to a sparsity constraint on the gain matrix can then be formulated as follows. Given a sparsity pattern $\mathbf{E} \in \mathbb{R}^{n\times o}$ and an initial state estimate $\hat{\mathbf{x}}(0|0)$

with associated error covariance $\mathbf{P}(0|0) \succeq \mathbf{0}$, solve the optimization problem

$$\begin{aligned} \underset{\substack{\mathbf{K}(i)\in\mathbb{R}^{n\times o}\\ i=1,2,\ldots,W}}{\text{minimize}} \quad & \sum_{k=1}^{W} \operatorname{tr}(\mathbf{P}(k|k)) \\ \text{subject to} \quad & \mathbf{K}(i) \in \text{Sparse}(\mathbf{E}),\ i = 1, 2, \ldots, W \end{aligned} \tag{15}$$

Now, consider that a constant gain $\mathbf{K}_\infty$ is chosen for all steps instead of a sequence of gains $\mathbf{K}(k)$. Then, if $\mathbf{K}_\infty$ stabilizes the error dynamics of the filter (11)-(14), it is well known (see e.g. Anderson and Moore (2012)) that the estimation error covariance converges to a steady-state solution, that is, there exists a constant $\mathbf{P}_\infty \in \mathbb{R}^{n\times n}$ such that

$$\lim_{k\to+\infty} \mathbf{P}(k|k) = \mathbf{P}_\infty. \tag{16}$$

Thus, finding the optimal steady-state gain for the Kalman filter subject to a sparsity constraint can be formulated as the optimization problem

$$\begin{aligned} \underset{\mathbf{K}_\infty\in\mathbb{R}^{n\times o}}{\text{minimize}} \quad & \operatorname{tr}(\mathbf{P}_\infty) \\ \text{subject to} \quad & \mathbf{K}_\infty \in \text{Sparse}(\mathbf{E}) \end{aligned} \tag{17}$$

The optimization problems (15) and (17) are both nonconvex and, unlike the unconstrained Kalman filter, finding their optimal solution is still an open problem. In the next two sections, two methods based on the covariance iterations (12) and (14), as well as the finite-window formulation (15), are presented to compute well-performing steady-state filter gains for the distributed state observer or, in other words, to approximate the optimal solution of (17).

This framework provides a systematic approach to compute suitable gains for the distributed state observer of the previous section. First, set $\mathbf{A} = \mathbf{A}_g$, $\mathbf{B} = \mathbf{B}_g$, $\mathbf{C} = \mathbf{C}_g$, $\mathbf{Q} = \operatorname{diag}(\mathbf{Q}_1, \ldots, \mathbf{Q}_N)$, and $\mathbf{R} = \operatorname{diag}(\mathbf{R}_1, \ldots, \mathbf{R}_N)$ in (10), and choose an appropriate matrix $\mathbf{E}$ to define the sparsity pattern, that is, one that is consistent with the desired structure for $\mathbf{K}_g$ as defined in (9). Then, use one of the methods detailed in the next two sections to compute a well-performing steady-state filter gain $\mathbf{K}$ with the desired sparsity pattern. Finally, recover the individual $\mathbf{K}_i$ for each local filter from the block diagonal $\mathbf{K}$ that resulted from this process.

4. One-step method for computation of observer gains

This section presents the first of two methods detailed in this paper to compute gains for a state observer subject to a sparsity constraint on the gain.

For this method, the strategy usually employed for the classical Kalman filter is followed, that is, the gain for each step is computed by minimizing the estimation error covariance of that same step, by solving the optimization problem

$$\begin{aligned} \underset{\mathbf{K}(k+1)\in\mathbb{R}^{n\times o}}{\text{minimize}} \quad & \operatorname{tr}(\mathbf{P}(k+1|k+1)) \\ \text{subject to} \quad & \mathbf{K}(k+1) \in \operatorname{Sparse}(\mathbf{E}) \end{aligned} \tag{18}$$

given the predicted error covariance $\mathbf{P}(k+1|k)$ at that time step. By substituting (14) in the objective function of (18), it is possible to verify that (18) is a quadratic optimization problem subject to linear equality constraints, and can thus be solved efficiently with numerical methods, much like in the unconstrained case. However, while for the unconstrained Kalman filter this method yields the optimal solution for the infinite-horizon problem (14), it is suboptimal when the filter is subjected to a sparsity constraint. Nevertheless, this method provides a computationally efficient way to compute steady-state gains that stabilize the error dynamics of distributed state observers such as the one detailed in Section 2.

The optimal solution for the optimization problem (18) can actually be obtained in closed-form, as detailed in the following result.

Theorem 4.1. *Let $\boldsymbol{l}_i$ denote a column vector whose entries are all set to zero except for the i-th one, which is set to 1, and define $\boldsymbol{\mathcal{L}}_i := \operatorname{diag}(\boldsymbol{l}_i)$. Define a vector $\boldsymbol{m}_i \in \mathbb{R}^o$ to encode the nonzero entries in the i-th row of $\mathbf{K}(k+1)$, following*

$$\begin{cases} \boldsymbol{m}_i(j) = 0 \;\text{ if }\; [\mathbf{E}]_{ij} = 0 \\ \boldsymbol{m}_i(j) = 1 \;\text{ if }\; [\mathbf{E}]_{ij} \neq 0 \end{cases}, \; j = 1, 2, \ldots, o,$$

and let $\boldsymbol{\mathcal{M}}_i = \operatorname{diag}(\boldsymbol{m}_i)$. Then, the optimal one-step gain that solves (18) is

given by

$$\mathbf{K}(k+1) = \sum_{i=1}^{n} \boldsymbol{\mathcal{L}}_i \mathbf{P}(k+1|k)\mathbf{C}^{\mathsf{T}} \boldsymbol{\mathcal{M}}_i (\mathbf{I} - \boldsymbol{\mathcal{M}}_i + \boldsymbol{\mathcal{M}}_i \mathbf{S}(k+1)\boldsymbol{\mathcal{M}}_i)^{-1}, \quad (19)$$

where $\mathbf{S}(k+1)$ *is the innovation covariance at step* $k+1$*, given by*

$$\mathbf{S}(k+1) = \mathbf{C}\mathbf{P}(k+1|k)\mathbf{C}^{\mathsf{T}} + \mathbf{R} \in \mathbb{R}^{o \times o}. \quad (20)$$

Proof. See Appendix Appendix A. □

Note that (19) shares some similarities with the closed-form gain formula for the unconstrained case, which is given by

$$\mathbf{K}(k+1) = \mathbf{P}(k+1|k)\mathbf{C}^{\mathsf{T}}\mathbf{S}^{-1}(k+1).$$

The main difference is that, instead of using the whole innovation covariance $\mathbf{S}(k+1)$, the entries of each row of $\mathbf{K}(k+1)$ are computed using the term

$$\boldsymbol{\mathcal{M}}_i \mathbf{S}(k+1) \boldsymbol{\mathcal{M}}_i,$$

which can be seen as a reduced form of $\mathbf{S}(k+1)$ that does not take into account all output variables, but only those corresponding to the entries in the i-th row of $\mathbf{K}(k+1)$ that are not forced to zero by the sparsity constraint.

To obtain constant gains that stabilize the state observer subject to sparsity constraints, propagate the covariance equations (12) and (14) using at each step the gain $\mathbf{K}(k+1)$ computed with (19), until the error covariance reaches a constant steady-state value.

4.1. Example of gain computation

To provide an example, this method was applied to a randomly generated system with $n = 5$ and $o = 4$. The values of the relevant system matrices,

rounded to 3 decimal places, are given by

$$\mathbf{A} = \begin{bmatrix} 0.152 & 0.092 & 0.235 & 0.642 & 0.506 \\ 0.397 & 0.615 & 0.448 & 0.221 & 0.279 \\ 0.375 & 0.011 & 0.569 & 0.837 & 0.747 \\ 0.131 & 0.573 & 0.061 & 0.971 & 0.237 \\ 0.435 & 0.790 & 0.496 & 0.846 & 0.957 \end{bmatrix}, \tag{21}$$

$$\mathbf{C} = \begin{bmatrix} 0.620 & 0.255 & 0.725 & 0.404 & 0.511 \\ 0.600 & 0.859 & 0.230 & 1.988 & 0.061 \\ 0.173 & 0.911 & 0.576 & 0.090 & 0.726 \\ 0.090 & 0.700 & 0.811 & 0.321 & 0.557 \end{bmatrix}, \tag{22}$$

$$\mathbf{Q} = \begin{bmatrix} 3.318 & 4.662 & 1.598 & -1.542 & -1.999 \\ 4.662 & 11.520 & 2.608 & -2.093 & -5.442 \\ 1.598 & 2.608 & 4.691 & 0.647 & -0.410 \\ -1.542 & -2.093 & 0.647 & 2.968 & 0.803 \\ -1.999 & -5.442 & -0.410 & 0.803 & 2.851 \end{bmatrix}, \quad \text{and} \tag{23}$$

$$\mathbf{R} = \begin{bmatrix} 3.624 & 2.601 & -0.042 & -0.944 \\ 2.601 & 7.343 & -0.729 & -2.786 \\ -0.042 & -0.729 & 0.745 & -0.242 \\ -0.944 & -2.786 & -0.242 & 1.612 \end{bmatrix}. \tag{24}$$

The sparsity pattern, which was also randomly generated, is given by

$$\mathbf{E} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}. \tag{25}$$

Figure 2 depicts the evolution of the trace of the covariance $\mathbf{P}(k+1|k+1)$ for several different initial covariances $\mathbf{P}(0|0)$. In all cases, the covariance converges

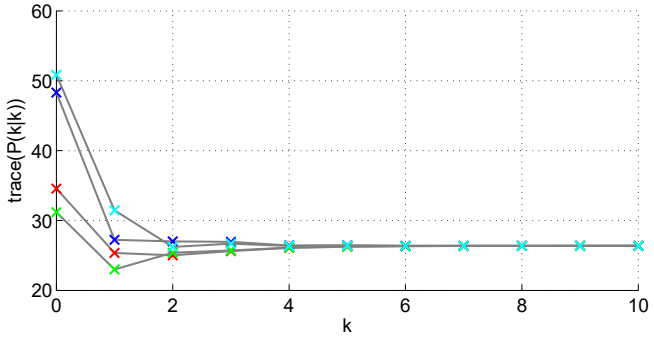

Figure 2: Convergence of the one-step method error covariance for different initial error covariances

very fast to the same steady-state value, and the corresponding constant steady-state gain matrix is

$$\mathbf{K}_\infty = \begin{bmatrix} 0.159 & 0 & 0.296 & -0.005 \\ 0 & 0.329 & 0 & 0.005 \\ 0 & 0 & 0.592 & 0 \\ 0.279 & 0.156 & -0.196 & 0 \\ 0.509 & -0.251 & 0 & -0.031 \end{bmatrix},$$

again rounded to 3 decimal places.

