Chapter 12
Delays in Distributed Estimation and Control over Communication Networks

Pablo Millán, Luis Orihuela and Isabel Jurado

Abstract This chapter introduces a distributed estimation and control technique with application to networked systems. The problem consists of monitoring and controlling a large-scale plant using a network of agents which collaborate exchanging information over an unreliable network. We propose an agent-based scheme based on an estimation structure that combines local measurements of the plant with remote information received from neighboring agents. We discuss the design of stabilizing distributed controllers and observers when the interagent communication is affected by delays and packet dropouts. Some simulations will be shown to illustrate the performance of this approach.

12.1 Introduction

Wireless communication network is a technology that has been attracting interest in the past decade due to its large variety of applications and utilities. One of the most important characteristic of this kind of systems is that allows the integration of different devices, providing flexibility, robustness, and ease of configuration of the system.

The devices interconnected in the wireless network (WN) are agents that may have sensing and actuation interfaces, as well as computation and communication capabilities. These particular systems are very useful in applications as process control systems [31, 41], mobile vehicles [4, 6, 10], tracking and surveillance [35, 38], or water delivery control [3, 17].

Among many advantages of this kind of systems [30], the capability of each agent to cooperate makes WNs a powerful network for facing complex problems.

P. Millán (✉) · L. Orihuela · I. Jurado
Dpto. de Ingeniería, Universidad Loyola Andalucía, Sevilla, Spain
e-mail: pmillan@uloyola.es

L. Orihuela
e-mail: dorihuela@uloyola.es

I. Jurado
e-mail: ijurado@uloyola.es

A. Seuret et al. (eds.), *Delays and Networked Control Systems*,
Advances in Delays and Dynamics 6, DOI 10.1007/978-3-319-32372-5_12

Due to the complexity of these problems and to the fact that WNs are usually large-scale systems, it is not feasible or advisable to control these systems in a centralized manner. Very often, decentralized techniques are not recommendable either since they do not include communication between agents. On the contrary, distributed schemes can provide suitable solutions to be implemented over WNs [26].

The goal of distributed control techniques is to make all the agents in the network seek for the same system-wide objective [22]. A fundamental aspect of this scheme is that the agents have to act according to partial measurements of the state and ignoring the actual control signal that is being applied to the plant. As the performance of the closed-loop system depends on the decisions of all the agents, communication is a very important issue in this approach [19].

One of the major difficulties of distributed solutions over WNs is the fact that transmission channels are not completely reliable due to noises, limited bandwidth, and large number of concurrent transmitters over the same channel. The most common consequences of network congestion are packet dropouts and time delays that can degrade the performance or even destabilize the systems.

On the one hand, many research has been developed to study these effects and propose centralized solutions, see [14, 15, 23, 34, 39, 40] and references therein. On the other hand, there exists a vast literature in the field of distributed control considering ideal networks including MPC-based approaches [2, 5, 8, 18, 25, 32, 33, 37], techniques for large-scale plants [11, 21, 24], and distributed versions of the Kalman filter [1, 16, 20, 27, 28].

In this chapter, the problem of distributed control and estimation is addressed together with network-induced delays and dropouts. This work is an extension of the papers [24, 30], dealing this time with the problem of network-induced delays and dropouts within this distributed paradigm.

The objective is to control a discrete linear time-invariant (LTI) system using a set of agents connected through a communication network. These agents must be able to estimate the state of the system, as well as to control it. However, each one has access only to some outputs of the plant, which makes the interconnection between agents (with its associated delays and dropouts) an essential issue to achieve the system-wide objective. The specific estimation structure implemented in the agents merges a local Luenberger-like observer with consensus strategies. Since the Separation Principle does not hold, it is necessary to design the controllers and the observers in a unique centralized offline step. The stability of the system and estimation errors is ensured by using a Lyapunov–Krasovskii framework. The synthesis problem is posed as a matrix inequality which can be solved using the well known cone complementary algorithm [9].

The chapter is organized as follows. Section 12.2 describes the different elements involved in the problem, namely: plant, agents, and the communication network. The dynamics of the state and estimation errors is studied in Sect. 12.3. Section 12.4 presents the design method based on the Lyapunov–Krasovskii theorem. Section 12.5 illustrates the effectiveness of the approach with some simulations. Finally, Sect. 12.6 outlines the main conclusions and future work.

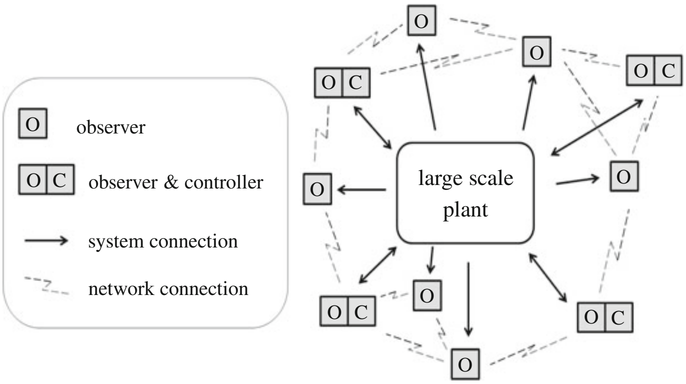

Fig. 12.1 Distributed scheme for the control of a large-scale plant

12.2 System Description and Problem Formulation

This chapter considers an estimation and control scheme composed by a large-scale plant, a communication network, and a set of distributed agents, as depicted in Fig. 12.1. In the following, the different elements composing the distributed system are described in detail.

12.2.1 Plant

We consider a discrete LTI system described in state-space representation. As Fig. 12.1 illustrates, the plant is controlled and/or observed by a set of p agents, each one possibly managing a different control signal. The dynamics of the system can be described as

$$x(k+1) = Ax(k) + \sum_{i=1}^{p} B_i u_i(k), \tag{12.1}$$

where $x \in \mathbb{R}^n$ is the state of the plant and $u_i \in \mathbb{R}^{d_i}$ $(i = 1, \ldots, p)$ is the control signal that agent i applies to the system, and $A \in \mathbb{R}^{n\times n}$ and $B_i \in \mathbb{R}^{n\times d_i}$ are known matrices. For those agents with no direct access to plant inputs, matrices B_i are set to zero.

