SEVENTH FRAMEWORK PROGRAMME
THEME – ICT
[Information and Communication Technologies]

SEVENTH FRAMEWORK PROGRAMME

Contract Number:	223854
Project Title:	Hierarchical and Distributed Model Predictive Control of Large-Scale Systems
Project Acronym:	HD-MPC

HD−MPC

Deliverable Number:	D5.1
Deliverable Type:	Report
Contractual Date of Delivery:	September 1, 2010
Actual Date of Delivery:	**August 26, 2010**
Title of Deliverable:	**Report on the state of the art in distributed state and variance estimation, and on preliminary results on disturbance modelling for distributed systems**
Dissemination level:	Public
Workpackage contributing to the Deliverable:	WP5
WP Leader:	Riccardo Scattolini
Partners:	KUL, POLIMI, UNC, UWM
Author(s):	M. Farina, R. Scattolini, J. Garcia, J. Espinosa, J.B. Rawlings

© Copyright by the HD-MPC Consortium

Table of contents

Project co-ordinator

Name:	Bart De Schutter
Address:	Delft Center for Systems and Control Delft University of Technology Mekelweg 2, 2628 Delft, The Netherlands
Phone Number:	+31-15-2785113
Fax Number:	+31-15-2786679
E-mail:	b.deschutter@tudelft.nl
Project web site:	`http://www.ict-hd-mpc.eu`

Executive Summary

This report describes the research activity in the STREP research project **Hierarchical and Distributed Model Predictive Control of Large Scale Systems (HD-MPC)**, focusing on WP5 - "Distributed state estimation algorithms". Specifically, the report aims at presenting the main results available in the literature on the objects of Task 5.1 (State estimation) and Task 5.2 (Variance estimation). The report also presents a concise review of the disturbance modeling approaches used in Model Predictive Control to guarantee tracking properties or to achieve viability in front of unknown inputs and modeling uncertainties. This analysis will be used to extend the main ideas on disturbance modeling also to distributed systems, either to achieve tracking properties, or to design distributed MPC algorithms with guaranteed stability.
The report is organized in six chapters:

- Chapter 1 presents an introduction to the problem of distributed state estimation. A classification of the existing algorithms is proposed in terms of the topology of the communication network, of the amount of information transmitted among the processing units (agents) performing distributed estimation and, most importantly, of the specific problem considered. In particular, a distinction is made between *distributed estimation*, where each agent estimates the state of the whole system, and *partition-based estimation*, where each agent estimates only part of the whole state based on its own measurements and on the information transmitted by its neighborhoods, including the estimates of other system's components. This distinction is not always clear in the technical literature, but it has a major impact on the algorithms to be used in the two different problems.

- Chapter 2 is devoted to present the main algorithms available in the literature for distributed estimation. After an introductory example, the well-known observability property is conjugated with the information available at any node (*local observability*, *regional observability*, i.e. the one based on the information directly collected by the agent and/or provided by its neighborhoods, and *collective observability*). A number of distributed algorithms based on the Kalman filter and consensus algorithms are then described.

- Chapter 3 reviews the partition-based estimation algorithms proposed in the literature. Specifically, two algorithms, coping with overlapping partitions of large-scale systems, are described in detail; they are called *distributed and decentralized Kalman filter* and *consensus-based overlapping decentralized estimator*. While for the former the communication scheme is induced by the presence of overlapping states, for the latter the topology of the network is defined by dependencies among the states of subsystems, resulting in a neighbor-to-neighbor communication scheme.

- The design of state estimators, based either on the Kalman filter or on the moving horizon approach, require the knowledge, or a reliable estimate, of the noise covariance to compute the optimal estimator gain. This is a tight requirement, in particular for what concerns the disturbance acting on the state variables, so that some approaches to variance estimation have been proposed in the literature. In Chapter 4, these techniques are reviewed, focusing attention on two algorithms that appear to be the most reliable and efficient solutions nowadays available to the considered problem.

- Chapter 5 describes the main disturbance modeling assumptions used in Model Predictive Control (MPC) to achieve specific properties for the resulting closed-loop system. In particular, in MPC disturbances are usually included in the problem formulation as constant signals to obtain feedback control laws guaranteeing asymptotic zero error regulation for constant reference signals. Alternatively, the presence of bounded disturbances is considered in the design of robust MPC laws leading to Input-to-State-Stable closed-loop systems. Both the cases are summarized in the chapter and their use in the design of distributed control systems is discussed.
- Finally, in Chapter 6 some conclusions are drawn and some hints for future developments on distributed state and variance estimation are reported.

Chapter 1

Introduction to state estimation for distributed sensing architectures

Recent advances in electronic devices with increasing computational power as well as in wireless communication apparatuses allow for the development of sensor nodes with data processing and transmission capabilities characterized by low costs, low dimensions and low power consumption. As an example, it is possible to mention the huge research efforts at the University of California, Berkeley's, on the so-called smart dust, i.e. tiny wireless micro-electromechanical sensors which can transmit information like air quality-related measurements, temperature, humidity, light or vibrations, see for example [68].

Given a large number of interconnected sensors, i.e. a *sensor network*, a big challenge still widely open is to develop algorithms and protocols allowing the nodes of the network to possess self-organization capabilities and to operate cooperatively, so that each node can carry out local computations and transmit to the other nodes only the partially processed data required to achieve overall sensing objectives.

Therefore, the advantages and challenges of sensor networks are related to the possibility to build large-scale networks, to implement sophisticated communication protocols, to reduce the amount of communication required to perform tasks by distributed and/or local computations and, last but not least, to implement complex power saving modes of operation. Among the many applications of sensor networks, it is possible to recall the following:

- Health: sensor nodes can be deployed to monitor patients.
- Environmental monitoring: prevention of forest fires, forecast pollutant distribution over regions.
- Domotics: Improve quality and energy efficiency of environmental controls (air conditioning, ventilation systems, ...), while allowing reconfiguration and customization, besides saving wiring costs.

As also for the case of distributed control (see the survey paper [54]), to establish a taxonomy could be useful to cast different estimation algorithms, which will be presented in the following, into different classes. Different estimation methods can be classified according to the information exchange between subsystems or nodes and according to the prior information that each sensor has about the process model.

A first classification can be made depending on the topology of the communication network. Two cases can be identified:

- information is transmitted (and received) from any sensor node to all the other nodes (i.e. *all-to-all communication*);
- information is received by a given sensor node i from a given subset of the others $\mathscr{V}_i$, namely the set of i's neighbors (*neighbor-to-neighbor communication*).

The exchange of information (also denoted data delivery) among nodes can be performed according to different protocols, which are presently under investigation (for an extensive review see [47]). In [65], sensor networks are classified as: continuous, event-driven and observer-initiated. In continuous models the nodes transmit information at a given transmission rate (which can be different from the observed process sampling frequency), in event-driven models the sensors transmit information only when a given event occurs, and in observer-initiated data models (also denoted request-reply models) communications occur only in response to an explicit request (e.g. from the neighbors). In this work we focus on continuous data delivery models. Two main classes of continuous data exchange protocols can be identified, on the basis of the number of exchange events among subsystems. Namely:

- information is transmitted (and received) by the sensor nodes only once within each sampling time (*non-iterative algorithms*);
- information can be transmitted (and received) by the local regulators many times (denoted N_T) within the sampling time (*iterative algorithms*).

It is apparent that the amount of information available to the local regulators with iterative algorithms is higher (for example, in the limit case where $N_T \to \infty$, optimality of estimation algorithms can be, in general, guaranteed).

As also discussed in the survey paper [56], two main classes of estimation techniques for distributed sensing schemes are presently under investigation. They are generally both referred, in the literature, to as *distributed state-estimation* algorithms. For the sake of clarity and to avoid confusion, we now propose a new classification, adopted throughout the report, of these two problems.

- The first class of algorithms has the objective to make each node of the sensor network recover the estimate of the whole state vector. In this case, the solution relies on consensus (on measurements or on state estimates) and/or sensor-fusion algorithms. The main drawbacks of such an approach are that each node should know the dynamic model of the overall observed system and that the estimation problem, solved by each sensor node, is a full order problem. This problem will be denoted *distributed estimation*.
- The second approach consists of estimating, for each node, a part of the global state-vector, using information transmitted by other sensors of the network. This problem, which will be denoted *partition-based estimation*, gives rise to low-order estimation problems solved in a decentralized way, and is particularly useful when the observed process is a large scale system.

Although they both aim at solving estimation problems for distributed sensing architectures, the mathematical formulations of the two mentioned issues are deeply dissimilar, and their solutions require different mathematical methods. For these reason, they will be dealt with in different chapters (i.e. Chapter 2 and 3, respectively).