5. Finite-horizon method for computation of observer gains

This section details another method to compute gains for a state observer subject to a sparsity constraint on the gain. For this method, instead of optimizing the observer gain at step k with respect only to the covariance $\mathbf{P}(k|k)$ at that step, look instead at the finite-horizon problem

$$\begin{aligned} \underset{\substack{\mathbf{K}(i)\in\mathbb{R}^{n\times o} \\ i=1,2,\ldots,W}}{\text{minimize}} \quad & \sum_{k=1}^{W} \operatorname{tr}(\mathbf{P}(k|k)) \\ \text{subject to} \quad & \mathbf{K}(i) \in \text{Sparse}(\mathbf{E}),\ i = 1, 2, \ldots, W \end{aligned} \tag{26}$$

repeated here for the sake of clarity. The idea behind this formulation is to approximate steady-state behavior by making W large enough, and thus obtain

Table 1: Algorithm for computation of gains using the finite-horizon method

(1) **Initialization:** Select a window size, W, an initial covariance $\mathbf{P}(0|0) \succeq \mathbf{0}$, and compute a set of initial covariances $\mathbf{P}(k|k),\ k = 1, 2, \ldots, W$ and observer gains $\mathbf{K}(k),\ k = 1, \ldots, W$ using, e.g., the one-step method.

Select a stopping criterion, for example a minimum improvement on the objective function of (26) or a fixed number of iterations.

(2) Set i = W.

(3) **Inner loop:**

  (a) Fix all observer gains except for $\mathbf{K}(i)$, and solve (27) for $\mathbf{K}(i)$, replacing the previous value for that gain.

  (b) Set $i = i - 1$. If $i = 0$, exit the inner loop to Step (4). Otherwise, go to (a).

(4) Recompute the covariances $\mathbf{P}(1|1)$ through $\mathbf{P}(W|W)$ using the new observer gains.

(5) If the stopping criterion is met, stop execution. Otherwise, go to (2).

a constant filter gain that outperforms the one computed with the one-step method. However, the optimization problem (26) is nonconvex, as the objective function is polynomial in the optimization variables $\mathbf{K}(k),\ k = 1, 2, \ldots, W$. This can be verified by expressing the covariances $\mathbf{P}(k|k),\ k = 1, 2, \ldots, W$ in terms of the initial covariance $\mathbf{P}(0|0)$ and the gains $\mathbf{K}(k)$ using (12) and (14).

On the other hand, if all the gains are fixed to an arbitrary constant value except for a single $\mathbf{K}(k)$, the optimization problem (26) takes the form

$$\begin{aligned} \underset{\mathbf{K}(k) \in \mathbb{R}^{n \times o}}{\text{minimize}} \quad & \sum_{i=1}^{W} \operatorname{tr}(\mathbf{P}(i|i)) \\ \text{subject to} \quad & \mathbf{K}(k) \in \text{Sparse}(\mathbf{E}) \end{aligned} \tag{27}$$

which is quadratic, and can thus be solved using conventional methods.

To obtain constant steady-state gains for the state observer subject to sparsity constraints, the algorithm proposed in Table 1 (see Fig. 3 for a schematic

representation) is executed with a window size W large enough to recover the steady-state behavior expected from the optimal filter and thus approximate the optimal solution of (16). Then, from the sequence of gains $\mathbf{K}(k)$, $k = 1, 2, \ldots, W$, the one that yields the best steady-state covariance $\mathbf{P}_\infty$ is selected, found by propagating the covariance equations (12) and (14) for each of the gains.

The solution of the partial problem (27) can actually be computed in closed-form, as detailed in the next theorem.

Theorem 5.1. *Define a matrix* $\mathbf{Z}$ *such that the vector* $\mathbf{Z}\text{vec}(\mathbf{K}(k))$ *contains the nonzero elements of* $\mathbf{K}(k)$ *according to the desired sparsity pattern. The closed-form solution of* (27) *is given by*

$$\text{vec}(\mathbf{K}(k)) = \mathbf{Z}^\mathsf{T} \left(\mathbf{Z}(\mathbf{S}(k) \otimes \mathbf{\Lambda}(k+1))\mathbf{Z}^\mathsf{T} \right)^{-1} \mathbf{Z}\, \text{vec}(\mathbf{\Lambda}(k+1)\mathbf{P}(k|k-1)\mathbf{C}^\mathsf{T}), \tag{28}$$

where $\mathbf{S}(k)$ *is computed as in* (20)*, and*

$$\mathbf{\Lambda}(k+1) = \mathbf{I}_n + \sum_{i=k+1}^{W} \mathbf{\Gamma}^\mathsf{T}(k+1, i)\mathbf{\Gamma}(k+1, i),$$

with

$$\mathbf{\Gamma}(k_i, k_f) = \prod_{j=k_i}^{k_f} (\mathbf{I}_n - \mathbf{K}(k_i + k_f - j)\mathbf{C})\mathbf{A}. \tag{29}$$

When $k_i > k_f$*, the convention* $\mathbf{\Gamma}(k_i, k_f) = \mathbf{I}$ *is followed.*

Proof. See Appendix Appendix B. □

As an example of how to compute $\mathbf{Z}$, consider $\mathbf{K}(k)$ of the form

$$\mathbf{K}(k) = \begin{bmatrix} k_1 & 0 \\ k_2 & k_3 \\ 0 & k_4 \end{bmatrix}. \text{ Then, } \text{vec}(\mathbf{K}(k)) = \begin{bmatrix} k_1 \\ k_2 \\ 0 \\ 0 \\ k_3 \\ k_4 \end{bmatrix} \text{ and } \mathbf{Z} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

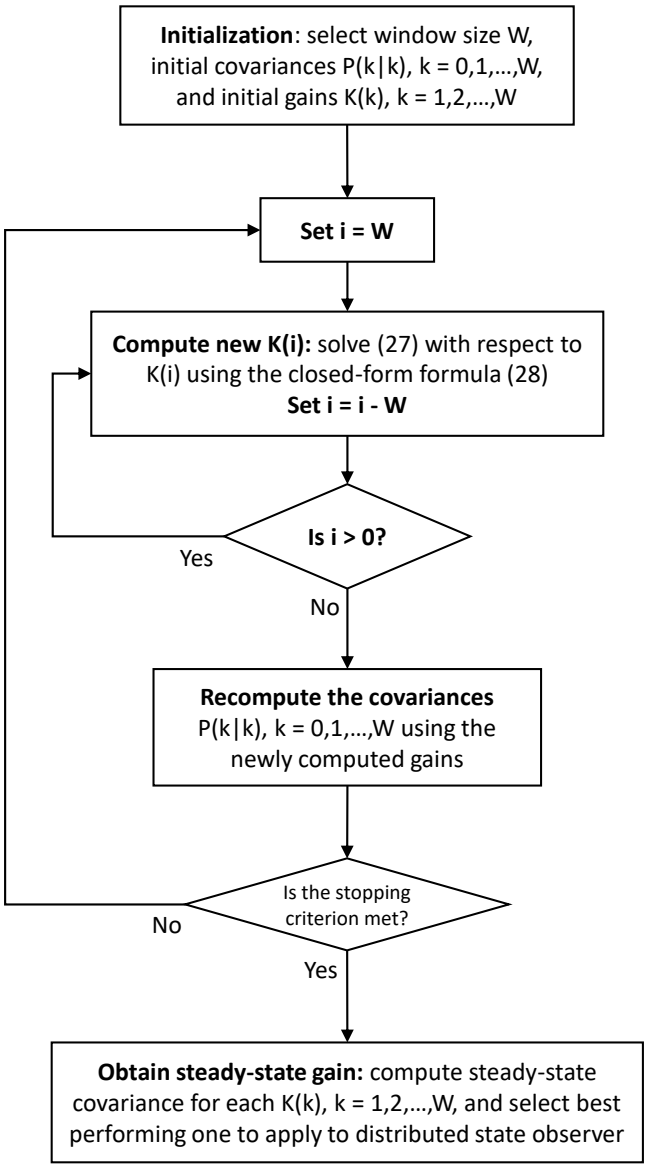

Figure 3: Schematic diagram for the finite-horizon algorithm detailed in Table 1. The stopping criterion can be either a fixed number of iterations or a minimum improvement between iterations.

Table 2: Comparison of steady-state performance for both methods

	One-step	Finite-horizon
$\mathrm{tr}(\mathbf{P}_\infty)$	26.375	21.914

In this case, the chosen $\mathbf{Z}$ preserves the order of the nonzero elements with respect to $\mathrm{vec}(\mathbf{K}(k))$. However, this is not mandatory as any permutation of the rows of $\mathbf{Z}$ will still yield the optimal gain using (28).

Remark 1. Note that the computation of $\mathbf{K}(k)$ from (28) only uses the covariances from steps up to k. Thus, by computing the gains in reverse order in the inner loop of the algorithm in Table 1, the covariances only need to be recomputed after computing all observer gains.

Remark 2. The algorithm detailed in Table 1 requires a set of initial gains and error covariances to begin its execution. As it is not necessary to initialize it with stabilizing gains, the simplest solution is to set all the $\mathbf{K}(k)$ to zero and compute the corresponding error covariances using (12) and (14). However, this can lead to numerical problems for unstable systems and large window sizes, as the error covariance necessarily grows unbounded with time. Another solution is to initialize the algorithm using a stabilizing gain found using a fast method such as the one detailed in the previous section. Finally, note that it is possible to initialize the algorithm with centralized gains, that is, not subjected to the sparsity constraint. While this is counterintuitive, those initial gains will be replaced by gains with the desired sparsity pattern after the first outer loop iteration.