Defining an augmented control matrix as

$$B \triangleq \begin{bmatrix} B_1 \; B_2 \; \ldots \; B_p \end{bmatrix} \tag{12.2}$$

and an augmented control vector

$$\mathscr{U}(k) \triangleq \left[u_1^T(k)\ u_2^T(k)\ \ldots\ u_p^T(k) \right]^T, \tag{12.3}$$

Equation (12.1) can be compactly rewritten as

$$x(k+1) = Ax(k) + B\mathscr{U}(k), \tag{12.4}$$

where $\mathscr{U}(k) \in \mathbb{R}^d$, with $d = \sum_{i=1}^{p} d_i$. The pair (A, B) in (12.4) is required to be stabilizable.

12.2.2 Network

In the proposed scheme, the agents are linked using a communication network to make possible the information exchange in real time. Each agent is restricted to receive information only from neighboring agents.

The resulting communication topology can be represented using a directed graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, with $\mathscr{V} = 1, 2, \ldots, p$ being the set of nodes (agents) of the graph (network), and $\mathscr{E} \subset \mathscr{V} \times \mathscr{V}$, being the set of links. Assuming that the cardinality of $\mathscr{E}$ is equal to l, and defining $\mathscr{L} = 1, 2, \ldots, l$, it is obvious that a bijective function $g : \mathscr{E} \rightarrow \mathscr{L}$ can be built so that a given link can be either referenced by the pair of nodes it connects $(i, j) \in \mathscr{E}$ or the link index $r \in \mathscr{L}$, so that $r = g(i, j)$. The set of nodes connected to node i is named the neighborhood of i, and denoted as $\mathscr{N}_i \triangleq \{j \in \mathscr{V} | (i, j) \in \mathscr{E}\}$. Directed communications are considered so that link (i, j) implies that node i receives information from node j.

Network links are not assumed to be completely reliable. This way, the packets that the agents exchange may be dropped or delayed. Figure 12.2 illustrates a possible time scheduling in which both effects appear.

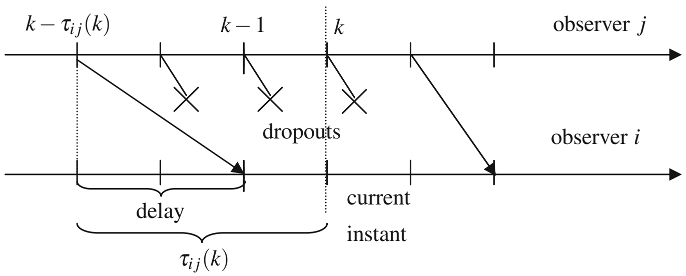

Fig. 12.2 Time scheduling

The input delay approach [12] makes possible to define artificial delays $\tau_{ij} \in \mathbb{N}$ that include the effect of sampling, communication delays, and packet dropouts. Concretely, τ_{ij} is the difference between the current time instant k and the last instant in which a packet from node j was received by node i. It is assumed that the number of consecutive data dropouts is bounded by n_p, and the maximum network-induced delay is bounded by $\bar{d}$. Under these assumptions, the artificial delay can be bounded as $\tau_{ij}(k) \leq \bar{d} + n_p \triangleq \tau_M, \ \forall k$.

Note that each delay τ_{ij} is directly associated to a link, in such a way that the following equivalent notation for the delays can be used:

$$\tau_r(k) = \tau_{ij}(k), \ \ r = 1, \ldots, l, \tag{12.5}$$

where $r = g(i, j)$. That is, we can either refer the delays to a pair of nodes (τ_{ij}) or to a link (τ_r).

12.2.3 Agents

As said before, the large-scale plant (12.1) is collectively monitored and controlled by a network of agents. Each of these agents can be endowed with all or part of the following capabilities:

- sensing plant outputs,
- computing estimations of the plant state,
- applying control actions,
- communicating with neighboring agents.

The approach adopted in this chapter is a distributed scheme in which every agent builds its own estimations of the plant's states based on the information locally collected by the agent (plant outputs) and that received from neighboring agents. Based on these estimations, those agents with access to a control channel compute the control actions to be applied.

Let us define y_i as the plant output measured by agent i:

$$y_i(k) = C_i x(k) \in \mathbb{R}^{r_i}, \tag{12.6}$$

where matrices $C_i \in \mathbb{R}^{r_i \times n}$ are known. If an agent j has no sensing capabilities, then its corresponding matrix C_j is set to zero. Let C denote an augmented output matrix defined as

$$C \triangleq \left[\, C_1^T \; C_2^T \; \ldots C_p^T \,\right]^T.$$

It is assumed that the pair and (A, C) is detectable.

On the other hand, the control counterpart of each agent generates an estimation-based control input to the plant, $u_i(k)$, in the form

$$u_i(k) = K_i\hat{x}_i(k) \in \mathbb{R}^{d_i}, \tag{12.7}$$

where $\hat{x}_i \in \mathbb{R}^n$ denotes the estimation of the plant state computed by agent i, and $K_i \in \mathbb{R}^{d_i\times n}$ $(i \in \mathscr{V})$ are local controllers to be designed. Let K denote the augmented control matrix, defined by

$$K = \left[\, K_1^T \; K_2^T \; \dots \; K_p^T \,\right]^T .$$

Every agent $i \in \mathscr{V}$ implements an estimator of the plant's state based on the following structure:

$$\begin{aligned}\hat{x}_i(k+1) = {} & A\hat{x}_i(k) + B\mathscr{U}_i(k) \\ & + M_i(y_i(k) - C_i\hat{x}_i(k)) \quad \textit{local information} \\ & + \sum_{j\in\mathscr{N}_i} N_{ij}[\hat{x}_j(k-\tau_{ij}(k)) - \hat{x}_i(k-\tau_{ij}(k))], \quad \textit{remote information}\end{aligned} \tag{12.8}$$

where $\mathscr{U}_i(k) = K\hat{x}_i(k) \in \mathbb{R}^d$ is the estimation of the whole control action applied to the plant.