Chapter 2

Distributed state estimation

As discussed in the previous chapter, many theoretical and technological challenges have still to be tackled in order to fully exploit the potentialities of sensor networks. As specified in the Introduction, one of the main issues is that of distributed state estimation, which can be described as follows. Assume that any sensors of the network measures some variables, computes a local estimate of the overall state of the system under monitoring and transmits to its neighbors the measured values, the computed state estimation and the corresponding covariances. Then, the main challenge is to provide a methodology which guarantees that all the sensors asymptotically reach a common reliable estimate of the state variables, i.e. the local estimates reach a *consensus*. This goal must be achieved even if the measurements performed by any sensor are not sufficient to guarantee observability of the process state (i.e. *local observability*), provided that all the sensors, if put together, guarantee such property (i.e. *collective observability*). The transmission of measurements and of estimates among the sensors must lead to the twofold advantage of enhancing the property of observability of the sensors and of reducing the uncertainty of state estimates computed by each node.

Early works [21, 52] proposed distributed Kalman filters based on the parallelization of a centralized Kalman filter which do not rely on consensus algorithms, but require all-to-all communication. Consensus algorithms for distributed state estimation based on Kalman filters have recently been proposed in [15, 4, 44, 41, 60, 42, 25]. In particular, in [44, 41, 60], *consensus on measurements* is used to reduce their uncertainty and Kalman filters are applied by each agent. In [42], three algorithms for distributed filtering are proposed. The first algorithm is similar to the one described in [41], save for the fact that sensors exploit only partial measurements of the state vector. The second approach relies on communicating the state estimates among neighboring agents (*consensus on estimates*). The third algorithm, named *iterative Kalman consensus filter*, is based on the discrete-time version of a continuous-time Kalman filter plus a *consensus step* on the state estimates, which is proved to be stable. However, stability has not been proved for the discrete-time version of the algorithm and optimality of the estimates has not been addressed. Recently, convergence in mean of the local state estimates obtained with the algorithm presented in [41] has been proved in [25], provided that the observed process is stable, and a stability analysis of the state estimator presented in [42] is provided in [43].

In [4] consensus on the estimates is used together with Kalman filters. The weights of the sensors' estimates in the consensus step and the Kalman gain are optimized in order to minimize the estimation error covariance. A two-step procedure is also used in [15], where the considered observed signal is a *random walk*. A two-step algorithm is proposed, where filtering and consensus are performed subsequently, and the estimation error is minimized with respect to both the observer gain and the

consensus weights. This guarantees optimality of the solution.
Some of the methods mentioned above have been reviewed and compared in [56].

More in general, the issue of distributed sensor fusion has been widely studied in the past years, see e.g. [13, 61]. The paper [13] provides an algorithm accounting for dynamically changing interconnections among sensors, unreliable communication links, and faults, where convergence of the estimates to the true values is proved, under suitable hypothesis of "dynamical" graph connectivity, while in [61] the authors propose a minimum variance estimator for distributed tracking of a noisy time-varying signal.

As a simple example of a distributed state estimation problem, taken from [15], assume to have M sensors measuring the same temperature T, whose dynamic evolution is

$$T_{k+1} = T_k + w_k \tag{2.1}$$

where w is a white noise, any sensor i, $i = 1, \dots, M$ provides the measurement

$$y_k^i = T_k + v_k^i \tag{2.2}$$

and the v^i's are white noises with the same variance.
In order to obtain a more reliable estimate of T, a *data fusion* algorithm is required. This can trivially rely on a centralized estimator placed in a *base station* computing the estimate

$$\widehat{T}_{k+1} = \widehat{T}_k + L(\bar{y}_k - \widehat{T}_k) \tag{2.3}$$

where L is a gain to be suitably selected and

$$\bar{y}_k = \frac{1}{M} \sum_{i=1,\dots,M} y_k^i \tag{2.4}$$

Alternatively, the sensors can be arranged in a communication graph configuration. Their measurements are not sent simultaneously and instantaneously to the base station; rather, each sensor computes a local estimation $\widehat{T}_k^i$ based on its available information, i.e. its local measurements, and the information provided by its neighborhoods. In this case, a suitable local estimator can be described by

$$\widehat{T}_{k+1}^i = m(\widehat{T}_k)^i + L(m(y_k)^i - m(\widehat{T}_k)^i) \tag{2.5}$$

where $m(\widehat{T}_k)^i$ and $m(y_k)^i$ represent *mean* (to be specified) values of estimates and measurements, respectively, and are computed on the basis of the information exchanged by sensor i with its neighborhoods. In this case, it is possible to say that the estimate $\widehat{T}^i$ depends on *regional* quantities, i.e. on quantities available to sensor i through its sensing capabilities and through the nodes linked to it. The terms $m(\widehat{T}_k)^i$ and $m(y_k)^i$ are then average values computed from the regional quantities by means of *consensus*. Increasing the number of transmissions N_T among neighboring sensors within one sampling time, it is possible to obtain that

$$m(y_k)^i \to \bar{y}_k \tag{2.6}$$

so that, even if $m(\widehat{T}_k)^i = \widehat{T}_k^i$, the local filters become "optimal", with the performance of the centralized filter. In view of these considerations, it is apparent that consensus is crucial to achieve an agreement among local variables, and it is the milestone of the distributed estimation algorithms proposed so far in the literature. In fact, all the methods available basically rely on *consensus on measurements*, *consensus on estimates* or on both of them.

In the following, some prototype algorithms will be briefly presented to illustrate the main characteristics of the different approaches. First of all, a formal statement of the problem will be given, and the classical (centralized) Kalman filter will be recalled in its *information form*.

2.1 Statement of the problem

In order to state the distributed estimation problem more formally and in general terms, assume that the measured system evolves according to the linear dynamics

$$x_{t+1} = Ax_t + w_t \tag{2.7}$$

where the state x and the disturbance w are constrained as follows: $x \in \mathbb{X} \subseteq \mathbb{R}^n$, $w \in \mathbb{W} \subseteq \mathbb{R}^n$, where $\mathbb{X}$ and $\mathbb{W}$ are closed and compact sets. The initial state x_0 is a random variable with mean μ and covariance Π_0, while the covariance of w is denoted by Q. The system is assumed to be sensed by M nodes, with sensing models

$$y_t^i = C^i x_t + v_t^i,\ i = 1, \cdots, M \tag{2.8}$$

where v^i is a white noise with covariance $R^i \in \mathbb{R}^{p_i \times p_i}$.
The communication network is described by a directed graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ where $\mathscr{V}$ is the set of vertices and $\mathscr{E} \in \mathscr{V} \times \mathscr{V}$ is the set of edges. Moreover, $\mathscr{V}_i^k$ is the set of vertices v_j such that there exists a path of length at most k from v_j to v_i. Finally, associated to the graph $\mathscr{G}$ it is possible to define a matrix K compatible with the graph itself, whose elements (i, j) are such that $k_{ij} \geq 0$ if $(i, j) \in \mathscr{E}$, $k_{ij} = 0$ otherwise, and $\sum_{j=1}^{M} k_{ij} = 1$ for all $i = 1, \ldots M$. Given a graph topology, the freedom allowed in the choice of the elements k_{ij} can be fruitfully exploited to enhance the performance of the adopted consensus algorithms. Matrix K is often used in consensus algorithms to perform averaging on the measurements or on the estimates.
Finally, it is useful to distinguish between *local*, *regional* and *collective* quantities. Specifically, for the node i, the quantity z will be denoted:

- local (indicated with z^i), if related to node i solely;
- regional (indicated with $\bar{z}^i$), if referred to $\mathscr{V}_i^{N_T}$;
- collective (indicated with $\mathbf{z}$), if referred to the whole network.

Accordingly, given the measurements y^i, $\bar{y}^i$ and $\mathbf{y}$, it is possible to trivially define the output transformation matrices C^i (local output transformation, see (2.8)), $\overline{C}^i$ (regional output transformation) and $\mathbf{C}$ (collective output transformation). Then, it will be said that the system is

- locally observable by node i if the pair (A, C^i) is observable;
- regionally observable by node i if the pair $(A, \overline{C}^i)$ is observable;
- collectively observable if the pair $(A, \mathbf{C})$ is observable.

2.2 Information filter

Consider the linear system with its (collective) output described by

$$\begin{aligned} x_{t+1} &= Ax_t + w_t \\ \mathbf{y}_t &= \mathbf{C}x_t + v_t \end{aligned}$$

Let $\hat{x}_{k_1/k_2} = \mathrm{E}[x_{k_1}/\mathbf{y}_1, \ldots, \mathbf{y}_{k_2}]$ be the expected value of x_{k_1} given the outputs up to time k_2, denote by $\Pi_{k_1/k_2} = \mathrm{E}\left[(x_{k_1} - \hat{x}_{k_1/k_2})(x_{k_1} - \hat{x}_{k_1/k_2})^T\right]$ its covariance and define $\mathbf{R} = \mathrm{diag}(R^1, \ldots, R^M)$. Then, the information filter evolves according to the following steps:

Predictor step

$$\begin{aligned}\Pi_{k/k-1} &= A\Pi_{k-1/k-1}A^T + Q \\ \hat{x}_{k/k-1} &= A\hat{x}_{k-1/k-1}\end{aligned}$$

Corrector step

$$\begin{aligned}\Pi_{k/k} &= (\Pi_{k/k-1}^{-1} + F)^{-1} \\ \hat{x}_{k/k} &= \Pi_{k/k}(\Pi_{k/k-1}^{-1}\hat{x}_{k/k-1} + f_k)\end{aligned}$$

where

$$\begin{aligned}F &= \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C} = \sum_{i=1}^{M} C^{iT}(R^i)^{-1}C^i \\ f_k &= \mathbf{C}^T\mathbf{R}^{-1}\mathbf{y}_k = \sum_{i=1}^{M} C^{iT}(R^i)^{-1}y_k^i\end{aligned}$$

Note that the predictor step can be performed *locally* by the M sensors, while the corrector step requires the *collective data* vector $\mathbf{y}$.