5.1. Example of gain computation

To provide an example, this method was applied to the same dynamic system that was used in Section 4.1. Fig. 4 depicts the evolution of the sum of the trace of all covariances $\mathbf{P}(k|k),\ k = 1, 2, \ldots, W$ over the course of the execution of the algorithm of Table 1, for several different initial conditions. The different initial conditions for the algorithm, which are sequences of covariances and gains, were

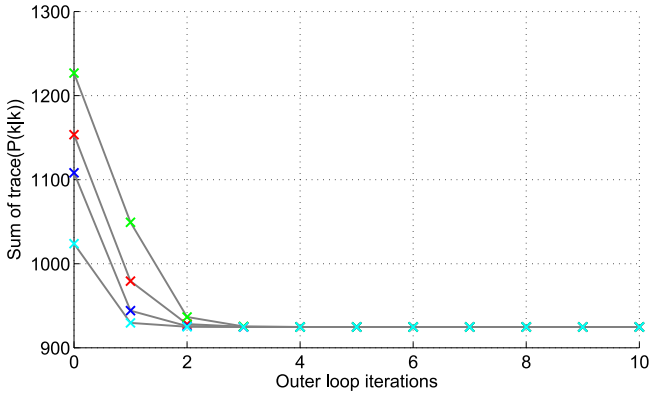

Figure 4: Convergence of the finite-horizon method error covariance for different initial conditions

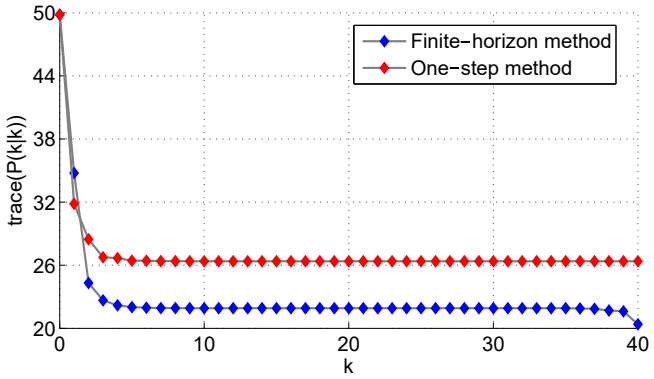

Figure 5: Comparison of the evolution of the error covariance for both methods

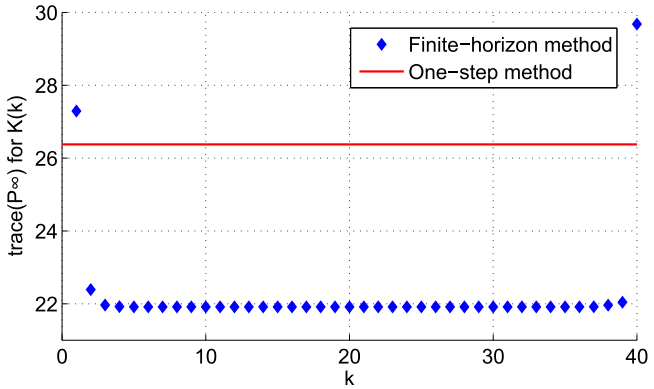

Figure 6: Steady-state performance for both methods

computed using the one-step method for different values of $\mathbf{Q}$ and $\mathbf{R}$, with the window size set to $W = 40$. As it can be seen, in all cases the algorithm converges very fast to the same value for the objective function.

Figures 5 and 6 provide a comparison of both methods. In Fig. 5, the evolution of the trace of $\mathbf{P}(k|k)$ is depicted for the one-step method and the finite-horizon method, starting at the same initial covariance $\mathbf{P}(0|0)$. Note that, while both solutions achieve a steady-state behavior, the finite-horizon method clearly outperforms the one-step method. The slight drop in the value of the trace of $\mathbf{P}(k|k)$ at the end for the finite-horizon case is due to the fact that, as the gains close to the end of the window are computed without taking into account the performance several steps ahead, allowing for an instantaneous gain in performance. Fig. 6 depicts the steady-state performance for each gain $\mathbf{K}(k),\ k = 1, 2, \ldots, W$, computed by propagating the covariance equations (12) and (14) for each constant gain $\mathbf{K}(k)$. As expected, the gains corresponding to the steady-state behavior observed in Fig. 5 perform the best, and offer a clear improvement in performance when compared with the one-step method. Note that the steady-state performance worsens significantly for the gains near the end of the chosen time window, as they do not take into account future performance as much as the gains computed for earlier time steps. Finally, Table 2 details the steady-state value of the trace of $\mathbf{P}(k|k)$ for both methods, and the best steady-state gain found using the finite-horizon method is

$$\mathbf{K}_\infty = \begin{bmatrix} -0.140 & 0 & 0.480 & 0.179 \\ 0 & 0.308 & 0 & 0.101 \\ 0 & 0 & 0.769 & 0 \\ 0.023 & 0.203 & -0.134 & 0 \\ 0.208 & -0.271 & 0 & 0.161 \end{bmatrix},$$

rounded to 3 decimal places.

The results detailed in this section as well as in Section 4.1 show that both proposed methods are able to compute stabilizing distributed filter gains for a generic system with an arbitrary given sparsity pattern. The difference between

the two methods is essentially a trade-off between computational efficiency and achieved performance. The one-step method gain is computed very quickly as the covariance equations only need to be propagated once until steady-state is achieved. On the other hand, the finite-horizon method achieves better steady-state performance, but at a higher computational cost. To begin with, it is necessary to iterate several times over the chosen finite time window for the algorithm to converge. Moreover, the closed-form solution for the optimization problem solved at each step is more complex than its one-step counterpart and requires more computations. Nevertheless, the finite-horizon method is still computationally efficient and scalable, but not as much as the one-step algorithm.

Remark 3. Regarding the choice of windows size W, the appropriate number will vary depending on the system model. For the case detailed in this section, Fig. 5 shows that $W = 40$ is sufficient to achieve steady-state behavior and thus obtain a suitable constant gain. However, for more complex systems such as the ones considered in the simulations of the next section, a larger window size might be necessary. Nevertheless, one can always verify if the chosen W is large enough by inspecting the evolution of the estimation error covariance, using for example a plot similar to the one in Fig. 5.

6. Simulation results for a formation of AUVs

This section details the application of the solutions presented in this chapter to the practical case of state estimation in a formation of AUVs, supported by simulation results to assess their performance. This application is similar to the one introduced in previous work such as Viegas et al. (2012), in particular the continuous-discrete filter implementation can also be found in Viegas et al. (2013). Nevertheless, it is briefly detailed in this section for the sake of completeness.

6.1. Application to a formation of AUVs

Consider a formation composed of N AUVs, and suppose that each vehicle has sensors mounted on-board which give access to either measurements of its own position in an inertial reference coordinate frame $\{I\}$, or measurements of its position relative to one or more AUVs in the vicinity. For the latter case, it is assumed that the AUVs also receive updated position estimates from other AUVs in the vicinity through communication. In underwater applications, the relative measurements can be provided by an ultra-short baseline (USBL) positioning system in an inverted configuration, see e.g. Morgado et al. (2011). The inertial measurements can be provided, e.g., by a long baseline (LBL), or by an USBL positioning system.

Let $\{B_i\}$ denote a coordinate frame attached to the i-th AUV, denoted in the sequel as the body-fixed coordinate frame associated with the i-th AUV. The linear motion of each AUV can be written as

$$\dot{\mathbf{p}}_i(t) = \boldsymbol{\mathcal{R}}_i(t)\mathbf{s}_i(t), \tag{30}$$

where $\mathbf{p}_i(t) \in \mathbb{R}^3$ is the inertial position of the i-th AUV, $\mathbf{s}_i(t) \in \mathbb{R}^3$ denotes its velocity relative to $\{I\}$, expressed in body-fixed coordinates, and $\boldsymbol{\mathcal{R}}_i(t) \in SO(3)$ is the rotation matrix from $\{B_i\}$ to $\{I\}$, which satisfies

$$\dot{\boldsymbol{\mathcal{R}}}_i(t) = \boldsymbol{\mathcal{R}}_i(t)\boldsymbol{\mathcal{S}}(\boldsymbol{\omega}_i(t)),$$

where $\boldsymbol{\omega}_i(t) \in \mathbb{R}^3$ is the angular velocity of $\{B_i\}$, expressed in body-fixed coordinates of the i-th AUV, and $\boldsymbol{\mathcal{S}}(\boldsymbol{\omega})$ is the skew-symmetric matrix such that $\boldsymbol{\mathcal{S}}(\boldsymbol{\omega})\boldsymbol{x}$ is the cross product $\boldsymbol{\omega} \times \boldsymbol{x}$. It is assumed that an attitude and heading reference system (AHRS) installed on-board each AUV provides measurements of both $\boldsymbol{\mathcal{R}}_i(t)$ and $\boldsymbol{\omega}_i(t)$. Additionally, suppose that each AUV has access to a linear acceleration measurement $\mathbf{a}_i(t) \in \mathbb{R}^3$, which follows

$$\mathbf{a}_i(t) = \dot{\mathbf{s}}_i(t) + \boldsymbol{\mathcal{S}}(\boldsymbol{\omega}_i(t))\mathbf{s}_i(t) - \mathbf{g}_i(t), \tag{31}$$

where $\mathbf{g}_i(t) \in \mathbb{R}^3$ is the acceleration of gravity, expressed in body-fixed coordinates of the i-th AUV. Even though its value is well-known, it is treated as an

unknown variable for performance reasons, see Batista et al. (2009) for further details. Its time derivative is given by

$$\dot{\mathbf{g}}_i(t) = -\boldsymbol{\mathcal{S}}(\boldsymbol{\omega}_i(t))\mathbf{g}_i(t). \tag{32}$$