Looking at Eq. (12.8), each agent has two different sources of information to correct its estimates. The first one is the output measured from the plant, $y_i(k)$, which is used similarly to a classical Luenberger observer, $M_i(y_i(k) - \hat{y}_i(k))$, being $M_i,\ i \in \mathscr{V}$, the observers matrices to be designed. The second source of information comes from the estimates received from neighboring nodes, which are also used to correct estimations through the terms $N_{ij}(\hat{x}_j(k-\tau_{ij}(k)) - \hat{x}_i(k-\tau_{ij}(k))),\ \forall j \in \mathscr{N}_i$, where $N_{ij},\ (i, j) \in \mathscr{E}$, are consensus gains to be synthesized. Please notice that the estimations are sent through the communication network, and thus they are affected by delays and dropouts modeled through the extended delays $\tau_{ij}(k)$.

It is worth recalling that the individual agents cannot access to all the control actions being applied to the plant, as each agent implements different control actions based on its particular state estimation (12.7), that is, $B\mathscr{U}_i(k) \neq B\mathscr{U}(k)$.

Ideally, Eq. (12.8) should be implemented using the augmented control vector $\mathscr{U}(k)$ that the network, as a whole, applies to the plant. However, this information is not available to the agents. To circumvent this difficulty and make Eq. (12.8) realizable, the proposed solution consists, roughly speaking, in letting each agent to run its observer with the augmented control vector obtained from its particular estimate. In general, estimated and actual control inputs are different, but if the observers are properly designed and the nodes estimations converge to the plant states, these differences progressively vanish.

12.2.4 Problem Formulation

Once all the elements of the control scheme have been introduced, this section ends with the formal definition of the problem to be solved in this chapter. First, let us define the following sets:

$$\mathscr{M} \triangleq \{M_i, i \in \mathscr{V}\}, \tag{12.9}$$

$$\mathscr{N} \triangleq \{N_{ij}, (i,j) \in \mathscr{E}\}, \tag{12.10}$$

$$\mathscr{K} \triangleq \{K_i, i \in \mathscr{V}\}. \tag{12.11}$$

The goal is to design the set of distributed observers $\mathscr{M}$, consensus matrices $\mathscr{N}$, and controllers $\mathscr{K}$, in such a way that all the estimation errors of each agent $e_i(k) \triangleq x(k) - \hat{x}_i(k)$ and the plant $x(k)$ are stabilized in spite of the delays and packet dropouts affecting the communication.

12.3 Dynamics of the State and Estimation Errors

In order to provide a solution for the previous problem, the dynamics of the state and of the estimation errors are studied in detail.

Proposition 1 *The dynamics of the plant state $x(k)$ is given by*

$$x(k+1) = (A + BK)\, x(k) + \Upsilon(\mathscr{K}) e(k), \tag{12.12}$$

where

$$\Upsilon(\mathscr{K}) = \begin{bmatrix} -B_1K_1 & -B_2K_2 & \cdots & -B_pK_p \end{bmatrix}.$$

The proof is immediate from Eq. (12.4) and the definition of the estimation errors.

The following proposition studies the evolution of the error vector defined as $e(k) \triangleq [e_1^T(k), \ldots, e_p^T(k)]^T \in \mathbb{R}^{np}$.

Proposition 2 *The dynamics of the error vector $e(k)$ is given by*

$$e(k+1) = (\Phi(\mathscr{M}) + \Psi(\mathscr{K}))\, e(k) + \Lambda(\mathscr{N}) d(k), \tag{12.13}$$

where $d(k) \triangleq \left[e^T(k - \tau_1(k)), \ldots, e^T(k - \tau_r(k))\right]^T$ is a vector stacking l delayed versions of the error vector and

$$\Phi(\mathscr{M}) = diag\{(A - M_1C_1), \ldots, (A - M_pC_p)\},$$

$$\Psi(\mathscr{K}) = diag\{BK, \ldots, BK\} + \begin{bmatrix} -B_1K_1 & \ldots & -B_pK_p \\ \vdots & \ddots & \vdots \\ -B_1K_1 & \ldots & -B_pK_p \end{bmatrix},$$

$$\Lambda(\mathscr{N}) = \left[\, \Theta_1(N_{ij}) \; \ldots \; \Theta_r(N_{ij} \; \ldots \; \Theta_l(N_{ij}) \,\right],$$

being $\Theta_r(N_{ij})$ $(r = 1, \ldots, l)$ *a matrix associated with link r and a couple of agents* $(i, j) = g^{-1}(r)$ *with the following structure:*

$$\begin{array}{c} \\ \Theta(N_{ij}) = \end{array} \begin{array}{c} \textit{column} \quad i \qquad\qquad j \qquad\quad \\ \begin{bmatrix} 0 \cdots & 0 & \cdots & 0 & \cdots 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 \cdots & -N_{ij}C_{ij} & \cdots & N_{ij}C_{ij} & \cdots 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 \cdots & 0 & \cdots & 0 & \cdots 0 \end{bmatrix} \end{array} \textit{row } i.$$

Proof The observation error at instant $k + 1$ can be obtained using Eq. (12.8) and Proposition 1:

$$\begin{aligned} e_i(k+1) &= x(k+1) - \hat{x}_i(k+1) \\ &= (A + BK)x(k) + \Upsilon(\mathscr{K})e(k) - A\hat{x}_i(k) \\ &\quad - B\mathscr{U}_i(k) - M_iC_i(x(k) - \hat{x}_i(k)) \\ &\quad - \sum_{j\in\mathscr{N}_i} N_{ij}C_{ij}(\hat{x}_j(k - \tau_{ij}(k)) - \hat{x}_i(k - \tau_{ij}(k))). \end{aligned} \tag{12.14}$$

After some mathematical manipulations, Eq. (12.14) can be rewritten as

$$\begin{aligned} e_i(k+1) &= (A - M_iC_i)e_i(k) + BKe_i(k) + \Upsilon(\mathscr{K})e(k) \\ &\quad - \sum_{j\in\mathscr{N}_i} N_{ij}C_{ij}(e_i(k - \tau_{ij}(k)) - e_j(k - \tau_{ij}(k))). \end{aligned}$$

Finally, since the error vector has been defined as $e^T(k) = \left[\, e_1^T(k) \; \ldots \; e_p^T(k) \,\right]$, it is easy to see that the dynamics of $e(k)$ is (12.13). □

Remark 1 The structure of (2) reveals that, even in the absence of time delays, the Separation Principle does not hold, for matrix $\Psi(\mathscr{K})$ depends on the controllers to be designed. This can be easily justified if we recall that the agents ignore the actual control signal being applied to the plant, and resort to estimations based on the knowledge of the distributed controllers. However, despite this drawback, it will be shown that it is possible to propose an unified design in which all the elements, namely controllers and observers, can be designed to guarantee the overall stability of the system.