2.3 Distributed Kalman filter based on consensus on measurements

Starting from the centralized form of the Information Filter, in [44, 41, 60, 25] a distributed implementation has been proposed, relying on transmission and consensus operations on measurements solely. In particular, letting $\hat{x}^i_{k_1/k_2}$ be the estimate of x_{k_1} carried out by sensor i at instant k_2, and denoting by $\Pi^i_{k_1/k_2} = \mathrm{E}\left[(x_{k_1} - \hat{x}^i_{k_1/k_2})(x_{k_1} - \hat{x}^i_{k_1/k_2})^T\right]$ its covariance, the prediction and correction steps are modified as follows:

Predictor step

$$\begin{aligned}\Pi^i_{k/k-1} &= A\Pi^i_{k-1/k-1}A^T + Q \\ \hat{x}^i_{k/k-1} &= A\hat{x}^i_{k-1/k-1}\end{aligned}$$

Corrector step

$$\begin{aligned}\Pi^i_{k/k} &= ((\Pi^i_{k/k-1})^{-1} + \hat{F}^i)^{-1} \\ \hat{x}^i_{k/k} &= \Pi^i_{k/k}((\Pi^i_{k/k-1})^{-1}\hat{x}^i_{k/k-1} + \hat{f}^i_k)\end{aligned}$$

where $\hat{F}^i$ is the local estimate of $F = \sum_{i=1}^{M} C^{iT}(R^i)^{-1}C^i$ and $\hat{f}^i_k$ is the local estimate of $f_k = \sum_{i=1}^{M} C^{iT}(R^i)^{-1}y^i_k$. These local estimates can be obtained with consensus filters on the basis of regional measurements. However, if regional observability does not hold, the algorithm can not give reliable results; although, for specific topologies of graphs, the distributed estimation algorithm can be considered as a good approximation of the centralized Information Filter for a large number of intercommunications between successive sampling times ($N_T \gg 1$). A stronger results has been reported in [25], where it has been proved that for asymptotically stable systems, the local state estimates produced by the distributed algorithm converge in the mean to the centralized state estimate. A "smarter" way to use the propagation of information through the network has been discussed in [4], where the exploited rationale is to take advantage of the delayed information passing from one node to another even when $N_T = 1$. This can be easily obtained by considering an augmented system where the current state is considered together with its past values over a window of suitable length.

2.4 Distributed Kalman filter based both on consensus on measurements and on consensus on estimates

An evolution of the previous distributed Kalman filters has been described in [42], where transmission of both the local measurements and the local estimates is used to achieve consensus. The algorithm evolves according to the following steps:

Predictor step

$$\begin{aligned}\Pi^i_{k/k-1} &= A\Pi^i_{k-1/k-1}A^T + Q \\ \hat{x}^i_{k/k-1} &= A\hat{x}^i_{k-1/k-1}\end{aligned} \tag{2.9a}$$

Consensus on measurements

$$\begin{aligned}\hat{F}^i &= \sum_{j\in\mathscr{V}^i} C^{jT}(R^j)^{-1}C^j \\ \hat{f}^i_k &= \sum_{j\in\mathscr{V}^i} C^{jT}(R^j)^{-1}y^j_k\end{aligned} \tag{2.9b}$$

Corrector step+consensus on estimates

$$\begin{aligned}\Pi^i_{k/k} &= ((\Pi^i_{k/k-1})^{-1} + \hat{F}^i)^{-1} \\ \hat{x}^i_{k/k} &= \Pi^i_{k/k}((\Pi^i_{k/k-1})^{-1}\hat{x}^i_{k/k-1} + \hat{f}^i_k) + K^i_{cons}\sum_{j\in\mathscr{V}^i}(\hat{x}^j_{k/k-1} - \hat{x}^i_{k/k-1}) \\ &= \hat{x}^i_{k/k-1} + \Pi^i_{k/k}(\hat{f}^i_k - \hat{F}^i\hat{x}^i_{k/k-1}) + K^i_{cons}\sum_{j\in\mathscr{V}^i}(\hat{x}^j_{k/k-1} - \hat{x}^i_{k/k-1})\end{aligned} \tag{2.9c}$$

where K^i_{cons} is the consensus on estimates gain. With respect to the above algorithm, two remarks are in order. First, in [42] it has been proved that it is possible to achieve convergence of the estimates for an analogous algorithm developed in continuous time and under the main assumption of collective observability, but no theoretical results are given for the discrete time implementation previously described. Second, as discussed in [15], the algorithm does not guarantee optimality, since in case of distributed algorithms the optimal gain does not coincide with the Kalman gain.

These issues have been explored in a recent contribution [43], where a formal stability proof is given and performance analysis of the algorithm are provided. Specifically, in [43], the algorithm (2.9) is denoted Kalman Consensus Information Filter if the consensus on measurement step (2.9b) is performed, and it is called Kalman Consensus Filter if consensus on measurement (2.9b) is not performed (i.e. if $\hat{F}^i = C^{iT}(R^i)^{-1}C^i$ and if $\hat{f}^i_k = C^{iT}(R^i)^{-1}y^i_k$). For the Kalman Consensus Filter, for a specific choice of the consensus gain K^i_{cons}, under the assumption that the information matrix $(C^i)^T R^i C^i$ is positive definite for all i and $k \geq 0$, the error dynamics of the Kalman-Consensus filter is globally asymptotically stable. Furthermore, all estimators asymptotically reach a consensus on state estimates, i.e. $\hat{x}^1_k = \cdots = \hat{x}^M_k$ for $k \to \infty$.

2.5 Distributed Kalman filter based on consensus on estimates with optimality properties

As discussed in [4, 5, 6], to guarantee optimality of the distributed Kalman filter, the Kalman gain and the weights on the sensors' estimates (i.e. the elements k_{ij} of the graph matrix K introduced in the previous Chapter) should be the result of an optimization. For instance, the following algorithm is proposed in [4]:

Local information update

$$\hat{x}^{i,local}_{k/k} = \hat{x}^{i,reg}_{k/k-1} + G^i(y^i_k - C^i\hat{x}^{i,reg}_{k/k-1})$$

Regional consensus on the estimates

$$\hat{x}^{i,reg}_{k/k} = \sum_{j\in\mathcal{V}^i} k_{ij}\hat{x}^{i,local}_{k/k}$$

Prediction

$$\hat{x}^{i,reg}_{k+1/k} = A\hat{x}^{i,reg}_{k/k}$$

In this algorithm, the gains G^i and the matrix $K = \{k_{ij}\}$ must be determined to minimize the steady state estimation error covariance matrices. Unfortunately, this minimization problem is not convex, while some bootstrap (iterative 2-step) algorithms have been proposed in [4]. In [5], an evaluation of the performance of such algorithm applied to an ultrasound based positioning application with seven sensor nodes is provided. In [6], the weight selection process has been analyzed yielding performance improvements for some studied examples, and solutions to both optimization problems involved in the iterative off-line weight selection process are given in terms of closed form expressions. The convergence properties of the presented method are still an open problem.

Chapter 3

Partition-based state estimation

Partition-based state-estimation algorithms for large-scale systems decomposed into physically coupled subsystems is of paramount importance in many engineering control problems, such as power networks [57], transport networks [53], process control [66] and robotics [38]. Starting from the idea to decompose a large-scale problems into small-scale ones in order to handle complexity, high computer memory and computational load involved in the solution of centralized state-estimation problems, many studies focused on the design of partition-based filters. The different solutions proposed can be classified according to the model used by each subsystem for state-estimation purposes and to the topology of the communication network among subsystems. Besides the computational benefits of such an approach, in [58] it is highlighted that decomposition can provide insight about the structural properties of dynamical systems, i.e. robustness of stability, optimality, controllability and observability to structural perturbations and to uncertainties (e.g. on the models of the subsystems and on the connections between them). Recently, there has been a revival of interest on these issues, leading to the plug-and-play paradigm [64].