For the first case, i.e., with inertial position readings, grouping equations (30), (31), and (32), and measuring the inertial position, yields the system

$$\begin{cases} \dot{\mathbf{p}}_i(t) = \boldsymbol{\mathcal{R}}_i(t)\mathbf{s}_i(t) \\ \dot{\mathbf{s}}_i(t) = -\boldsymbol{\mathcal{S}}(\boldsymbol{\omega}_i(t))\mathbf{s}_i(t) + \mathbf{g}_i(t) + \mathbf{a}_i(t) \\ \dot{\mathbf{g}}_i(t) = -\boldsymbol{\mathcal{S}}(\boldsymbol{\omega}_i(t))\mathbf{g}_i(t) \\ \mathbf{y}_i(t) = \mathbf{p}_i(t) \end{cases}.$$

Using in each vehicle the Lyapunov state transformation introduced in Batista et al. (2010),

$$\mathbf{x}_i(t) := \begin{bmatrix} \mathbf{x}_i^1(t) \\ \mathbf{x}_i^2(t) \\ \mathbf{x}_i^3(t) \end{bmatrix} = \boldsymbol{\mathcal{T}}_i(t) \begin{bmatrix} \mathbf{p}_i(t) \\ \mathbf{s}_i(t) \\ \mathbf{g}_i(t) \end{bmatrix},$$

with

$$\boldsymbol{\mathcal{T}}_i(t) = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{R}}_i(t) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{R}}_i(t) \end{bmatrix} \in \mathbb{R}^{9\times 9}, \tag{33}$$

which preserves stability and observability properties (Brockett (1970)), yields the LTI system

$$\begin{cases} \dot{\mathbf{x}}_i(t) = \mathbf{A}_{CT}\mathbf{x}_i(t) + \mathbf{B}_{CT}\mathbf{u}_i(t) \\ \mathbf{y}_i(t) = \mathbf{C}_{CT}\mathbf{x}_i(t) \end{cases}, \tag{34}$$

where $\mathbf{u}_i(t) := \boldsymbol{\mathcal{R}}_i(t)\mathbf{a}_i(t)$ is the input,

$$\mathbf{A}_{CT} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{9\times 9},\ \mathbf{B}_{CT} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{9\times 3},$$

and $\mathbf{C}_{CT} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{3\times 9}$.

To achieve a discrete-time model of the dynamics, suppose that the inertial position measurements are obtained with a constant sampling rate, with corresponding period T, and assume that the measurements from the AHRS and the accelerometer are obtained with a much faster sampling rate. Then, the system (34) can be described by the discrete-time LTI system (1), with

$$\begin{cases} \mathbf{A}_L = e^{\mathbf{A}_{CD}T} \\ \mathbf{B}_L = \mathbf{I}_9 \\ \mathbf{C}_L = \mathbf{C}_{CD} \end{cases}$$

and

$$\mathbf{u}_i(k) = \int_0^T e^{\mathbf{A}_{CD}(T-\tau)}\mathbf{B}_{CD}\mathbf{u}_i(t_k+\tau)d\tau,$$

in which t_k denotes the sampling instant of sample k.

Regarding the second case, i.e., when the AUV has access to relative position measurements and receives position estimates from the corresponding vehicles, a similar procedure can be carried out. The relative position measurements available to the i-th AUV are denoted by

$$\Delta\mathbf{p}_i(t) := \begin{bmatrix} \mathbf{p}_i(t) - \mathbf{p}_{d_i^1}(t) \\ \mathbf{p}_i(t) - \mathbf{p}_{d_i^2}(t) \\ \vdots \\ \mathbf{p}_i(t) - \mathbf{p}_{d_i^{\nu_i}}(t) \end{bmatrix} \in \mathbb{R}^{3\nu_i}. \tag{35}$$

Grouping equations (30), (31), and (32) and taking the relative position measurements (35) as the output yields the system

$$\begin{cases} \dot{\mathbf{p}}_i(t) = \boldsymbol{\mathcal{R}}_i(t)\mathbf{s}_i(t) \\ \dot{\mathbf{s}}_i(t) = -\boldsymbol{\mathcal{S}}(\boldsymbol{\omega}_i(t))\mathbf{s}_i(t) + \mathbf{g}_i(t) + \mathbf{a}_i(t) \\ \dot{\mathbf{g}}_i(t) = -\boldsymbol{\mathcal{S}}(\boldsymbol{\omega}_i(t))\mathbf{g}_i(t) \\ \mathbf{y}_i(t) = \Delta\mathbf{p}_i(t) \end{cases},$$

and applying the state transformation (33) yields the system

$$\begin{cases} \dot{\mathbf{x}}_i(t) = \mathbf{A}_{CT}\mathbf{x}_i(t) + \mathbf{B}_{CT}\mathbf{u}_i(t) \\ \mathbf{y}_i(t) = \mathbf{C}_{CT}^i\Delta\mathbf{x}_i(t) \end{cases},$$

where $\mathbf{A}_{CT}$ and $\mathbf{B}_{CT}$ are defined as in the previous case, and $\mathbf{C}_{CT}^{i} = \mathbf{I}_{\nu_i} \otimes \mathbf{C}_{CT} \in \mathbb{R}^{3\nu_i \times 9\nu_i}$. The equivalent discrete-time model can be described by (4), with $\mathbf{A}_L$, $\mathbf{B}_L$, and $\mathbf{u}_i(k)$ defined as in (34), and $\mathbf{C}_i = \mathbf{C}_{CT}^{i}$.

6.2. Filter implementation

Following the previous discussion, it is possible to design local state observers following (2) and (3) or (5), depending on the available measurements, and to apply the methods of Sections 4 and 5 to compute suitable gains.

In practice, the use of noisy attitude and angular velocity measurements will inject multiplicative noise into the computations, so it is advantageous to implement the local observers in the body-fixed coordinate frame of each respective AUV to minimize its effect. To do so, consider for the first case (inertial measurements) the following continuous-discrete representation: denoting the state estimate by $\hat{\mathbf{x}}_i(t) \in \mathbb{R}^9$, its update between samples of the output follows

$$\dot{\hat{\mathbf{x}}}_i(t) = \mathbf{A}_{CT}\hat{\mathbf{x}}_i(t) + \mathbf{B}_{CT}\mathbf{u}_i(t).$$

Then, when the output sample k is obtained at time t_k, update the state following

$$\hat{\mathbf{x}}_i(t_k^+) = \hat{\mathbf{x}}_i(t_k) + \mathbf{K}_i(\mathbf{y}_i(t_k) - \mathbf{C}_L\hat{\mathbf{x}}_i(t_k)).$$

Finally, reverse the Lyapunov state transformation (33) by defining new state estimates $\hat{\mathbf{z}}_i(t) = \boldsymbol{\mathcal{T}}_i^{\mathsf{T}}(t)\hat{\mathbf{x}}_i(t)$. The update between samples of the output becomes

$$\dot{\hat{\mathbf{z}}}_i(t) = \mathbf{A}_i(t)\hat{\mathbf{z}}_i(t) + \mathbf{B}_{CT}\mathbf{a}_i(t),$$

with

$$\mathbf{A}_i(t) = \begin{bmatrix} \mathbf{0} & \boldsymbol{\mathcal{R}}_i(t) & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{\mathcal{S}}(\boldsymbol{\omega}_i(t)) & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & -\boldsymbol{\mathcal{S}}(\boldsymbol{\omega}_i(t)) \end{bmatrix} \in \mathbb{R}^{9\times 9}.$$

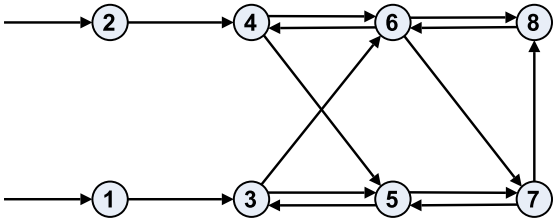

Figure 7: Measurement graph for the formation considered in the simulations

When an output sample is available, update the state estimate following

$$\hat{\mathbf{z}}_i(t_k^+) = \hat{\mathbf{z}}_i(t_k) + \boldsymbol{\mathcal{T}}_i^{\mathsf{T}}(t_k)\mathbf{K}_i(\mathbf{y}_i(t_k) - \mathbf{C}_L\hat{\mathbf{z}}_i(t_k)).$$

The same procedure can be carried out for the AUVs with relative measurements, yielding a similar continuous-discrete formulation. This representation allows for filter implementation in the body-fixed coordinate frame of each AUV and takes into account the different sampling rates between on-board instrumentation (accelerometer, AHRS) and the positioning measurements, and the observer gains $\mathbf{K}_i$ can be computed following the methods detailed in the previous sections, based on the discrete-time model derived in Section 6.1. For more on the theory and applications of discrete-time filters in guidance and navigation of vehicles, see e.g. Bar-Shalom et al. (2004) and Grewal et al. (2007).

6.3. Gain computation

Before computing the observer gains, both the measurement topology of the formation and the sensor noise need to be defined. In the simulations that were carried out, a formation composed of 8 AUVs was considered, and its measurement graph is depicted in Fig. 7. The positions measurements were sampled with constant period $T = 1$ s, to reflect what can be typically obtained from positioning systems based on acoustic measurements that are commonly used in underwater applications.

Regarding sensor noise, all the measurements were corrupted by additive, uncorrelated, zero-mean white Gaussian noise, with appropriate standard deviations given the equipment that would usually provide those measurements.

Table 3: Projected steady-state performance ($\mathrm{tr}(\mathbf{P}_\infty)$) for the four competing solutions

One-step	Finite-horizon	H_2 norm	Centralized
0.463	0.395	0.431	0.204

Regarding the positioning data, the standard deviation was set to 0.2 m for the relative measurements, and 0.1 m for the absolute measurements. Furthermore, some cross-correlation was added to the noise on the absolute measurements to account for possible similarities in the source of the data (such as both AUVs using GPS at the surface, or using a LBL positioning system and sharing the same set of landmarks), resulting in the following covariance matrix:

$$\mathbf{R}_0 = 0.01 \times \begin{bmatrix} 1 & 0.1 \\ 0.1 & 1 \end{bmatrix} \otimes \mathbf{I}_3.$$

For the other measurements, the standard deviation of the white noise was set as follows:

- Linear acceleration - 0.01 $\mathrm{m/s^2}$;
- Angular velocity - 0.05 °/s;
- Attitude, parametrized by Euler angles - 0.03 ° for the roll and pitch, 0.3 ° for the yaw.