12.4 Controller and Observer Design

This sections introduces a result to solve the design problem introduced in Sect. 12.2. The design method resorts to a Lyapunov–Krasovskii approach to prove asymptotic stability of the plant state and the estimation errors when network-induced delays and dropouts exist.

The following theorem proposes a centralized design method through an optimization problem subject to a nonlinear matrix inequality.

Theorem 1 *The problem formulated in Sect. 12.2 can be solved by finding positive definite matrices P_x, P_e, Z_1, Z_2, and sets $\mathscr{M}$, $\mathscr{N}$, $\mathscr{K}$ in (12.9)–(12.11) of observers, consensus matrices, and controllers in such a way that the following matrix inequality is satisfied:*

$$\begin{bmatrix} W & S^T \\ * & -H^{-1} \end{bmatrix} < 0, \tag{12.15}$$

where:

$$W = \begin{bmatrix} -P_x & 0 & 0 & 0 \\ * & -P_e + Z_1 - lZ_2 & \bar{1} \otimes Z_2 & 0 \\ * & * & -2I \otimes Z_2 & \bar{1}^T \otimes Z_2 \\ * & * & * & -Z_1 - lZ_2 \end{bmatrix}, \tag{12.16}$$

$$S = \begin{bmatrix} A + BK & \Upsilon(\mathscr{K}) & 0 & 0 \\ 0 & \Phi(\mathscr{M}) + \Psi(\mathscr{K}) & \Lambda(\mathscr{N}) & 0 \\ 0 & \Phi(\mathscr{M}) + \Psi(\mathscr{K}) - I & \Lambda(\mathscr{N}) & 0 \end{bmatrix}, \tag{12.17}$$

$$H^{-1} = \begin{bmatrix} P_x^{-1} & 0 & 0 \\ * & P_e^{-1} & 0 \\ * & * & \frac{1}{l\tau_M^2} Z_2^{-1} \end{bmatrix}. \tag{12.18}$$

Proof Consider the following quadratic Lyapunov–Krasovskii functional:

$$V(x(k), e(k)) = x^T(k)P_x x(k) + e^T(k)P_e e(k) + \sum_{i=k-\tau_M}^{k-1} e^T(i) Z_1 e(i) \tag{12.19}$$

$$+ l \times \tau_M \sum_{j=-\tau_M+1}^{0} \sum_{i=k+j-1}^{k-1} \Delta e^T(i) Z_2 \Delta e(i), \tag{12.20}$$

where P_x and P_e are positive definite matrices and $\Delta e(k) \triangleq e(k+1) - e(k)$.

The forward difference of the functional (12.19) can be expressed in the following way:

$$\begin{aligned}\Delta V(x(k), e(k)) = {} & x^T(k+1)P_x x(k+1) - x^T(k)P_x x(k) + e^T(k+1)P_e e(k+1) \\ & - e^T(k)P_e e(k) + e^T(k)Z_1 e(k) - e^T(k-\tau_M)Z_1 e(k-\tau_M) \\ & + l \times \tau_M^2 \Delta e^T(k)Z_2\Delta e(k) - l \times \tau_M \sum_{j=k-\tau_M}^{k-1} \Delta e^T(j)Z_2\Delta e(j).\end{aligned}$$

Using Propositions 1 and 2 to substitute the evolution of the plant state and the estimation error in last equation, it yields

$$\begin{aligned}\Delta V(x(k), e(k)) = {} & x^T\left[(A+BK)^T P_x (A+BK)\right]x(k) \\ & + e^T(k)\left[\Upsilon^T(\mathscr{K})P_x\Upsilon(\mathscr{K})\right]e(k) \\ & + 2x^T(k)\left[(A+BK)^T P_x\Upsilon(\mathscr{K})\right]e(k) - x^T(k)P_x x(k) \\ & + e^T(k)\left[(\Phi(\mathscr{M})+\Psi(\mathscr{K}))^T P_e(\Phi(\mathscr{M})+\Psi(\mathscr{K}))\right]e(k) \\ & + d^T(k)\Gamma^T(\mathscr{N})P_e\Gamma(\mathscr{N})d(k) \\ & + 2e^T(k)(\Phi(\mathscr{M})+\Psi(\mathscr{K}))^T P_e\Gamma(\mathscr{N})d(k) - e^T(k)P_e e(k) \\ & + e^T(k)Z_1 e(k) - e^T(k-\tau_M)Z_1 e(k-\tau_M) \\ & + l \times \tau_M^2 \Delta e^T(k)Z_2\Delta e(k) - l \times \tau_M \sum_{j=k-\tau_M}^{k-1} \Delta e^T(j)Z_2\Delta e(j).\end{aligned}$$

Note that the last term is included l times, one for each link. To take into account the delay of each different communication link ($\tau_r(k), \forall r = 1, \ldots, l$), we split it in l terms, each one considering the delay in each specific link:

$$\begin{aligned}-\tau_M \sum_{j=k-\tau_M}^{k-1} \Delta e^T(j)Z_2\Delta e(j) = {} & -\tau_M \sum_{j=k-\tau_M}^{k-\tau_r(k)-1} \Delta e^T(j)Z_2\Delta e(j) \\ & - \tau_M \sum_{j=k-\tau_r(k)}^{k-1} \Delta e^T(j)Z_2\Delta e(j),\end{aligned}$$