For large scale continuous-time systems, decentralized estimation schemes have been proposed in [58] and [1], where stability conditions for the design of decentralized observers are established using the theory developed in the framework of decentralized control in [57].
The estimation problem for large-scale systems in the discrete-time framework has received more attention, during the years. In [22] a two-level decentralized computational structure is developed, applied to a large-scale system consisting in M dynamically coupled subsystems with uncoupled measurements. The method proposed in that paper provides optimal estimation and is designed in such a way that each subsystem performs low-order computations. The main drawback is that the algorithm requires all-to-all communication and it is iterative.
Later work aimed at reducing the computational complexity of centralized Kalman filtering by parallelizing computations [21, 52]. The algorithm proposed in [21] assumes local processing capabilities for each subsystem, but relies on a central processing unit for global data fusion. On the other hand, the algorithm proposed in [52] does not require a base processing station for global data fusion. Therefore it is denoted in [52] as *fully decentralized*. However, since both [21] and [52] require all-to-all communication and assume each subsystem has full knowledge of the whole dynamics, they should be considered as distributed estimation algorithms rather than partition-based ones.
Starting from the idea of model distribution and of local (nodal) models [10], in [37] the focus is on the use of reduced-order and decoupled models for each subsystem. The proposed solutions, beside neglecting coupling, exploit communication networks that are almost fully connected. Subsystems with overlapping states have been considered in [27, 62, 66, 67]. While in the estimation schemes in

[66, 67] the communication scheme is induced by the presence of overlapping states (which, in principle, can lead to an all-to-all communication scheme), in [27, 62] the topology of the network is defined by dependencies among the states of subsystems resulting in a neighbor-to-neighbor communication scheme.

In the following, some partition-based algorithms will be presented. First, a general statement of the problem (in the linear framework) is given.

3.1 Statement of the problem

Consider the autonomous discrete-time linear system

$$\mathbf{x}_{t+1} = \mathbf{A}\,\mathbf{x}_t + \mathbf{w}_t, \tag{3.1}$$

where $\mathbf{x}_t \in \mathbb{R}^n$ is the state vector, while $\mathbf{w}_t$ represents a disturbance with variance $\mathbf{Q} > 0$. The initial condition $\mathbf{x}_0$ is a random variable with mean $\mathbf{m}_{x_0}$ and covariance matrix $\mathbf{\Pi}_0 > 0$. Measurements on the state vector are performed according to the sensing model

$$\mathbf{y}_t = \mathbf{C}\,\mathbf{x}_t + \mathbf{v}_t \tag{3.2}$$

where $\mathbf{v}_t \in \mathbb{R}^p$ is a white noise with variance $\mathbf{R} > 0$.

Let system (3.1) be partitioned in M low order interconnected submodels, i.e. where a generic submodel has $x_t^{[i]} \in \mathbb{R}^{n_i}$ as state vector. We say that the subsystems states are overlapping if there is at least a component of $\mathbf{x}_t$ which is part of the state vector of more than one subsystem. Otherwise, we say that the subsystems states are non-overlapping. We can denote, for all $i = 1, \ldots, M$, $x_t^{[i]} = T_i \mathbf{x}_t$, where T_i is a linear nodal transformation matrix.
Accordingly, the state transition matrices $A^{[1]} \in \mathbb{R}^{n_1 \times n_1}, \ldots, A^{[M]} \in \mathbb{R}^{n_M \times n_M}$ of the M subsystems are given by $A^{[i]} = T_i \mathbf{A} T_i^{\dagger}$, where $\dagger$ denotes the generalized inverse. In general, T_i is assumed for simplicity to be a scaled orthonormal transformation. Note that, in general, $\sum_{i=1}^{M} n_i \geq n$, where the equality holds only if the states of the subsystems are non-overlapping.
The i-th subsystem obeys to the linear dynamics

$$x_{t+1}^{[i]} = A^{[i]}\,x_t^{[i]} + u_t^{[i],x} + w_t^{[i]}, \tag{3.3}$$

where $x_t^{[i]}$ is the state vector, $u_t^{[i],x}$ collects the effect of state variables of other subsystems, and the term $w_t^{[i]}$ is a disturbance with variance $Q^{[i]}$. The initial condition $x_0^{[i]}$ is a random variable with mean $m_{x_0}^{[i]}$ and covariance matrix $\Pi_0^{[i]}$. Note that $Q^{[i]} > 0$, $R^{[i]} > 0$ and $\Pi_0^{[i]} > 0$ can be obtained from $\mathbf{Q}$, $\mathbf{R}$ and $\mathbf{\Pi}_0$. For example, in [62], $Q^{[i]}$, $R^{[i]}$ and $\Pi_0^{[i]}$ are diagonal blocks of $\mathbf{Q}$, $\mathbf{R}$ and $\mathbf{\Pi}_0$ of appropriate size.
According to (3.2) and to the state partition, the outputs of the subsystems are given by

$$y_t^{[i]} = C^{[i]}\,x_t^{[i]} + u_t^{[i],y} + v_t^{[i]} \tag{3.4}$$

where $u_t^{[i],y}$ collects the effect of the state variables of other subsystems, and the term $v_t^{[i]} \in \mathbb{R}^{p_i}$ represents white noise with variance equal to $R^{[i]}$.
We denote with $\hat{x}_{t_1/t_2}^{[i]}$ the estimate of $x_{t_1}^{[i]}$ performed at time t_2 by subsystem i. Its error covariance matrix is denoted with $\Pi_{t_1/t_2}^{[i]}$

Remark In general, some outputs of system (3.1) can be considered as outputs of more than one subsystem, i.e. $\overline{p} = \sum_{i=1}^{m} p_i \geq p$. Notice, however, that in decentralized control, each local subsystem commonly uses local information, which reduces the amount of transmitted information between subsystems.

The system partition induces an interconnected network of subsystems, which can be described by a directed graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, where the nodes in $\mathscr{V}$ are the subsystems and the edge (j,i) in the set $\mathscr{E} \subseteq \mathscr{V} \times \mathscr{V}$ models that the j-th subsystem influences the dynamics or the output of the i-th subsystem.

3.2 The distributed and decentralized Kalman filter

In [66, 67] and algorithm denoted distributed and decentralized Kalman filter (DDKF) is presented, designed for overlapping partitions. Furthermore, one of the main contributions of [66] is an intriguing discussion on sampling and partitioning of large-scale systems. For instance, the authors point out that the main issues involved in partitioning of the overall system into subsystems are (i) similarity of the subsystems network to the actual plant, (ii) the computational load at each subsystem, (iii) the communication burden and (iv) the available computational resources. They also point out that there is a trade-off between (ii) and (iii) and they propose an heuristic procedure for partitioning.
As far as the DDKF algorithm is concerned, it neglects the dynamic and the output coupling terms (denoted $u_t^{[i],x}$ and $u_t^{[i],y}$ in (3.3) and in (3.4), respectively), and it is composed in two steps:

- **Prediction step:** for all $i = 1, \ldots, M$

$$\hat{x}_{t+1/t}^{[i]} = A^{[i]} \hat{x}_{t/t}^{[i]} \tag{3.5}$$

$$\Pi_{t+1/t}^{[i]} = A^{[i]} \Pi_{t/t}^{[i]} (A^{[i]}) + Q^{[i]} \tag{3.6}$$

- **Estimation step:** for all $i = 1, \ldots, M$

$$\hat{x}_{t/t}^{[i]} = \Pi_{t/t}^{[i]} \left[(\Pi_{t/t-1}^{[i]})^{-1} \hat{x}_{t/t-1}^{[i]} + \sum_{j=1}^{M} (\tilde{\Pi}_{t/t}^{[ij]})^{\dagger} \tilde{x}_{t/t}^{[ij]} \right] \tag{3.7}$$

$$\Pi_{t/t}^{[i]} = \left[(\Pi_{t/t-1}^{[i]})^{-1} + \sum_{j=1}^{M} (\tilde{\Pi}_{t/t}^{[ij]})^{\dagger} \right]^{-1} \tag{3.8}$$

where, for subsystem i, $\tilde{x}_{t/t}^{[ij]}$ and $\tilde{\Pi}_{t/t}^{[ij]}$ are the estimate and the covariance matrix related to the measurement $y_t^{[j]}$, for $j = 1, \ldots, M$, and they are defined as follows

$$\tilde{x}_{t/t}^{[ij]} = T_i T_j^{\dagger} ((C^{[j]})^T (R^{[j]})^{\dagger} C^{[j]})^{\dagger} (C^{[j]})^T (R^{[j]})^{\dagger} y_t^{[j]} \tag{3.9}$$

$$\tilde{\Pi}_{t/t}^{[ij]} = T_i \left[T_j^T ((C^{[j]})^T (R^{[j]})^{\dagger} C^{[j]}) T_j \right]^{\dagger} T_i^T \tag{3.10}$$

Notice that, for $j \neq i$, the transmissions between subsystem j to subsystem i must be performed only if subsystems i and j have overlapping state variables (i.e. $T_i T_j^{\dagger} \neq 0$). Otherwise, if $T_i T_j^{\dagger} = 0$ identically, we have that $\tilde{x}_{t/t}^{[ij]} = 0$ and $\tilde{\Pi}_{t/t}^{[ij]} = 0$ and therefore such terms do not appear in equations (3.7) and (3.8).