The noise covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ used for computing the observer gains were set to

$$\begin{cases} \mathbf{Q} = 10^{-6}\mathbf{I} \\ \mathbf{R} = \mathrm{diag}(\mathbf{R}_0, 0.04\mathbf{I}, 0.04\mathbf{I}, \ldots, 0.04\mathbf{I}) \end{cases}.$$

Observer gains were first computed with the one-step method, and the resulting set of gains and covariance matrices were used to initialize the finite-horizon algorithm. The results are depicted in Fig. 8, which depicts the evolution of the updated error covariance $\mathbf{P}(k|k)$ for both methods. The initial error covariance was set to $\mathbf{P}(0|0) = \mathbf{0}$ to avoid large initial transients which would be detrimental to the clarity of the figure. As it can be seen, both methods

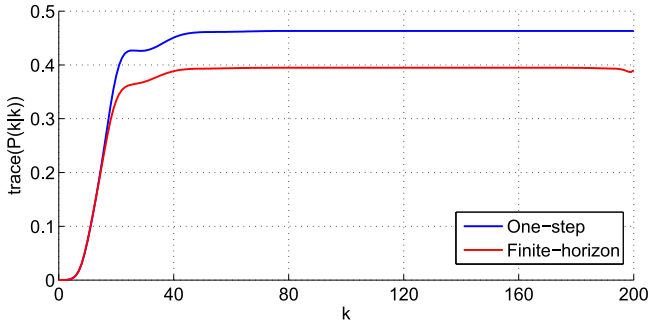

Figure 8: Evolution of the estimation error covariance for both methods

achieved steady-state behavior, and the finite-horizon algorithm shows a clear improvement in performance over the one-step method. To provide a comparison term, observer gains were obtained using the two methods introduced in Viegas et al. (2012) and adapted to the discrete-time case in Viegas et al. (2013). This method computes distributed observer gains through H_2 norm minimization, using an algorithm based on the $\mathcal{P} - \mathcal{K}$ iterations used in some controller synthesis problems (Fransson and Lennartson (2003)). Observer gains for a centralized Kalman filter were also computed to provide a second comparison term. The projected steady-state performance for each method is shown in Table 3. Comparing the different solutions, it is clear that the finite-horizon method achieves the best results of the three distributed gain computing methods, followed by the H_2 norm minimization algorithm. The distributed solutions are outperformed by the centralized Kalman filter, which is to be expected given that the latter is working with full information. It should be noted that both of the methods presented in this paper are much more efficient computationally than the aforementioned H_2 norm minimization algorithm for two reasons. Firstly, the H_2 norm method requires solving an optimization problem with LMI constraints at each step, while the methods proposed here feature closed-form solutions for the optimization problems solved in each iteration. Secondly, the H_2 norm method is very sensitive to initial conditions, that is, in general each initial gain used to start the algorithm will yield a different optimized

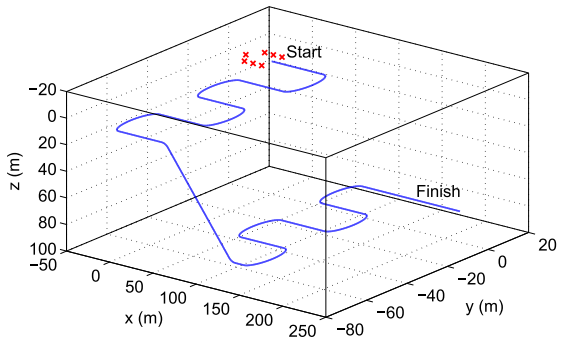

Figure 9: Initial positions of the AUVs and trajectory followed by the formation

gain, while in all tests that were carried out both the one-step method and the finite-horizon algorithm consistently yield an unique solution for each given system. Thus, while a single execution of the algorithms detailed in this paper is sufficient to compute the desired distributed gains, the H_2 norm minimization algorithm needs to be executed with different sets of initial conditions to obtain suitable filter gains.

6.4. Performance comparison

The initial positions of the agents and the trajectory followed by the formation during the simulation are depicted in Fig. 9. As for the local observers, and also the centralized Kalman filter, the initial values for all state estimates were set to zero, except for the ones corresponding to the acceleration of gravity. As $\mathbf{g}_i$ is known approximately, it was set to its initial real value in all estimators to speed up the convergence of the estimation error. Nevertheless, these could have also been initialized at zero with no bearing on the stability and steady-state performance of the distributed state observer.

Figures 10 and 11 depict, respectively, the initial evolution and steady-state behavior of the norm of the total estimation error, defined as the vector composed of all estimation error variables in the formation. As it can be seen, for all four competing solutions the estimation error converges to the vicinity of zero after a transient caused by the large error in the initial estimates, and remains

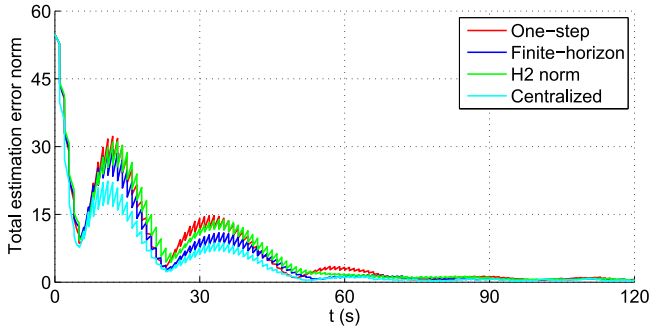

Figure 10: Initial evolution of the norm of the total estimation error for the 4 competing solutions

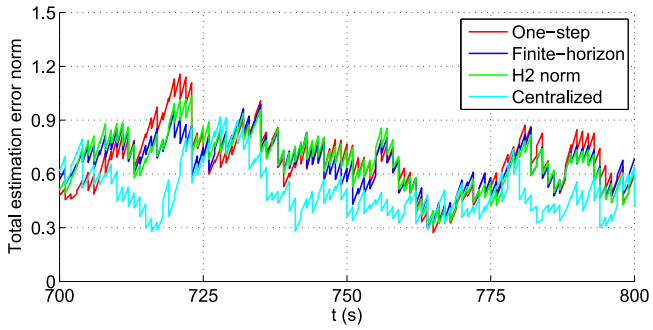

Figure 11: Detailed view of the norm of the steady-state estimation error for the 4 competing solutions

there over the course of the simulation. Note also the characteristic sawtooth shape of the graphs, due to the continuous-discrete implementation of the local observers. Between samples of the position measurements, the error dynamics are integrated in open-loop, which naturally increases the error. When measurements are obtained, the state estimates are updated, causing a sharp drop in the estimation error.

To supplement the graphical data, Table 4 depicts the sum of the variances of all estimation error variables in the formation, averaged over 1000 runs of the simulation. To compute these variances, only the updated state estimates were taken into account. As it can be seen, the finite-horizon method achieves the best performance out of the three distributed solutions, followed by the H_2 norm minimization algorithm, as was expected from the projected values of Table 3. The finite-horizon algorithm is clearly the best distributed design method of the three featured in these simulations, as it outperforms the other two methods and is more computationally efficient than its closest competitor, the H_2 norm minimization algorithm. Finally, the three distributed solutions are outperformed by the centralized Kalman filter. This is expected, as the centralized solution is operating with full information of all measurements and state estimates in the formations. However, the authors believe that the loss in performance is an acceptable trade-off for the much lower communication load in the formation, especially in underwater applications in which communication is oftentimes problematic. Note also that the results in terms of steady-state performance differ slightly from the projected values of Table 3. This can be explained by the fact that the model used for computing the gains does not take into account the multiplicative noise injected into the dynamics through the angular position and velocity measurements.

Remark 4. In the simulation results that were detailed in this section, it was assumed that the measurements were corrupted solely by additive zero-mean white Gaussian noise. However, as the proposed methods are based on very general models, they can be adapted fairly straightforwardly to more complex

Table 4: Measured steady-state performance (sum of the variances of all estimation error variables) for the four competing solutions, averaged over 1000 runs of the simulation

One-step	Finite-horizon	H_2 norm	Centralized
0.417	0.373	0.390	0.204

sensor models. In some cases, the dynamics of the vehicles can be augmented to include additional quantities such as accelerometer bias, see e.g. Batista et al. (2011). In other cases, these issues can be handled autonomously by each vehicle. One example of this is the issue of rate gyro bias, which can be handled within the AHRS, see e.g. Batista et al. (2012).

7. Further simulation results

While the previous section presented simulation results for the practical case of a formation of AUVs, this section details the results of simulations based on the randomly generated model used for the examples in Sections 4 and 5. The aim here is to provide a better comparison between the different distributed filtering solutions presented in this chapter, by working with a dynamic system that coincides exactly with the model used to design the filters. Thus, the issues present in the previous section, such as the presence of multiplicative noise, are avoided.

As mentioned above, the dynamic system in consideration for this section is the LTI system (10) with state and noise covariances matrices given by (21)-(24). The sparsity pattern was kept the same as in the aforementioned examples, that is, the one detailed in (25). However, the system is unstable and the exponential increase in the system state might cause numerical problems during the simulations. To avoid this issue, a simple control law is introduced to keep the state within certain values. Set $\mathbf{B} = \mathbf{I}$, and the input $\boldsymbol{u}(k)$ is zero when the norm of the state $\boldsymbol{x}(k)$ is below a certain threshold. When $\|\boldsymbol{x}(k)\|$ is above the

Table 5: Projected steady-state performance ($\mathrm{tr}(\mathbf{P}_\infty)$) for the four competing solutions

One-step	Finite-horizon	H_2 norm	Centralized
26.375	21.914	22.748	10.009

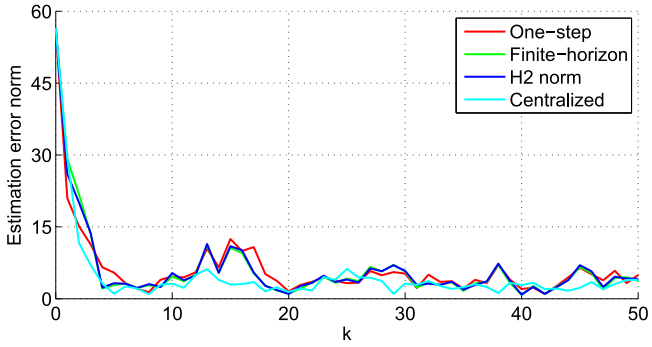

Figure 12: Initial evolution of the estimation error norm for the four competing solutions

chosen threshold, the input is set to

$$\boldsymbol{u}(k) = -0.9\mathbf{A}\boldsymbol{x}(k).$$

For the simulations detailed here, the threshold was set to 20. Notice that this does not have any impact in the estimation performance whatsoever, neither positively nor negatively.