The resulting terms can be bounded using the Jensen inequality:

$$-\tau_M \sum_{j=k-\tau_M}^{k-\tau_r(k)-1} \Delta e^T(j)Z_2\Delta e(j) \le -\left[\sum_{j=k-\tau_M}^{k-\tau_r(k)-1} \Delta e(j)\right]^T Z_2 \left[\sum_{j=k-\tau_M}^{k-\tau_r(k)-1} \Delta e(j)\right],$$

$$-\tau_M \sum_{j=k-\tau_r(k)}^{k-1} \Delta e^T(j)Z_2\Delta e(j) \le -\left[\sum_{j=k-\tau_r(k)}^{k-1} \Delta e(j)\right]^T Z_2 \left[\sum_{j=k-\tau_r(k)}^{k-1} \Delta e(j)\right].$$

Consider now the term $l \times \tau_M^2 \Delta e^T(k) Z_2 \Delta e(k)$ in (12.21). Using Proposition 2, this term can be rewritten as

$$\begin{aligned} l \times \tau_M^2 \Delta e^T(k) Z_2 \Delta e(k) = & \; e^T(k)(\Phi^T(\mathscr{M}) + \Psi(\mathscr{K}) - I)^T \left[l \times \tau_M^2 Z_2 \right] \\ & \times \; (\Phi(\mathscr{M}) + \Psi(\mathscr{K}) - I) e(k) \\ & + \; 2e^T(k)(\Phi^T(\mathscr{M}) + \Psi(\mathscr{K}) - I)^T l \times \tau_M^2 Z_2 \Lambda(\mathscr{N}) d(k). \end{aligned}$$

Thus, it is possible to bound the forward difference of the Lyapunov–Krasovkii functional as follows:

$$\begin{aligned} \Delta V(x(k), e(k)) \le & \; x^T \left[(A + BK)^T P_x (A + BK) \right] x(k) \\ & + \; e^T(k) \left[\Upsilon^T(\mathscr{K}) P_x \Upsilon(\mathscr{K}) \right] e(k) \\ & + \; 2x^T(k) \left[(A + BK)^T P_x \Upsilon(\mathscr{K}) \right] e(k) - x^T(k) P_x x(k) \\ & + \; e^T(k) \left[(\Phi(\mathscr{M}) + \Psi(\mathscr{K}))^T P_e (\Phi(\mathscr{M}) + \Psi(\mathscr{K})) \right] e(k) \\ & + \; d^T(k) \Gamma^T(\mathscr{N}) P_e \Gamma(\mathscr{N}) d(k) \\ & + \; 2e^T(k)(\Phi(\mathscr{M}) + \Psi(\mathscr{K}))^T P_e \Gamma(\mathscr{N}) d(k) - e^T(k) P_e e(k) \\ & + \; e^T(k) Z_1 e(k) - e^T(k - \tau_M) Z_1 e(k - \tau_M) \\ & + \; e^T(k)(\Phi^T(\mathscr{M}) + \Psi(\mathscr{K}) - I)^T \left[l \times \tau_M^2 Z_2 \right] \\ & \times \; (\Phi(\mathscr{M}) + \Psi(\mathscr{K}) - I) e(k) \\ & + \; 2e^T(k)(\Phi^T(\mathscr{M}) + \Psi(\mathscr{K}) - I)^T l \times \tau_M^2 Z_2 \Lambda(\mathscr{N}) d(k) \\ & - \sum_{r=1}^{l} \left(\left[\sum_{j=k-\tau_M}^{k-\tau_r(k)-1} \Delta e(j) \right]^T Z_2 \left[\sum_{j=k-\tau_M}^{k-\tau_r(k)-1} \Delta e(j) \right] \right) \\ & - \sum_{r=1}^{l} \left(\left[\sum_{j=k-\tau_r(k)}^{k-1} \Delta e(j) \right]^T Z_2 \left[\sum_{j=k-\tau_r(k)}^{k-1} \Delta e(j) \right] \right). \end{aligned}$$

Defining an augmented state vector as

$$\xi(k) = \left[x^T(k) \quad e^T(k) \quad d^T(k) \quad e^T(k - \tau_M) \right]^T,$$

the bound of $\Delta V(x(k), e(k))$ can be rewritten in a compact way using the matrices W, S, and T defined in Theorem 1:

$$\Delta V(x(k), e(k)) \le \xi^T(k) W \xi(k) - \xi^T(k) S^T H S \xi(k).$$

Therefore, if matrix $\xi^T(k) W \xi(k) - \xi^T(k) S^T H S \xi(k)$ is negative definite, the state of the system and the estimation errors are asymptotically stable. Using Schur complement, this matrix inequality if equivalent to (12.15), and thus the proof is completed. □

The main hindrance of the design method proposed in Theorem 1 is the nonlinearity of the the matrix inequality (12.15) because of the presence of the matrix H^{-1}. Nonetheless, it is possible to adapt the cone complementary algorithm in (see [9]), which let us address the nonlinearities H^{-1} by introducing some new matrix variables and constraints.

First, define a new matrix variable T. Then replace the matrix H^{-1} in (12.15) by the term T and add the additional LMI $H^{-1} \geq T$, which is equivalent to:

$$\begin{bmatrix} H^{-1} & I \\ I & T^{-1} \end{bmatrix} \geq 0.$$

Then introducing variables $\hat{T}, \hat{H}$, the original matrix inequality (12.15) can be substituted by

$$\begin{bmatrix} W & S^T \\ * & T \end{bmatrix} < 0, \; \begin{bmatrix} \hat{H} & I \\ I & \hat{T} \end{bmatrix} \geq 0, \; \hat{T} = T^{-1}, \; \hat{H} = H^{-1}.$$

Using a cone complementarity algorithm, it is possible to obtain feasible solutions for the optimization problem in Theorem 1 by solving the following problem:

$$\text{Minimize } \mathbf{Tr}\left(\hat{H}H + \hat{T}T\right)$$

subject to

$$\begin{cases} \begin{bmatrix} W & S^T \\ * & T \end{bmatrix} < 0, \\ \begin{bmatrix} \hat{H} & I \\ I & \hat{T} \end{bmatrix} \geq 0, \begin{bmatrix} T & I \\ I & \hat{T} \end{bmatrix} \geq 0, \begin{bmatrix} H & I \\ I & \hat{H} \end{bmatrix} \geq 0. \end{cases} \tag{12.21}$$

In order to find a solution for this problem, the iterative algorithm introduced in [9] can be applied. See [23] for further details.