3.3 A consensus based overlapping decentralized estimator

In [62, 63] an algorithm is proposed, which combines local Kalman-type estimators with a dynamic consensus strategy. While [63] deals with continuous-time systems, [62] encompasses the case of discrete-time systems.
Assuming that the pairs $(A^{[i]}, C^{[i]})$ are detectable and that the pairs $(A^{[i]}, (Q^{[i]})^{\frac{1}{2}})$ are stabilizable for all $i = 1, \dots, M$, the proposed method is composed by two steps: a local Kalman filter step and a consensus one. While the former involves low-order computations, the consensus step requires that each node computes and stores an estimate of the large-scale-scale system state $\mathbf{x}_t$. Importantly however, each node does not need to have knowledge about the whole system dynamics (3.1).
We define $\hat{\mathbf{x}}^{[i]}_{t_1/t_2}$ as the estimate of the overall system state $\mathbf{x}_{t_1}$ performed by the i-th subsystem at time t_2.

To perform a local Kalman filter, for all $i = 1, \dots, M$ the local estimator is updated according to the equation

$$\hat{x}^{[i]}_{t+1/t} = A^{[i]} \left(\hat{x}^{[i]}_{t/t-1} + \gamma_i(t) L^{[i]} \left[y^{[i]}_t - C^{[i]} \hat{x}^{[i]}_{t/t-1} \right] \right) \tag{3.11}$$

where $L^{[i]}$ is the steady-state Kalman gain given by

$$L^{[i]} = \Pi^{[i]} (C^{[i]})^T \left[C^{[i]} \Pi^{[i]} (C^{[i]})^T + R^{[i]} \right]^{-1}$$

$\Pi^{[i]}$ is a solution of the algebraic Riccati equation

$$\Pi^{[i]} = A^{[i]} \left[\Pi^{[i]} - L^{[i]} C^{[i]} \Pi^{[i]} \right] (A^{[i]})^T + Q^{[i]}$$

and $\gamma_i(t) = 1$ when the i-th agent receives measurements $y^{[i]}_t$, and 0 otherwise (in order to account for missing observations).

The large-scale system state estimator performed by each agent requires the definition of a number of matrices on the basis of the local-system matrices: $\mathbf{A}^{[i]} \in \mathbb{R}^{n \times n}$ has at most $n_i \times n_i$ nonzero elements equivalent to those of $A^{[i]}$, placed at suitable positions, i.e. being $x^{[i]}_t = T_i \mathbf{x}_t$, then $\mathbf{A}^{[i]} = T_i^{\dagger} A^{[i]} T_i$. Matrices $\mathbf{C}^{[i]}$ and $\mathbf{L}^{[i]}$ are $p_i \times n$ and $n \times p_i$ matrices, respectively, obtained from $C^{[i]}$ and $L^{[i]}$ in the same way as $\mathbf{A}^{[i]}$ is obtained from $A^{[i]}$. From (3.11), the consensus-based Kalman-like estimator equations, for $i = 1, \dots, M$, are

$$\hat{\mathbf{x}}^{[i]}_{t/t} = \hat{\mathbf{x}}^{[i]}_{t/t-1} + \gamma_i(t) \mathbf{L}^{[i]} \left[y^{[i]}_t - \mathbf{C}^{[i]} \hat{\mathbf{x}}^{[i]}_{t/t-1} \right] \tag{3.12}$$

$$\hat{\mathbf{x}}^{[i]}_{t+1/t} = \sum_{j=1}^{M} C_{ij}(t) \mathbf{F}^{[j]} \hat{\mathbf{x}}^{[j]}_{t/t} \tag{3.13}$$

where $C_{ij}(t) \in \mathbb{R}^{n \times n}$, $i, j = 1, \dots, M$ are time varying gain matrices such that $C_{ij}(t) = 0$ if no communication between node j and i is allowed (i.e. if $(j,i) \notin \mathscr{E}$). Otherwise, $C_{ij}(t)$ are diagonal matrices with nonnegative entries. The terms $C_{ij}(t)$ are block elements of a consensus matrix $\tilde{C}(t) = \{C_{ij}(t)\}$, which must be row-stochastic, and it is compatible with the communication graph $\mathscr{G}$ by definition of $C_{ij}(t)$, for all $i, j = 1, \dots, M$.
Stability conditions for the collective estimation error dynamics $\boldsymbol{\varepsilon}_t = (\hat{\mathbf{x}}^{[1]}_{t/t} - \mathbf{x}_t, \dots, \hat{\mathbf{x}}^{[M]}_{t/t} - \mathbf{x}_t)$ are provided in [62], where an optimal procedure for choosing the matrices $C_{ij}(t)$ is also proposed.

Chapter 4

Literature survey on variance estimation

Model-based control methods, such as model predictive control (MPC), have become popular choices for solving difficult control problems. Higher performance, however, comes at a cost of greater required knowledge about the process being controlled. Expert knowledge is often required to properly commission and maintain the regulator, to compute target calculation, and to develop state estimators of MPC, for example. This chapter addresses the required knowledge for the project of the state estimator, and describes some techniques with which ordinary closed-loop data may be used to remove some of the information burden from the user. Consider the usual linear, time-invariant, discrete-time model

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k) + w(k) \\ y(k) &= Hx(k) + e(k) \end{aligned}$$

in which $\Phi \in \mathbb{R}^{n\times n}$, $\Gamma \in \mathbb{R}^{n\times m}$, $H \in \mathbb{R}^{p\times n}$, and $w(k)$ and $e(k)$ are uncorrelated zero-mean Gaussian noise sequences with covariances Q and R, respectively. The sequence $u(k)$ is assumed to be a known input. State estimates of the system are considered using a linear, time-invariant state estimator

$$\begin{aligned} \hat{x}(k+1/k) &= \Phi\hat{x}(k/k) + \Gamma u(k) \\ \hat{x}(k/k) &= \hat{x}(k/k-1) + L[y(k) - H\hat{x}(k/k-1)] \end{aligned}$$

in which L is the estimator gain, not necessarily optimal. We denote the residuals of the output equations $(y(k) - H\hat{x}(k/k-1))$ as the L-innovations when calculated using a state estimator with gain L. In order to use the optimal filter, we need to know the covariances of the disturbances, Q,R from which we can calculate the optimal estimator's error covariance and the optimal Kalman filter gain. In most industrial process control applications, however, the covariances of the disturbances entering the process are not known. To address this requirement, estimation of the covariances from open-loop data has long been a subject in the field of adaptive filtering, and can be divided into four general categories: Bayesian [7, 23], maximum likelihood [11, 26], covariance matching [39], and correlation techniques. Bayesian and maximum likelihood methods have fallen out of favor because of their sometimes excessive computation times. They may be well suited to a multi-model approach as in [8]. Covariance matching is the computation of the covariances from the residuals of the state estimation problem. Covariance matching techniques have been shown to give biased estimates of the true covariances. The fourth category is correlation techniques, largely pioneered by Mehra [34, 35] and Carew and Bélanger [9, 14]. In [40] an alternative method to the one presented in [34, 35] is described, where necessary and sufficient conditions for uniqueness of the estimated covariances are

also given. In [40] an exhaustive comparison with the method proposed in [34] is provided, showing cases where the latter is outperformed by the algorithm proposed in [40].
In [3] an algorithm is presented, following the analysis from [40] and generalized in [2] for systems with correlated process and measurement noises. Two contributions can be highlighted: the generalization of the autocovariance least-square method to systems with correlated noise, and the interior-point predictor-corrector algorithm for solving the symmetric semidefinite least-squares problem.
In Section 4.1 the problem of covariance estimation is formally stated, while in Sections 4.2 and 4.3 the algorithms presented in [34] and in [40], respectively, are reviewed in details.

4.1 Statement of the problem

Let us consider the process described by the autonomous system:

$$\begin{cases} x(k+1) &= \Phi x(k) + w(k) \\ y(k) &= H x(k) + e(k) \end{cases} \tag{4.1}$$

where $x, w \in \mathbb{R}^n$, $y, e \in \mathbb{R}^p$, and $\Phi \in \mathbb{R}^{n\times n}$ and $H \in \mathbb{R}^{p\times n}$.
In general the stochastic processes w and e are considered to be independent and uncorrelated white noises with zero mean and covariance matrices:

$$\mathbb{E}\left[w(k)\,w(k)^T\right] = Q^0 \tag{4.2a}$$

$$\mathbb{E}\left[e(k)\,e(k)^T\right] = R^0 \tag{4.2b}$$

We consider these matrices to be not known a priori, while an *a priori* assumption is made on the structure of Q^0 (i.e. position of the zero entries). For instance, we assume that Q^0 has $m \leq n$ unknown parameters. For notation purposes, we denote:

- $\hat{x}(k_1/k_2)$ is the estimate of $x(k_1)$ based on all the measurements up to k_2, i.e. $\{y(1),\ldots,y(k_2)\}$;
- $\varepsilon(k) = x(k) - \hat{x}(k/k-1)$ indicates the 1-step state prediction error;
- $\nu(k) = y(k) - H\hat{x}(k/k-1)$ is the innovation;
- $\hat{Q}$ and $\hat{R}$ the estimates of Q^0 and R^0;
- $\Pi_k = \mathbb{E}\left[\varepsilon(k)\,\varepsilon(k)^T\right]$ the covariance matrix of the prediction error;
- $V_k^i = \mathbb{E}\left[\nu(k)\,\nu(k-i)^T\right]$ the covariance function of the innovation. For simplicity we indicate $V_k^0 = V_k$.