Similarly to the previous section, the simulations were carried for the two solutions presented in this paper, as well as the H_2 norm minimization algorithm and a centralized Kalman filter to provide comparison terms. The projected performance of the four competing methods is detailed in Table 5. As it can be seen, the H_2 norm minimization algorithm is able to outperform the one-step method, but does not manage to reach the performance of the finite-horizon method. The centralized Kalman filter significantly outperforms the distributed solutions, which is to be expected given that each state estimate is updated using all the available measurements.

For the simulations, the initial state $\boldsymbol{x}(0)$ and state estimate $\hat{\boldsymbol{x}}(0)$ were sampled from normal distributions with zero mean and standard deviations $\sigma = 10$

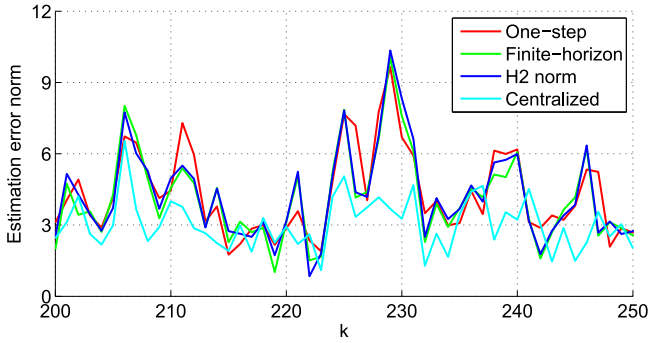

Figure 13: Steady-state behavior of the estimation error norm for the four competing solutions

Table 6: Measured steady-state performance (sum of the variances of all estimation error variables) for the four competing solutions, averaged over 1000 runs of the simulation

One-step	Finite-horizon	H_2 norm	Centralized
26.363	21.909	22.744	10.008

and $\sigma = 50$, respectively, to create a large initial estimation error. The results of the simulations are depicted in Figs. 12 and 13, and in Table 6. Fig. 12 depicts the initial evolution of the norm of the estimation error, while Fig. 13 depicts its steady-state behavior, and Table 6 details the sum of the steady-state variances of all estimation error variables, averaged over 1000 runs of the simulation. The values in Table 6 coincide almost exactly with the projected values in Table 5, which was expected given that in this case the model used to design the filters is the actual simulated system, without unmodeled behaviors such as multiplicative noise.

8. Conclusions

This paper addressed the problem of distributed state estimation in a multi-vehicle framework. In the scenario envisioned in this work, each vehicle aims to estimate its own state by implementing a local state observer which relies on locally available measurements and limited communication with other vehicles in the vicinity. The dynamics of the problem were formulated as a more gen-

eral discrete-time Kalman filtering problem with a sparsity constraint on the gain and, based on this formulation, two different algorithms for computation of steady-state observer gains for arbitrary fixed measurement topologies were introduced. The first method is an adaptation of the Kalman filter iterations to the distributed case and allows for computationally efficient computation of filter gains. The second method consists in the optimization of the time-varying distributed Kalman filter over a finite time window to approximate steady-state behavior and compute well-performing steady-state observer gains. The chosen framework was then particularized to the practical case of a formation of AUVs, and simulation results were presented to assess the performance of the proposed solutions. It was shown that the finite-horizon method outperforms the H_2 norm minimization algorithm used as a comparison term, in both filtering performance and computational cost.

Acknowledgments

The work presented in this paper was supported by the Macao Science & Technology Development Fund (FDCT) through Grant FDCT/065/2014/A, by project MYRG2015-00127-FST of the University of Macau, and by the Fundação para a Ciência e a Tecnologia (FCT) through ISR under contract LARSyS FCT [UID/EEA/50009/2013] and through IDMEC, under contract LAETA FCT [UID/EMS/50022/2013]. The work of D. Viegas was partially supported by the Ph.D. Student Scholarship SFRH/BD/71486/2010, from FCT.

Anderson, B.D.O., Moore, J.B., 2012. Optimal filtering. Courier Dover Publications.

Antonelli, G., 2013. Interconnected dynamic systems: An overview on distributed control. IEEE Control Systems 33, 76–88.

Antonelli, G., Arrichiello, F., Caccavale, F., Marino, A., 2014. Decentralized time-varying formation control for multi-robot systems. The International Journal of Robotics Research 33, 1029–1043.

Antonelli, G., Arrichiello, F., Chiaverini, S., Sukhatme, G., 2010. Observability analysis of relative localization for AUVs based on ranging and depth measurements, in: Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA), Anchorage, Alaska, USA. pp. 4286–4271.

Bahr, A., Leonard, J., Fallon, M., 2009. Cooperative localization for autonomous underwater vehicles. The International Journal of Robotics Research 28, 714–728.

Bakule, L., 2008. Decentralized control: An overview. Annual reviews in control 32, 87–98.

Bakule, L., 2014. Decentralized control: status and outlook. Annual Reviews in Control 38, 71–80.

Bamieh, B., Paganini, F., Dahleh, M.A., 2002. Distributed control of spatially invariant systems. IEEE Transactions on automatic control 47, 1091–1107.

Bamieh, B., Voulgaris, P.G., 2005. A convex characterization of distributed control problems in spatially invariant systems with communication constraints. Systems & Control Letters 54, 575–583.

Bar-Shalom, Y., Li, X.R., Kirubarajan, T., 2004. Estimation with applications to tracking and navigation: theory algorithms and software. John Wiley & Sons.

Batista, P., Silvestre, C., Oliveira, P., 2009. Position and velocity optimal zensor-based navigation filters for UAVs, in: Proceedings of the 2009 American Control Conference (ACC), Saint Louis, USA. pp. 5404–5409.

Batista, P., Silvestre, C., Oliveira, P., 2010. Optimal position and velocity navigation filters for autonomous vehicles. Automatica 46, 767–774.

Batista, P., Silvestre, C., Oliveira, P., 2011. On the observability of linear motion quantities in navigation systems. Systems & Control Letters 60, 101–110.

Batista, P., Silvestre, C., Oliveira, P., 2012. Sensor-based globally asymptotically stable filters for attitude estimation: Analysis, design, and performance evaluation. IEEE Transactions on Automatic Control 57, 2095–2100.

Bender, J.G., 1991. An overview of systems studies of automated highway systems. IEEE Transactions on Vehicular Technology 40, 82–99.

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

Bompart, V., Noll, D., Apkarian, P., 2007. Second-order nonsmooth optimization for H_∞ synthesis. Numerische Mathematik 107, 433–454.

Brockett, R.W., 1970. Finite Dimensional Linear Systems. Wiley and Sons.

Bullo, F., Cortes, J., Martinez, S., 2009. Distributed control of robotic networks: a mathematical approach to motion coordination algorithms. Princeton University Press.

Cao, M., Yu, C., Anderson, B.D.O., 2011. Formation control using range-only measurements. Automatica 47, 776–781.

Cao, Y., Yu, W., Ren, W., Chen, G., 2013. An overview of recent progress in the study of distributed multi-agent coordination. IEEE Transactions on Industrial informatics 9, 427–438.

Cortés, J., Martínez, S., Bullo, F., 2006. Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. IEEE Transactions on Automatic Control 51, 1289–1298.

Curtin, T.B., Bellingham, J.G., Webb, D., 1993. Autonomous oceanographic sampling networks. Oceanography 6, 86–94.

Davison, E.J., Aghdam, A.G., 2014. Decentralized control of large-scale systems. Springer Publishing Company, Incorporated.

Dörfler, F., Jovanović, M.R., Chertkov, M., Bullo, F., 2014. Sparsity-promoting optimal wide-area control of power networks. IEEE Transactions on Power Systems 29, 2281–2291.

Dunbar, W.B., Murray, R.M., 2006. Distributed receding horizon control for multi-vehicle formation stabilization. Automatica 42, 549–558.

Fransson, C.M., Lennartson, B., 2003. Low order multicriteria H_∞ design via bilinear matrix inequalities, in: Proceedings of the 42nd IEEE Conference on Decision and Control (CDC), Maui, Hawaii, USA. pp. 5161–5167.

Gadre, A., Stilwell, D., 2005. A complete solution to underwater navigation in the presence of unknown currents based on range measurements from a single location, in: Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Edmonton, Alberta, Canada. pp. 1420–1425.

Giulietti, F., Pollini, L., Innocenti, M., 2000. Autonomous formation flight. IEEE Control Systems Magazine 20, 34–44.

Grewal, M.S., Weill, L.R., Andrews, A.P., 2007. Global positioning systems, inertial navigation, and integration. John Wiley & Sons.

Healey, A.J., 2001. Application of formation control for multi-vehicle robotic minesweeping, in: Proceedings of the 40th IEEE Conference on Decision and Control (CDC), Orlando, FL, USA. pp. 1497–1502.

Kumar, V., Rus, D., Sukhatme, G.S., 2008. Networked robots, in: Springer Handbook of Robotics. Springer, pp. 943–958.

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

Lessard, L., Lall, S., 2016. Convexity of decentralized controller synthesis. IEEE Transactions on Automatic Control 61, 3122–3127.