Remark 2 Once the observers and controllers are designed, the implementation is fully distributed, and each agent requires only available local information to operate. Nonetheless, the design method that stems from Theorem 1 needs to be performed offline prior to the implementation, which requires that some information is known a priori, namely: network topology, outputs that every agent can measure, and control channels they has access to.

As regards the computational complexity in the design phase, the relevant figure here is the number of variables to be computed, which is $N_{\#} = n^2(6p^2 + l + \frac{1}{2}) + n\left(6p + \sum_{i=1}^{p}(r_i + d_i) + \frac{1}{2}\right)$. Thus, the number of variables grows rapidly with the number of agents and the number of states of the plant, which makes it hard to solve for large systems.

As an illustrative example, the application example introduced in the next section has 8 states and outputs, 4 agents, 4 control inputs, and a cycle graph. The number of variables is $N_{\#} = 6980$ and the computation time to solve the design problem is approximately 150 min, using Matlab LMI Toolbox on a PC with a 2.5 GHz Intel Core i5 processor and 8GB RAM.

12.5 Application Example

In this section, the proposed design method is tested on a simulated plant consisting of a set of coupled oscillators. First, the plant will be described, giving the necessary considerations with respect to the agents, their observability and control capacities, and communication delays. Finally, a set of different simulations will be shown.

12.5.1 System Description

We consider a set of N inverted pendulums coupled by springs, as Fig. 12.3 shows. The pendulums have all the same characteristics, that is, mass m, length l. The springs are characterized by the same elastic constant k. This mechanical system has been used as a testbed in engineering and control, see [13]. However, what is more interesting of this plant is the fact that it represents the dynamics of a set of coupled oscillators, which has numerous applications in fields as physics, medicine, or communications, see [7, 36]. The objective is to maintain all the pendulums or oscillators in their upright unstable equilibrium points.

In the following, we will consider that the pendulums are being controlled around the upright unstable equilibrium point. Each pendulum is described by two state variables: angular position θ_i and angular velocity $\omega_i \triangleq \dot{\theta}_i$. It is assumed that the control signal is a torque applied to the base of the pendulum. With the hypothesis of small angles, the dynamics of a single pendulum is given by

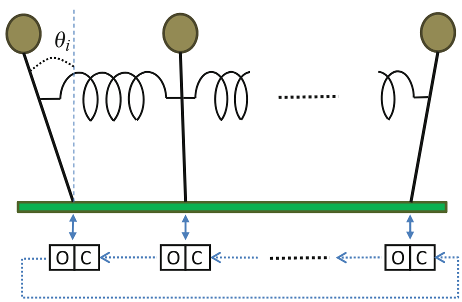

Fig. 12.3 Set of pendulums coupled by springs. Agents and communication graph

$$\begin{bmatrix} \dot{\theta}_i \\ \ddot{\theta}_i \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{g}{l} - \frac{a_i}{ml^2} & 0 \end{bmatrix} \begin{bmatrix} \theta_i \\ \dot{\theta}_i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix} u_i + \sum_j \begin{bmatrix} 0 & 0 \\ \frac{h_{ij}}{ml^2} & 0 \end{bmatrix} \begin{bmatrix} \theta_j \\ \dot{\theta}_j \end{bmatrix}, \tag{12.22}$$

where a_i is the number of springs connected to pendulum i and $h_{ij} = 1$ if pendulum i is connected to pendulum j with a spring, and 0 otherwise. Therefore, the third term represents the influence of the neighborhood in the dynamics of the pendulum i.

The state of the complete system will be the vector stacking all the angular position and velocities of all the agents, that is, $x = [\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, \ldots, \theta_N, \dot{\theta}_N]^T$. Finally, the system dynamics are discretized with sampling period T_s to obtain an equivalent equation to the one in (12.1).

12.5.1.1 Network of Agents

For this system, we will consider a simple network of N agents. Each agent measures the angular position and velocity of a pendulum and applies a torque to its base. The communication graph is a cycle, as Fig. 12.3 shows. In order to stabilize the whole set of pendulums, the agents must apply coordinated control actions. To do so, it becomes essential for an agent to know the rest of the states.[1]

The communication between agents is affected by delays. Not only do these delays come for communication drawbacks (congestion, dropouts, etc.), but also due to the sampling period. The inverted pendulum is, in general, a system with fast dynamics that needs very short sampling periods. It is fairly possible that the agents are not equipped with powerful communication devices to achieve the required rates. Anyway, even in the case they are, the sampling rate could be artificially enlarged pursuing a reduction of the energy consumption.

12.5.2 Simulation Results

For the simulations, we have chosen the set of parameters given in Table 12.1.

In the first experiment, it is shown that the distributed controllers achieve the stabilization of the system for an arbitrary initial condition close to the unstable equilibrium point (Fig. 12.4).

The figure below presents the estimation performance of agent 1. Concretely, it shows the angular velocity of pendulums 2, 3, and 4, together with the estimation of these states from agent 1. As we can see, the agent achieves nice estimations in spite of the communication delays and the distance in the network (Fig. 12.5).

---

[1]The reader may think that, for this particular system, it is only necessary to know the state of the pendulums in the neighborhood. If the agents do not need the estimations of the whole augmented state, we could implement here a sort of reduced-order distributed observer, as the one proposed in [29] for non-delayed systems.