Let us consider the classical structure of the Kalman filter

$$\begin{cases} \hat{x}(k+1/k) &= \Phi\hat{x}(k/k) \\ \hat{x}(k/k) &= \hat{x}(k/k-1) + L_k^o(y(k) - H\hat{x}(k/k-1)) \end{cases} \tag{4.3}$$

where L_k^o is the Kalman gain, whose expression will be computed in the following, in order to minimize the prediction error covariance.

4.1.1 Optimal case

Let's fist consider the case where the noise covariance matrices are exactly known *a priori*, namely where, letting $\hat{R}$ and $\hat{Q}$ the estimates used in the filter design

$$\hat{R} = R^0 \tag{4.4a}$$
$$\hat{Q} = Q^0 \tag{4.4b}$$

We can compute

$$\varepsilon(k+1) = \Phi\,(I - L_k^o H)\,\varepsilon(k) - \Phi L_k^o\, e(k) + w(k) \tag{4.5}$$

Therefore

$$\begin{aligned}\Pi_{k+1} &= \Phi\left(I - L_k^o H\right)\Pi_k\left(I - L_k^o H\right)^T\Phi^T + \Phi L_k^o R^0 (L_k^o)^T \Phi^T + Q^0 \\ &= \Phi\,\Pi_k\,\Phi^T + Q^0 + \Phi L_k^o\,(H\,\Pi_k\,H^T + R^0)\,(L_k^o)^T\Phi^T + \\ &\quad -\Phi L_k^o H\,\Pi_k\,\Phi^T - \Phi\,\Pi_k\,H^T\,(L_k^o)^T\Phi^T\end{aligned} \tag{4.6}$$

Let us define

$$S\,S^T = (H\,\Pi_k\,H^T + R^0) \tag{4.7}$$

Therefore we obtain that

$$\Pi_{k+1} = \Phi\,\Pi_k\,\Phi^T + Q^0 + \Phi(L_k^o S - A)\,(L_k^o S - A)^T\Phi^T - \Phi A A^T \Phi \tag{4.8}$$

where $A = \Phi\,\Pi_k\,H^T\,(S^T)^{-1}$. Minimizing Π_{k+1} with respect to the gain L_k^o we obtain:

$$\min_{L_k^o} \Pi_{k+1} = \Phi\,\Pi_k\,\Phi^T + Q^0 - \Phi\,\Pi_k\,H^T\,(H\,\Pi_k\,H^T + R^0)^{-1}\,H\,\Pi_k\,\Phi^T \tag{4.9}$$

with optimal gain:

$$L_k^o = A S^{-1} = \Pi_k\,H^T\,(H\,\Pi_k\,H^T + R^0)^{-1} \tag{4.10}$$

4.1.2 Suboptimal case

The *suboptimal case* corresponds to the case where uncertain values of the variance matrices R^0 and Q^0 are given, i.e.

$$\begin{aligned}\hat{R} &\neq R^0 \\ \hat{Q} &\neq Q^0\end{aligned}$$

In this case we might introduce two new variables $\tilde{\Pi}_k$ and $\hat{\Pi}_k$. We denote as $\tilde{\Pi}_k$ the real value assumed by the covariance matrix of the estimation error, i.e.

$$\tilde{\Pi}_k = \mathbb{E}\left[\varepsilon(k)\,\varepsilon(k)^T\right] \tag{4.11}$$

while $\hat{\Pi}_k$ represents the estimated value of such matrix, given in (4.6), where, instead of R^0 and Q^0, we use the values $\hat{R}$ and $\hat{Q}$. We have that

$$\tilde{\Pi}_{k+1} = \Phi\,(I - L_k^o H)\,\tilde{\Pi}_k\,(I - L_k^o H)^T\,\Phi^T + \Phi L_k^o R^0\,(L_k^o)^T\Phi^T + Q^0 \tag{4.12a}$$
$$\hat{\Pi}_{k+1} = \Phi\,(I - L_k^o H)\,\hat{\Pi}_k\,(I - L_k^o H)^T\,\Phi^T + \Phi L_k^o \hat{R}\,(L_k^o)^T\Phi^T + \hat{Q} \tag{4.12b}$$

We can easily see from the previous equations that, while L_k^o in (4.10) minimizes $\hat{\Pi}_k$, the real covariance of the estimation error $\tilde{\Pi}_k$ is not minimized by formula (4.10). Furthermore it is also possible to infer that $v(k)$ is a white noise process only in the optimal case (see the next Section 4.2).

4.2 Mehra's algorithm

In [34] a method is proposed, for the unbiased estimation of Q^0 and R^0, based on the analysis of $v(k)$ given by the application of a suboptimal Kalman filter. The main steps of the algorithm are the following

(i) application of (4.10) (steady state formulation);

(ii) test of optimality of the Kalman filter applied at step (i);

(iii) in case the optimality test fails, new estimates of matrices Q^0 and R^0 are computed.

The optimality test at step (ii) consists in verifying that the innovation sequence $v(t)$ is white. For details see [34]. In the next section a sketch of the algorithm proposed for the estimation of Q^0 and R^0 is provided. The author relies on two main assumptions: (I) the pair (Φ, H) is observable and (II) the transition matrix Φ is non-singular.
From now on the steady-state formulation of the Kalman filter is applied. Therefore, V^i, Π and L^o denote the asymptotic values of V_k^i, Π_k and L_k^o, respectively. Similarly, $\hat{\Pi}$ and $\tilde{\Pi}$ denote the asymptotic values of $\hat{\Pi}_k$ and $\tilde{\Pi}_k$, respectively. Specifically, $\hat{\Pi}$ is the result of the algebraic Riccati equation

$$\hat{\Pi} = \Phi(I - L^o H)\hat{\Pi}(I - L^o H)^T \Phi^T + \Phi L^o \hat{R}(L^o)^T \Phi^T + \hat{Q} \tag{4.13}$$

where

$$L^o = \hat{\Pi} H^T (H \hat{\Pi} H^T + \hat{R})^{-1}$$

while $\tilde{\Pi}$ is the result of the equation

$$\tilde{\Pi} = \Phi(I - L^o H)\tilde{\Pi}(I - L^o H)^T \Phi^T + \Phi L^o R^0 (L^o)^T \Phi^T + Q^0 \tag{4.14}$$

Schematically, the proposed method for variance estimation can be sketched as follows.

1) We will denote as $\bar{V}^i$ the sampled estimate of V^i, which can be computed, for example, according to the following equation

$$\bar{V}^i = \frac{1}{N_{data}} \sum_{k=1}^{N_{data}-i} v(k)v(k+i)^T$$

where N_{data} denotes the available number of data samples.

2) It can be shown that, in steady state, $v(k)$ has variance equal to

$$V = V^0 = H \tilde{\Pi} H^T + R^0 \tag{4.15}$$

and covariance V^i, for $i \geq 1$:

$$V^i = H \left[\Phi \left(I - L^o H\right)\right]^{i-1} \Phi \left[\tilde{\Pi} H^T - L^o V^0\right] \tag{4.16}$$

Rewriting (4.16), and replacing V^i with its sampling counterpart $\bar{V}^i$, for all $i = 0, \ldots, k$, we obtain that

$$\begin{aligned}
\bar{V}^1 &= H\Phi\tilde{\Pi}H^T - H\Phi L^o \bar{V}^0 \\
\bar{V}^2 &= H\Phi^2\tilde{\Pi}H^T - H\Phi L^o \bar{V}^1 - H\Phi^2 L^o \bar{V}^0 \\
&\vdots \\
\bar{V}^k &= H\Phi^k\tilde{\Pi}H^T - H\Phi L^o \bar{V}^{k-1} - \cdots - H\Phi^k L^o \bar{V}^0
\end{aligned}$$

Therefore, one obtains that an estimate $\hat{\tilde{\Pi}}\hat{H}^T$ of $\tilde{\Pi}H^T$ can be computed as

$$\hat{\tilde{\Pi}}\hat{H}^T = B^\dagger \begin{bmatrix} \bar{V}^1 + H\Phi L^o \bar{V}^0 \\ \bar{V}^2 + H\Phi L^o \bar{V}^1 + H\Phi^2 L^o \bar{V}^0 \\ \vdots \\ \bar{V}^k + H\Phi L^o \bar{V}^{k-1} + \cdots + H\Phi^k L^o \bar{V}^0 \end{bmatrix} \tag{4.17}$$

where $B^\dagger$ denotes the pseudo-inverse of B, which is defined as

$$B = \begin{bmatrix} H \\ H\Phi \\ \vdots \\ H\Phi^{k-1} \end{bmatrix} \Phi$$

Under the observability assumption, and if the transition matrix Φ is non-singular, if k is greater than the observability index of the pair (Φ, H), then the pseudo-inverse of B can be computed.