Lewis, F.L., Zhang, H., Hengster-Movric, K., Das, A., 2013. Cooperative control of multi-agent systems: optimal and adaptive design approaches. Springer Science & Business Media.

Lin, F., Fardad, M., Jovanovic, M.R., 2011. Augmented lagrangian approach to design of structured optimal state feedback gains. IEEE Transactions on Automatic Control 56, 2923–2929.

Liu, K., Lim, H.B., Frazzoli, E., Ji, H., Lee, V.C.S., 2014. Improving positioning accuracy using GPS pseudorange measurements for cooperative vehicular localization. IEEE Transactions on Vehicular Technology 63, 2544–2556.

Mahmoud, M.S., 2010. Decentralized control and filtering in interconnected dynamical systems. CRC Press.

Morgado, M., Batista, P., Oliveira, P., Silvestre, C., 2011. Position USBL/DVL sensor-based navigation filter in the presence of unknown ocean currents. Automatica 47, 2604–2614.

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

Olfati-Saber, R., 2007. Distributed kalman filtering for sensor networks, in: Proceedings of the 46th IEEE Conference on Decision and Control (CDC), IEEE, New Orleans, LA, USA. pp. 5492–5498.

Olfati-Saber, R., Murray, R.M., 2004. Consensus problems in networks of agents with switching topology and time-delays. IEEE Transactions on automatic control 49, 1520–1533.

Parker, L.E., 2008. Multiple mobile robot systems, in: Springer Handbook of Robotics. Springer, pp. 921–941.

Pasqualetti, F., Carli, R., Bullo, F., 2012. Distributed estimation via iterative projections with application to power network monitoring. Automatica 48, 747 – 758.

Rantzer, A., 2015. Scalable control of positive systems. European Journal of Control 24, 72–80.

Rotkowitz, M., Lall, S., 2006. A characterization of convex problems in decentralized control. IEEE Transactions on Automatic Control 51, 274–286.

Shah, P., Parrilo, P.A., 2008. A partial order approach to decentralized control, in: Proceedings of the 47th IEEE Conference on Decision and Control (CDC), IEEE, Cancun, Mexico. pp. 4351–4356.

Shah, P., Parrilo, P.A., 2013. H_2-optimal decentralized control over posets: a state-space solution for state-feedback. IEEE Transactions on Automatic Control 58, 3084–3096.

Siljak, D.D., 2011. Decentralized control of complex systems. Courier Corporation.

Sousa, R., Oliveira, P., Gaspar, T., 2009. Joint positioning and navigation aiding systems for multiple underwater robots, in: Proceedings of the 8th IFAC Conference on Manoeuvring and Control of Marine Craft (MCMC), Guaruja, Brazil. pp. 167–172.

Tanaka, T., Langbort, C., 2011. The bounded real lemma for internally positive systems and H_∞ structured static state feedback. IEEE transactions on automatic control 56, 2218–2223.

Viegas, D., Batista, P., Oliveira, P., Silvestre, C., 2012. Decentralized H_2 observers for position and velocity estimation in vehicle formations with fixed topologies. Systems & Control Letters 61, 443–453.

Viegas, D., Batista, P., Oliveira, P., Silvestre, C., 2013. GAS decentralized navigation filters in a continuous-discrete fixed topology framework, in: Proceedings of the 14th IEEE Mediterranean Conference on Control and Automation (MED), IEEE, Platanias-Chania, Crete, Greece. pp. 1286–1291.

Viegas, D., Batista, P., Oliveira, P., Silvestre, C., 2014. Position and velocity filters for ASC/I-AUV tandems based on single range measurements. Journal of Intelligent & Robotic Systems 74, 745–768.

Viegas, D., Batista, P., Oliveira, P., Silvestre, C., 2015a. Decentralized state observers for range-based position and velocity estimation in acyclic formations with fixed topologies. International Journal of Robust and Nonlinear Control (available online) .

Viegas, D., Batista, P., Oliveira, P., Silvestre, C., 2017. Discrete-time distributed Kalman filter design for multi-vehicle systems, in: 2017 American Control Conference, Seattle, WA, USA. p. (in print).

Viegas, D., Batista, P., Oliveira, P., Silvestre, C., Chen, C.L.P., 2015b. Distributed state estimation for linear multi-agent systems with time-varying measurement topology. Automatica 54, 72–79.

Wallis, W.D., 2007. A beginner's guide to graph theory. 2nd ed., Birkhauser.

Webster, S., Eustice, R., Singh, H., Whitcomb, L., 2012. Advances in single-beacon one-way-travel-time acoustic navigation for underwater vehicles. The International Journal of Robotics Research 31, 935–650.

West, D.B., 2001. Introduction to graph theory. 2nd ed., Pearson.

Witsenhausen, H., 1968. A counterexample in stochastic optimum control. SIAM Journal of Control 6, 131–147.

Wolfe, J., Chichka, D., Speyer, J., 1996. Decentralized controllers for unmanned aerial vehicle formation flight, in: Proceedings of the AIAA Conference on Guidance, Navigation, and Control (GNC), San Diego, CA, USA. pp. 96–3833.

Yanakiev, D., Kanellakopoulos, I., 1996. A simplified framework for string instability analysis in AHS, in: Proceedings of the 13th IFAC World Congress, San Francisco, CA, USA. pp. 177–182.

Yuan, Y., Tanner, H.G., 2010. Sensor graphs for guaranteed cooperative localization performance. Control and Intelligent Systems 38, 32–39.

Zhai, G., Ikeda, M., Fujisaki, Y., 2001. Decentralized H_∞ controller design: a matrix inequality approach using a homotopy method. Automatica 37, 565–572.

Zhou, X.S., Roumeliotis, S.I., 2008. Robot-to-robot relative pose estimation from range measurements. IEEE Transactions on Robotics 24, 1379–1393.

Appendix A. Derivation of the closed-form optimal gain, one-step method

The optimal gain for the one-step method is obtained by solving the optimization problem

$$\begin{aligned} \underset{\mathbf{K}(k+1)\in\mathbb{R}^{n\times o}}{\text{minimize}} \quad & \text{tr}(\mathbf{P}(k+1|k+1)) \\ \text{subject to} \quad & \mathbf{K}(k+1) \in \text{Sparse}(\mathbf{E}) \end{aligned} \tag{A.1}$$

The covariance $\mathbf{P}(k+1|k+1)$ is given by

$$\begin{aligned} &\mathbf{P}(k+1|k+1) = \\ &\mathbf{K}(k+1)\mathbf{R}\mathbf{K}^\mathsf{T}(k+1) + (\mathbf{I} - \mathbf{K}(k+1)\mathbf{C})\mathbf{P}(k+1|k)(\mathbf{I} - \mathbf{K}(k+1)\mathbf{C})^\mathsf{T}. \end{aligned} \tag{A.2}$$

Using (A.2), it can be shown that

$$\frac{\partial \text{tr}(\mathbf{P}(k+1|k+1))}{\partial \mathbf{K}} = -2\mathbf{P}(k+1|k)\mathbf{C}^\mathsf{T} + 2\mathbf{K}(k+1)\mathbf{S}(k+1), \tag{A.3}$$

in which

$$\mathbf{S}(k+1) = \mathbf{C}\mathbf{P}(k+1|k)\mathbf{C}^\mathsf{T} + \mathbf{R}.$$

The optimization problem (A.1) can be seen, instead of as a quadratic problem with a matrix variable $\mathbf{K}(k+1)$ subject to a sparsity constraint, as an unconstrained quadratic problem whose optimization variables are the nonzero

entries of $\mathbf{K}(k+1)$. Thus, the optimal solution can be found by differentiating the objective function with respective to these nonzero entries and equaling to zero. Define a set $\mathcal{X}$ of integer pairs of the form (i,j) to index the nonzero entries in $\mathbf{K}(k+1)$, that is,

$$\begin{cases} (i,j) \in \mathcal{X} & \text{if } [\mathbf{E}]_{ij} \neq 0 \\ (i,j) \notin \mathcal{X} & \text{otherwise} \end{cases}, \quad i = 1,\ldots,n,\ j = 1,\ldots,o. \tag{A.4}$$

Using (A.3) and taking into account the sparsity constraint, a set of $n \times o$ linear equations is obtained,

$$\begin{cases} \boldsymbol{l}_i^{\mathsf{T}}(\mathbf{P}(k+1|k)\mathbf{C}^{\mathsf{T}} - \mathbf{K}(k+1)\mathbf{S}(k+1))\boldsymbol{l}_j = 0 & \text{for all } (i,j) \in \mathcal{X} \\ \boldsymbol{l}_i^{\mathsf{T}}\mathbf{K}(k+1)\boldsymbol{l}_j = 0 & \text{for all } (i,j) \notin \mathcal{X} \end{cases}. \tag{A.5}$$

Now, consider an entire row of $\mathbf{K}(k+1)$. From the first part of (A.5), it follows that

$$\sum_{j}^{(i,j)\in\mathcal{X}} \boldsymbol{l}_i^{\mathsf{T}}(\mathbf{P}(k+1|k)\mathbf{C}^{\mathsf{T}} - \mathbf{K}(k+1)\mathbf{S}(k+1))\boldsymbol{\mathcal{L}}_j = 0. \tag{A.6}$$

Using the notation of Section 4, (A.6) can be written as

$$\boldsymbol{l}_i^{\mathsf{T}}\mathbf{K}(k+1)\mathbf{S}(k+1)\boldsymbol{\mathcal{M}}_i = \boldsymbol{l}_i^{\mathsf{T}}\mathbf{P}(k+1|k)\mathbf{C}^{\mathsf{T}}\boldsymbol{\mathcal{M}}_i. \tag{A.7}$$

The left-hand side of (A.7) is then rewritten as

$$\boldsymbol{l}_i^{\mathsf{T}}\mathbf{K}(k+1)\mathbf{S}(k+1)\boldsymbol{\mathcal{M}}_i = \boldsymbol{l}_i^{\mathsf{T}}\mathbf{K}(k+1)(\mathbf{I} + \boldsymbol{\mathcal{M}}_i - \boldsymbol{\mathcal{M}}_i)\mathbf{S}(k+1)\boldsymbol{\mathcal{M}}_i$$

and, using the second part of (A.5), it can be simplified to

$$\boldsymbol{l}_i^{\mathsf{T}}\mathbf{K}(k+1)\mathbf{S}(k+1)\boldsymbol{\mathcal{M}}_i = \boldsymbol{l}_i^{\mathsf{T}}\mathbf{K}(k+1)\boldsymbol{\mathcal{M}}_i\mathbf{S}(k+1)\boldsymbol{\mathcal{M}}_i. \tag{A.8}$$