Table 12.1 List of parameters

Parameter	Value	Unit	Description
N	4		Number of pendulums
p	4		Number of agents
m	1	kg	Mass of the pendulum
l	2	m	Length of the pendulum bar
k	5	N/m	Elastic constant of the string
g	9.8	m/s^2	Gravity
T_s	0.05	s	Sampling period
τ_M	2		Maximum delay

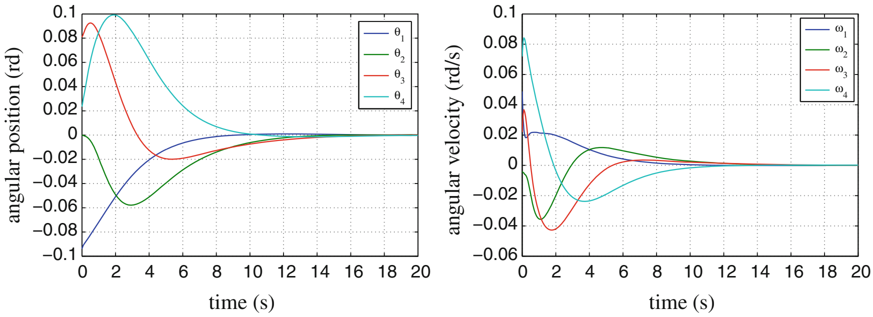

Fig. 12.4 *Left* Evolution of the angular positions. *Right* Evolution of the angular velocities

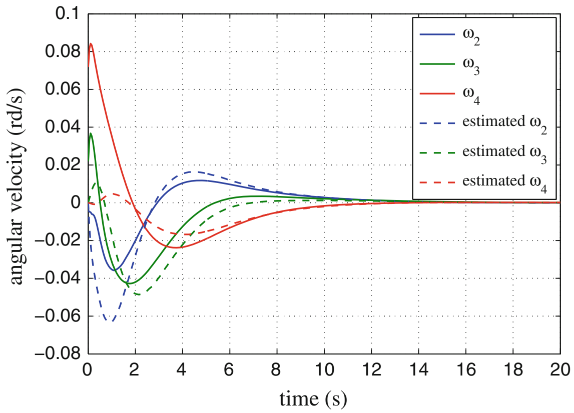

Fig. 12.5 Estimation from agent 1 and actual evolution of the angular velocities of pendulums 2, 3 and 4

The second experiment illustrates the response of the system to external disturbances. Consider that, starting from the equilibrium point, the third pendulum is affected by a disturbance that abruptly changes its position between seconds 5 and 6.

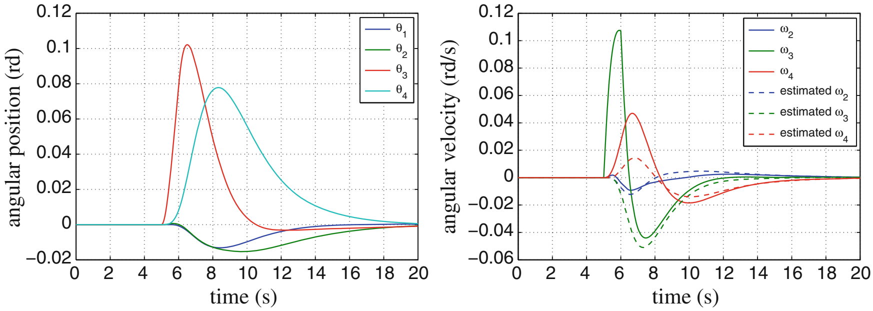

Fig. 12.6 *Left* Evolution of angular positions. *Right* Estimation of angular velocities from agent1

Because of the couplings, both pendulums 2 and 4 are affected as well. Figure 12.6 shows that the response of the controllers and observers are fairly good, despite they have not been designed to reject any disturbances. As expected, the state of pendulum 2 is estimated faster than the others.

12.6 Conclusions

This chapter has studied the problem of stabilizing a large-scale plant with an agent-based distributed paradigm. Unreliable networks affected by time-varying delays and dropouts have been considered. The observers' structure merges a Luenberger-like structure with consensus matrices.

The solution presented ensures the stabilization of both the system state and the observation errors using a Lyapunov–Krasovskii functional. As it has been shown, the design of the controllers and observers must be done in a unique centralized step, which constitutes the weak point of the solution. However, once the controllers and observers are designed, they work in a completely distributed fashion, requiring minimum computation and memory resources. The authors are currently working toward the development of a distributed design method.

Some simulations have been presented to show the performance of the obtained solution.

References

1. P. Alriksson, A. Rantzer, Distributed Kalman filtering using weighted averaging, in *International Symposium on Mathematical Theory of Networks and Systems (MTNS)*, 2006
2. I. Alvarado, D. Limon, D. Muñoz de la Peña, J.M. Maestre, M.A. Ridao, H. Scheu, W. Marquardt, R.R. Negenborn, B. De Schutter, F. Valencia, J. Espinosa, A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark. J. Process Control **21**(5), 800–815 (2011)