3) Compute a new estimate $\hat{\hat{R}}$ of R^0 according to the equation (4.15), i.e. $\hat{\hat{R}} = \bar{V}^0 - H(\hat{\tilde{\Pi}}\hat{H}^T)$

4) Compute an estimate of Q^0 using the equation (4.14). One can write, from (4.14), that

$$\tilde{\Pi} = Q^0 + \Phi\tilde{\Pi}\Phi^T + \Omega \tag{4.18}$$

where

$$\Omega = \Phi[-L^o H\tilde{\Pi} - \tilde{\Pi}H^T(L^o)^T + L^o\bar{V}^0(L^o)^T]\Phi^T \tag{4.19}$$

an estimate of which can be computed based on the previous steps. Iterating (4.18) one obtains that, for all $k > 1$

$$\tilde{\Pi} = \Phi^k\tilde{\Pi}(\Phi^k)^T + \sum_{j=0}^{k-1}\Phi^j\Omega(\Phi^j)^T + \sum_{j=0}^{k-1}\Phi^j Q^0(\Phi^j)^T \tag{4.20}$$

Pre-multiplying both sides of equation (4.20) by H and post-multiplying it by $(\Phi^{-k})^T H^T$ one obtains that, for $k \geq 1$

$$\sum_{j=0}^{k-1} H\Phi^j Q^0(\Phi^{j-k})^T H^T = H\tilde{\Pi}(\Phi^{-k})^T H^T - H\Phi^k\tilde{\Pi}H^T - \sum_{j=0}^{k-1} H\Phi^j\Omega(\Phi^{j-k})^T H^T \tag{4.21}$$

Note that an estimate of the right hand side of equation (4.21) is given, since $\bar{V}^0$ and $\hat{\tilde{\Pi}}\hat{H}^T$ have been previously computed. After choosing a linearly independent set of equations (4.21) (for details see [34]), one can identify the $m \leq n$ unknown entries of Q^0.

4.3 Variance estimation with [40]

Consider a linear, time-invariant, discrete-time autonomous model (4.1). State estimates of the system are considered using a linear, time-invariant state estimator with observer gain L

$$\begin{aligned} \hat{x}(k+1|k) &= \Phi\hat{x}(k|k) \\ \hat{x}(k|k) &= \hat{x}(k|k-1) + L[y(k) - H\hat{x}(k|k-1)] \end{aligned} \tag{4.22}$$

From (4.1) and (4.22), the prediction error evolves according to the system

$$\varepsilon(k+1) = \underbrace{\Phi(I-LH)}_{\bar{A}}\varepsilon(k) + \underbrace{\begin{bmatrix} I & -\Phi L \end{bmatrix}}_{\bar{G}}\underbrace{\begin{bmatrix} w(k) \\ e(k) \end{bmatrix}}_{\bar{w}(k)} \tag{4.23}$$

Then, the state-space model of the L-innovations is defined as:

$$\begin{aligned} \varepsilon(k+1) &= \bar{A}\varepsilon(k) + \bar{G}\bar{w}(k) \\ v(k) &= H\varepsilon(k) + e(k) \end{aligned} \tag{4.24}$$

where the L-innovation is $v(k) = y(k) - H\hat{x}(k|k-1)$.
From now on, we require that the system is detectable and that the chosen observer gain L makes equation (4.24) asymptotically stable. Namely, the basic assumptions are the following.
Assumption 1 The pair (Φ, H) is detectable.
Assumption 2 The matrix $\bar{A}$ is Schur.

In this formulation, the state and sensor noises are correlated. In fact

$$\begin{aligned} \mathbb{E}[\bar{w}(k)\bar{w}(k)^T] &= \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \equiv \bar{Q}_w \\ \mathbb{E}[\bar{w}(k)\bar{v}(k)^T] &= \begin{bmatrix} 0 \\ R \end{bmatrix} \end{aligned}$$

Furthermore, we assume that the initial estimation error is distributed with mean m_0 and covariance Π_0, i.e. $\mathbb{E}[\varepsilon(0)] = m_0$ and $\text{var}(\varepsilon_0) = \Pi_0$. Propagating the estimation error we have, provided that $\mathbb{E}[w(k)] = \mathbb{E}[e(k)] = 0$, $\mathbb{E}[\varepsilon_k] \to 0$ as $k \to \infty$ and that $\Pi_k = \text{var}(\varepsilon_k)$ evolves according to the equation

$$\Pi_{k+1} = \bar{A}\Pi_k\bar{A}^T + \bar{G}\bar{Q}_w\bar{G}^T \to \Pi$$

The algorithm proposed in [40] assumes that the time index k is sufficiently large, such that the effect of the initial conditions can be neglected (steady state assumption). Therefore, the error covariance matrix Π obeys to

$$\Pi = \bar{A}\Pi\bar{A}^T + \bar{G}\bar{Q}_w\bar{G}^T \tag{4.25}$$

Recall that the autocovariance V^i is defined as

$$V^i = \mathbb{E}[v(k)v(k+i)^T]$$

Moreover, we define the autocovariance matrix (referred to ACM in [40]) as

$$\mathscr{R}(N) = \begin{bmatrix} V^0 & \cdots & V^{N-1} \\ \vdots & \ddots & \vdots \\ (V^{N-1})^T & \cdots & V^0 \end{bmatrix} \tag{4.26}$$

where the number of lags N is a user-defined parameter. The ACM of the L-innovations can be written as:

$$\mathscr{R}(N) = \mathscr{O}\Pi\mathscr{O}^T + \mathscr{C}\left[\bigoplus_{i=1}^{N}\bar{G}\bar{Q}_w\bar{G}^T\right]\mathscr{C}^T + \Psi\left[\bigoplus_{j=1}^{N}R\right] + \left[\bigoplus_{j=1}^{N}R\right]\Psi^T + \left[\bigoplus_{j=1}^{N}R\right] \tag{4.27}$$

where

$$\mathscr{O} = \begin{bmatrix} H \\ H\bar{A} \\ \vdots \\ H\bar{A}^{N-1} \end{bmatrix}, \quad \mathscr{C} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ H & 0 & 0 & 0 \\ \vdots & \ddots & & \vdots \\ H\bar{A}^{N-2} & \cdots & H & 0 \end{bmatrix}, \quad \Psi = \Gamma \left[\bigoplus_{j=1}^{N} -\Phi L \right]$$

and the definitions of Kronecker sum and direct sum are used (see [12] and [55]).

Recall that a sampled estimator $\bar{\mathscr{R}}(N)$ of matrix $\mathscr{R}(N)$ can be computed from innovation data. In fact, a sampled (unbiased) estimate of V^i is

$$\bar{V}^i = \frac{1}{N_{data} - i} \sum_{k=1}^{N_{data}-i} v(k)k(k+i)$$

The algorithm proposed in [40] aims to "solve" equation (4.27) as a least square problem, with respect to the unknown parameters Q and R (note that the variable $\bar{\Pi}$ is expressed as a function of Q_w, in view of the algebraic equation (4.25)), where $\bar{\mathscr{R}}(N)$ is used as an estimator of $\mathscr{R}(N)$. To cast equation (4.27) as a linear regression, complex matrix transformations are involved (for details see [40]). Finally, the following theorem is stated.

Theorem
Provided that the mentioned least squares problem has a unique solution, the noise covariance estimates $\hat{Q}_w$ and $\hat{R}_n$ are unbiased for all sample sizes and converge asymptotically to the true covariances as the number of data available N_{data} converges to infinity.

Chapter 5

Preliminary results on disturbance modeling for distributed systems

In this Chapter, the use of disturbance models in MPC is briefly reviewed and some introductory considerations on the problem of disturbance modeling in distributed control systems are reported.
In the MPC literature, disturbances have been considered mainly from the following two different points of view:

1. disturbances modeled as (piecewise) constant signals have been included in the problem formulations to compute feedback control laws guaranteeing some specific properties, such as asymptotic zero error regulation for constant reference signals, see e.g. [36], [46] and the references quoted there;

2. disturbances have been modeled as unknown, but bounded, signals acting on the system state and to be rejected by the MPC control law to guarantee some fundamental properties, such as Input-to-State Stability (ISS) or Practical ISS (p-ISS). In this case, this problem has lead to the development of robust MPC algorithms both in closed-loop and in open-loop form, see e.g. [32], [50].

These two streams of research will be very concisely summarized in the following and will be related to the problem of designing distributed MPC laws with stability and tracking properties.