Substituting (A.8) in (A.7) yields

$$\boldsymbol{l}_i^{\mathsf{T}}\mathbf{K}(k+1)\boldsymbol{\mathcal{M}}_i\mathbf{S}(k+1)\boldsymbol{\mathcal{M}}_i = \boldsymbol{l}_i^{\mathsf{T}}\mathbf{P}(k+1|k)\mathbf{C}^{\mathsf{T}}\boldsymbol{\mathcal{M}}_i. \tag{A.9}$$

However, (A.9) cannot be solved for $\mathbf{K}(k+1)$ yet as the matrix $\boldsymbol{\mathcal{M}}_i\mathbf{S}(k+1)\boldsymbol{\mathcal{M}}_i$ is singular. To circumvent this, note that, using the second part of (A.5), the left-hand side of (A.9) can be rewritten as

$$\boldsymbol{l}_i^{\mathsf{T}}\mathbf{K}(k+1)\boldsymbol{\mathcal{M}}_i\mathbf{S}(k+1)\boldsymbol{\mathcal{M}}_i = \boldsymbol{l}_i^{\mathsf{T}}\mathbf{K}(k+1)(\mathbf{I} - \boldsymbol{\mathcal{M}}_i + \boldsymbol{\mathcal{M}}_i\mathbf{S}(k+1)\boldsymbol{\mathcal{M}}_i). \tag{A.10}$$

Then, substituting (A.10) in (A.9) yields

$$\boldsymbol{l}_i^{\mathsf{T}}\mathbf{K}(k+1)(\mathbf{I}-\boldsymbol{\mathcal{M}}_i+\boldsymbol{\mathcal{M}}_i\mathbf{S}(k+1)\boldsymbol{\mathcal{M}}_i)=\boldsymbol{l}_i^{\mathsf{T}}\mathbf{P}(k+1|k)\mathbf{C}^{\mathsf{T}}\boldsymbol{\mathcal{M}}_i,$$

which implies that the i-th row of the optimal one-step filter gain is given by

$$\boldsymbol{l}_i^{\mathsf{T}}\mathbf{K}(k+1)=\boldsymbol{l}_i^{\mathsf{T}}\mathbf{P}(k+1|k)\mathbf{C}^{\mathsf{T}}\boldsymbol{\mathcal{M}}_i(\mathbf{I}-\boldsymbol{\mathcal{M}}_i+\boldsymbol{\mathcal{M}}_i\mathbf{S}(k+1)\boldsymbol{\mathcal{M}}_i)^{-1}. \quad \text{(A.11)}$$

Finally, sum (A.11) for all rows of $\mathbf{K}(k+1)$ to obtain the closed-form expression

$$\mathbf{K}(k+1)=\sum_{i=1}^{n}\boldsymbol{\mathcal{L}}_i\mathbf{P}(k+1|k)\mathbf{C}^{\mathsf{T}}\boldsymbol{\mathcal{M}}_i(\mathbf{I}-\boldsymbol{\mathcal{M}}_i+\boldsymbol{\mathcal{M}}_i\mathbf{S}(k+1)\boldsymbol{\mathcal{M}}_i)^{-1}.$$

Appendix B. Derivation of the closed-form optimal gain, finite horizon algorithm

This appendix details the derivation of the closed-form optimal solution for the optimization problem (27). First, note that the optimization problem

$$\begin{aligned}\underset{\mathbf{K}(k)\in\mathbb{R}^{n\times o}}{\text{minimize}} \quad & \sum_{i=k}^{W}\text{tr}(\mathbf{P}(i|i)) \\ \text{subject to} \quad & \mathbf{K}(k)\in\text{Sparse}(\mathbf{E})\end{aligned} \quad \text{(B.1)}$$

has the same optimal solution as (27), as the covariances of previous steps are not affected by the value of $\mathbf{K}(k)$. As for the one-step case, the optimization problem (B.1) is solved by differentiating the objective function with respect to the nonzero entries in $\mathbf{K}(k)$ and equaling to zero. Without loss of generality, assume that $\mathbf{A}$ is invertible. Using (12) and (14), it can be shown that, for $j\geq k$, the error covariance can be expressed as

$$\begin{aligned}&\mathbf{P}(j|j)=\boldsymbol{\Gamma}(k,j)\mathbf{P}(k-1|k-1)\boldsymbol{\Gamma}^{\mathsf{T}}(k,j)+ \\ &\sum_{i=k}^{j}\boldsymbol{\Gamma}(i+1,j)\mathbf{K}(i)\mathbf{R}\mathbf{K}^{\mathsf{T}}(i)\boldsymbol{\Gamma}^{\mathsf{T}}(i+1,j)+\sum_{i=k}^{j}\boldsymbol{\Gamma}(i,j)\mathbf{A}^{-1}\mathbf{Q}(\mathbf{A}^{-1})^{\mathsf{T}}\boldsymbol{\Gamma}^{\mathsf{T}}(i,j),\end{aligned} \quad \text{(B.2)}$$

where $\mathbf{\Gamma}(k_i, k_f)$ is defined as in (29). Taking the trace of both sides of (B.2) and differentiating with respect to $\mathbf{K}(k)$ yields

$$\frac{\partial \operatorname{tr}(\mathbf{P}(j|j))}{\partial \mathbf{K}(k)} = 2\left(\mathbf{\Gamma}^{\mathsf{T}}(k+1,j)\mathbf{\Gamma}(k+1,j)\right)\left(\mathbf{K}(k)\mathbf{S}(k) - \mathbf{P}(k|k-1)\mathbf{C}^{\mathsf{T}}\right). \tag{B.3}$$

Summing (B.3) over $j = k, k+1, \ldots, W$ yields

$$\frac{\partial \sum_{j=k}^{W} \operatorname{tr}(\mathbf{P}(j|j))}{\partial \mathbf{K}(k)} = 2\mathbf{\Lambda}(k+1)\left(\mathbf{K}(k)\mathbf{S}(k) - \mathbf{P}(k|k-1)\mathbf{C}^{\mathsf{T}}\right).$$

Following the same procedure as in the previous appendix, the following set of linear equations is obtained,

$$\begin{cases} \boldsymbol{l}_i^{\mathsf{T}}\mathbf{\Lambda}(k+1)\left(\mathbf{K}(k)\mathbf{S}(k) - \mathbf{P}(k|k-1)\mathbf{C}^{\mathsf{T}}\right)\boldsymbol{l}_j = 0 & \text{for all } (i,j) \in \mathcal{X} \\ \boldsymbol{l}_i^{\mathsf{T}}\mathbf{K}(k+1)\boldsymbol{l}_j = 0 & \text{for all } (i,j) \notin \mathcal{X} \end{cases}, \tag{B.4}$$

where $\mathcal{X}$ is defined as in (A.4). Using vectorization operators and the selection matrix $\mathbf{Z}$ as defined in Section 5, the first part of (B.4) can be rewritten as

$$\mathbf{Z}\operatorname{vec}(\mathbf{\Lambda}(k+1)\mathbf{K}(k)\mathbf{S}(k)) = \mathbf{Z}\operatorname{vec}(\mathbf{\Lambda}(k+1)\mathbf{P}(k|k-1)\mathbf{C}^{\mathsf{T}}). \tag{B.5}$$

Note also that the second part of (B.4) implies that

$$\operatorname{vec}(\mathbf{K}(k)) = \mathbf{Z}^{\mathsf{T}}\mathbf{Z}\operatorname{vec}(\mathbf{K}(k)). \tag{B.6}$$

The left-hand side of (B.5) can be rewritten as

$$\begin{aligned} &\mathbf{Z}\operatorname{vec}(\mathbf{\Lambda}(k+1)\mathbf{K}(k)\mathbf{S}(k)) = \\ &\mathbf{Z}(\mathbf{S}(k)\otimes\mathbf{\Lambda}(k+1))\operatorname{vec}(\mathbf{K}(k)) = \mathbf{Z}(\mathbf{S}(k)\otimes\mathbf{\Lambda}(k+1))\mathbf{Z}^{\mathsf{T}}\mathbf{Z}\operatorname{vec}(\mathbf{K}(k)). \end{aligned} \tag{B.7}$$

Substituting (B.7) in (B.5) yields

$$\mathbf{Z}(\mathbf{S}(k)\otimes\mathbf{\Lambda}(k+1))\mathbf{Z}^{\mathsf{T}}\mathbf{Z}\operatorname{vec}(\mathbf{K}(k)) = \mathbf{Z}\operatorname{vec}(\mathbf{\Lambda}(k+1)\mathbf{P}(k|k-1)\mathbf{C}^{\mathsf{T}}),$$

which implies that

$$\mathbf{Z}\operatorname{vec}(\mathbf{K}(k)) = \left(\mathbf{Z}(\mathbf{S}(k)\otimes\mathbf{\Lambda}(k+1))\mathbf{Z}^{\mathsf{T}}\right)^{-1}\mathbf{Z}\operatorname{vec}(\mathbf{\Lambda}(k+1)\mathbf{P}(k|k-1)\mathbf{C}^{\mathsf{T}}).$$

Finally, multiplying by $\mathbf{Z}^\mathsf{T}$ on both sides and using (B.6) yields the closed-form solution

$$\mathrm{vec}(\mathbf{K}(k)) = \mathbf{Z}^\mathsf{T}\left(\mathbf{Z}(\mathbf{S}(k)\otimes\mathbf{\Lambda}(k+1))\mathbf{Z}^\mathsf{T}\right)^{-1}\mathbf{Z}\,\mathrm{vec}(\mathbf{\Lambda}(k+1)\mathbf{P}(k|k-1)\mathbf{C}^\mathsf{T}).$$