3. L. Bencini, G. Collodi, D. Di Palma, A. Manes, G. Manes, Advanced distributed monitoring system for agriculture based on wireless sensor network technology, in *World Congress on Computers in Agriculture and Natural Resources*, 2009
4. L. Briñón Arranz, A. Seuret, C. de Canudas Wit, Translation control of a fleet circular formation of AUVs under finite communication range, in *IEEE Conference on Decision and Control (CDC) and Chinese Control Conference (CCC)*, 2009
5. E. Camponogara, D. Jia, B.H. Krogh, S. Talukdar, Distributed model predictive control. IEEE Control Syst. **22**(1), 44–52 (2002)
6. J. Cortés, S. Martínez, T. Karatas, F. Bullo, Coverage control form mobile sensing networks. IEEE Trans. Robot. Autom. **20**(2), 243–255 (2004)
7. A. Díaz-Guilera, A. Arenas, Phase patterns of coupled oscillators with application to wireless communication. Lectures Notes on Computer Science: Bio-inspired Computing and Communication **5151**, 184–191 (2008)
8. W.B. Dunbar, Distributed receding horizon control of dynamically coupled nonlinear systems. IEEE Trans. Autom. Control **52**(7), 1249–1263 (2007)
9. L. El Ghaoui, F. Oustry, M. AitRami, A cone complementary linearization algorithm for static output-feedback and related problems. IEEE Trans. Autom. Control **42**(8), 1171–1176 (1997)
10. D. Estrin, R. Govindan, J. Heidemann, S. Kumar, Next century challenges: scalable coordination in sensor networks, in *ACM/IEEE International Conference on Mobile Computing and Networking*, 1999
11. M. Farina, G. Ferrari-Trecate, R. Scattolini, Distributed moving horizon estimation for linear constrained systems. IEEE Trans. Autom. Control **55**(11), 2462–2475 (2010)
12. E. Fridman, A. Seuret, J.P. Richard, Robust sampled-data stabilization of linear systems: an input delay approach. Automatica **40**(9), 1441–1446 (2004)
13. M. Guinaldo, D.V Dimarogonas, K.H. Johansson, J. Sánchez, S. Dormido, Distributed event-based control for interconnected linear systems, in *IEEE Conference on Decision and Control (CDC) and European Control Center (ECC)*, 2011
14. X. Jiang, Q.L. Han, S. Liu, A. Xue, A new H_∞ stabilization criterion for networked control systems. IEEE Trans. Autom. Control **53**(4), 1025–1032 (2008)
15. I. Jurado, M.G. Ortega, D.E. Quevedo, F.R. Rubio, An H_∞ suboptimal robust control approach for systems with uncertainties and data dropouts. Int. J. Syst. Sci. **46**(11), 1971–1981 (2014)
16. M. Kamgarpour, C. Tomlin, Convergence properties of a decentralized Kalman filter, in *IEEE Conference on Decision and Control (CDC)*, 2008
17. M. Lin, Y. Wu, I. Wassell, Wireless sensor network: Water distribution monitoring system, in *IEEE Radio and Wireless Symposium (RWS)*, 2008
18. J.P Lynch, K.H. Law, J.A. Blume, Decentralized control techniques for large-scale civil structural systems, in *International Modal Analysis Conference*, 2002
19. J.M. Maestre, D.M. de la Peña, E.F. Camacho, A distributed mpc scheme with low communication requirements, in *American Control Conference (ACC)*, 2009
20. J.M. Maestre, P. Giselsson, A. Rantzer, Distributed receding horizon Kalman filter, in *IEEE Conference on Decision and Control (CDC)*, 2010
21. J.M. Maestre, D. Muñoz de la Peña, E.F. Camacho, Wireless sensor network analysis through a coalitional game: Application to a distributed Kalman filter, in *IEEE International Conference on Networking, Sensing and Control (ICNSC)*, 2011
22. J.M. Maestre, R.R. Negenborn, *Distributed Model Predictive Control Made Easy* (Springer, 2014)
23. P. Millán, L. Orihuela, G. Bejarano, C. Vivas, T. Alamo, F.R. Rubio, Design and application of suboptimal mixed H_2/H_∞ controllers for networked control systems. IEEE Trans. Control Syst. Technol. **20**(4), 1057–1065 (2012)
24. P. Millán, L. Orihuela, C. Vivas, F.R. Rubio, Distributed consensus-based estimation considering network induced delays and dropouts. Automatica **48**(10), 2726–2729 (2012)
25. R.R. Negemborn, De Shutter B., J. Hellendoorn, Multi-agent model predictive control for transportation networks: Serial versus parallel schemes. Eng. Appl. Artif. Intell. **21**(3), 353–366 (2008)

26. R. Olfati-Saber, Distributed Kalman filter with embedded consensus filters, in *IEEE Conference on Decision and Control (CDC) and the European Control Conference (ECC)*, 2005
27. R. Olfati-Saber, Distributed Kalman filtering for sensor networks, in *IEEE Conference on Decision and Control (CDC)*, 2007
28. R. Olfati-Saber, Kalman-consensus filter: Optimality, stability, and performance, in *IEEE Conference on Decision and Control (CDC) and Chinese Control Conference (CCC)*, 2009
29. L. Orihuela, P. Millán, C. Vivas, F.R. Rubio, Reduced-order H_2/H_∞ distributed observer for sensor networks. Int. J. Control **86**(10), 1870–1879 (2013)
30. L. Orihuela, P. Millán, C. Vivas, F.R. Rubio, Suboptimal distributed control and observation: application to a four coupled tanks system. Int. J. Syst. Sci. **47**(8), 1755–1771 (2016)
31. H. Ramamurthy, B.S. Prabhu, R. Gadh, A.M. Madni, Wireless industrial monitoring and control using a smart sensor platform. IEEE Sens. J. **7**(5), 611–617 (2007)
32. S. Roshany-Yamchi, M. Cychowski, R.R. Negenborn, B. De Schutter, K. Delaney, J. Connell, Kalman filter-based distributed predictive control of large-scale multi-rate systems: application to power networks. IEEE Trans. Control Syst. Technol. **21**(1), 27–39 (2012)
33. R. Scattolini, Architectures for distributed and hierarchical model predictive control: a review. J. Process Control **19**(5), 723–731 (2009)
34. A. Seuret, Networked control under synchronization errors, in *American Control Conference (ACC)*, 2008
35. G. Simon, M. Maroti, A. Ledeczi, G. Balogh, B. Kusy, A. Nadas, G. Pap, J. Sallai, K. Frampton, Sensor network-based countersniper system, in *International Conference on Embedded Networked Sensor Systems (SenSys)*, 2004
36. A. Stefanovska, Coupled oscillators: complex but not complicated cardiovascular and brain interactions. IEEE Eng. Med. Biol. Mag. **26**(6), 25–29 (2007)
37. A.N. Venkat, J.B. Rawlings, S.J. Wright, Stability and optimality of distributed model predictive control, in *IEEE Conference on Decision and Control (CDC) and European Control Conference (ECC)*, 2005
38. J. Yick, B. Mukherjee, D. Ghosal, Analysis of a prediction-based mobility adaptive tracking algorithm, in *International Conference on Broadband Networks (BWCCA)*, 2005
39. M. Yu, L. Wang, T. Chu, F. Hao, Stabilization of networked control systems with data packet dropout and transmission delays: continuous-time case. Eur. J. Control **11**(1), 4049 (2005)
40. D. Yue, Q.L. Han, J. Lam, Network-based robust H_∞ control of systems with uncertainty. Automatica **41**(6), 999–1007 (2005)
41. G. Zhao, Wireless sensor networks for industrial process monitoring and control: a survey. Netw. Protocols Algorithms **3**(1), 46–63 (2011)