5.1 Disturbance modeling in Model Predictive Control for offset-free tracking

In MPC, many approaches have been proposed to guarantee an offset free response for constant reference signals. The most polite and effective solution is to resort to the classical approach based on the so-called Internal Model Principle (IMC), see [20], which consists of augmenting the process model with a set of integrators fed by the tracking errors. A stabilizing regulator is then synthesized for the augmented system. Finally, the overall regulator is composed by the ensemble of the stabilizing one and of the integrators, i.e. of the internal model of the reference signal. The solution based on the IMC is very effective also to guarantee asymptotic zero error regulation for references generated by unstable exosystems and can be used for MPC of nonlinear process models ([30]).
A criticism to IMC-based solutions applied to MPC, [36], lies in the need to enlarge the dimension of the system state with the internal model of the exosystem. In the case of large scale systems, this can

significantly increase the computational burden associated to the on-line solution of the dynamic optimization problem. For this reason, MPC solutions are often based on the computation of the control variations with respect to the steady state value required to force the controlled outputs to the desired constant reference value. In more formal terms, consider a system under control described by the discrete-time linear model

$$\begin{aligned} \widetilde{x}(k+1) &= A\widetilde{x}(k)+B\widetilde{u}(k) \\ \widetilde{y}(k) &= C\widetilde{x}(k) \end{aligned} \tag{5.1}$$

a constant reference y^o, and an equilibrium pair $(\widetilde{x}_s, \widetilde{u}_s)$ such that

$$\begin{aligned} \widetilde{x}_s &= A\widetilde{x}_s+B\widetilde{u}_s \\ y^o &= C\widetilde{x}_s \end{aligned} \tag{5.2}$$

Let $x(k) = \widetilde{x}(k) - \widetilde{x}_s$, $u(k) = \widetilde{u}(k) - \widetilde{u}_s$, $y(k) = \widetilde{y}(k) - y^o$, and denote the model (5.1) centered at $(\widetilde{x}_s, \widetilde{u}_s)$ by

$$\begin{aligned} x(k+1) &= Ax(k)+Bu(k) \\ y(k) &= Cx(k) \end{aligned} \tag{5.3}$$

With reference to system (5.3), the MPC action is computed by minimizing at any time instant k and with respect to present and future controls $u(k), u(k+1), \ldots$ a cost function penalizing the future output and control variables over an (ideally infinite) future horizon, so that an optimization problem of the form (or similar ones):

$$\min J = \sum_{i=0}^{\infty} y'(k+i)Qy(k+i)+u'(k+i)Ru(k+i) \tag{5.4}$$

is solved, possibly subject to constraints on the future control moves and state trajectories. Finally, according to a receding horizon strategy, the overall control action is given by $\widetilde{u}(k) = u(k) + \widetilde{u}_s$

It is apparent however that the above procedure does not guarantee zero steady-state error if the original system (5.1) is subject to disturbances which are neglected in the computation of the equilibrium $(\widetilde{x}_s, \widetilde{u}_s)$ through (5.2) or in case of modeling errors. Therefore, it is a common practice (see e.g. [36, 46] and the references quoted there) to assume that the state and/or the output of model (5.3) is subject to an additional disturbance $d(k)$ with a given dynamics. The state of the corresponding augmented system is then estimated on-line with a Kalman-type filter and the estimated value of the disturbance is used to re-compute the steady-state pair $(\widetilde{x}_s, \widetilde{u}_s)$. The typical choice is to consider for the disturbance an integrating dynamics, so that model (5.3) subject to disturbances takes the form

$$\begin{aligned} x(k+1) &= Ax(k)+Bu(k)+\Gamma d(k)+w_x(k) \\ d(k+1) &= d(k)+w_d(k) \\ y(k) &= Cx(k)+Md(k)+v(k) \end{aligned} \tag{5.5}$$

where the matrices Γ, M are design parameters and w_x, w_d and v are zero-mean white noises. Once the estimate $\begin{bmatrix} \widehat{x}'(k) & \widehat{d}'(k) \end{bmatrix}'$ has been computed with a suitable observer, assuming that the future disturbance value equals the current one, i.e. $\widehat{d}(k+i) = \widehat{d}(k) = \widehat{d}$, $i > 0$, the new equilibrium point $(\widetilde{x}_s, \widetilde{u}_s)$ is recomputed (whenever possible) as the solution of the following set of linear equations:

$$\begin{bmatrix} I-A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} \widetilde{x}_s \\ \widetilde{u}_s \end{bmatrix} = \begin{bmatrix} \Gamma & 0 \\ M & I \end{bmatrix} \begin{bmatrix} \widehat{d} \\ y^o \end{bmatrix} \tag{5.6}$$

The conceptual approach above described must be deeply exploited to clarify and solve a number of fundamental problems for its effective implementation. Among then, the following are of paramount importance and have been extensively studied in the already mentioned papers [36], [46]:

- the augmented system (5.5) must be observable, or detectable, to allow for the estimation of the enlarged state;
- the set of linear equations (5.6) must be solvable. This is equivalent to ask for conditions on the number of controlled and manipulated outputs and/or on the absence of plant invariant zeros in $z = 1$;
- the steady-state pair $(\tilde{x}_s, \tilde{u}_s)$ must satisfy the control and state constraints of the problem. If this does not hold, feasible solutions must be computed by solving suitable least-squares problems.

A critical analysis of these points is beyond the scopes of this report, and the interested reader is referred to [36], [46], [45]. However, a couple of remarks are worth recalling. First, additional dynamics can be assumed to generate the disturbance, which turns out to be given by the output of a stable system fed by the integrators, see again [46] or [29]. This can be useful to center the estimation and the control design in a prescribed frequency band specified by the additional dynamics. Second, as suggested by [48], system (5.5) (with null disturbances w_x, w_d, v), can be given the velocity form

$$\begin{array}{lcl} \Delta x(k+1) & = & A\Delta x(k) + B\Delta u(k) \\ z(k+1) & = & z(k) + CA\Delta x(k) + CB\Delta u(k) \\ y(k) & = & z(k) + v(k) \end{array} \tag{5.7}$$

where $\Delta x(k) = x(k) - x(k-1)$ and $\Delta u(k) = u(k) - u(k-1)$. An MPC algorithm can then be used for system (5.7) to compute the future control increments Δu. This implicitly corresponds to plug an integral action on the input variables, so that an IMC-type solution is obtained. This approach traces back to early predictive control algorithms, such as Generalized Minimum Variance (GMV), see [16], or Generalized Predictive Control (GPC), see [17], where CARIMA (Controlled AutoRegressive Integrating Moving Average models) were used.

In the design of MPC algorithms for distributed systems, the disturbance modeling approach described in this section can be easily extended to cope with the same objectives, such as the tracking of constant references. Specifically, given a large scale system made by a number of interacting subsystems, according to the previously described guidelines it is possible to design for any subsystem a local MPC regulator guaranteeing tracking properties provided that the overall stability is achieved. In this sense, the described methodologies for disturbance modeling do not appear to be crucial to solve the fundamental questions related to the design of distributed control laws, such as the amount of information transmitted among the subsystems and its use in the design of the control laws, the required strength of the interconnections, the achievable stability properties.

5.2 Disturbance rejection and ISS stability in distributed control with MPC

In recent years, research in MPC has been focused in the development of algorithms guaranteeing some fundamental stability properties also in presence of model uncertainties, parameter variations, external disturbances, see e.g. [28, 31, 51, 32, 33]. In this framework, the system under control is usually assumed to be described by

$$x(k+1) = f(x(k), u(k), w(k)),\ k \geq t,\ x(t) = \hat{x} \tag{5.8}$$

where the disturbance $w \in R^q$ can model a wide number of the uncertainties above described; as such it can be assumed to be either a state and control dependent term, i.e.

$$w(k) = f_w(x(k), u(k)) \tag{5.9}$$

where $f_w(\cdot, \cdot)$ is a suitable function, or an external bounded signal, i.e.

$$w \in W \tag{5.10}$$

where $W \subseteq R^q$ is a known compact set containing the origin.

The presence of the disturbance w strongly impacts on feasibility. In fact, even though at a generic time instant k the optimization problem is feasible, the effect of the disturbance could bring the state outside the feasibility region in the next time instants. To deal with this problem, in the design of robust Nonlinear Model Predictive Control (NMPC) algorithms a common approach consists of considering the a-priori worst possible effect of the future disturbances and guaranteeing that, event in this case, feasibility is not lost. In so doing, a *min-max* optimization problem must be stated and solved, where the maximization is associated to the effect of the disturbance over the considered prediction horizon, while the minimization of the selected cost function must be performed with respect to the future control actions given by the control law

$$u(k+i) = u_{op}(k+i) + \kappa_i(x(k+i)),\ i = 0, \ldots, N-1 \tag{5.11}$$

where $u_{op}(k+i), i = 1, \ldots, N$ are open-loop terms to be computed through the optimization problem, while the functions $\kappa_i(x)$ are closed-loop terms which can be either time-invariant and selected a-priori or whose parameters can be optimized on-line. The algorithms proposed in the literature for linear and nonlinear systems can be roughly classified as follows.

- Methods where the maximization problem is solved off-line and the design parameters are suitably modified to guarantee feasibility. In these approaches, the basic idea is to include in the problem formulation some additional constraints on the states $\widetilde{x}(k+i), i = 1, \ldots, N$ predicted over the considered horizon. Specifically, the predicted state $\widetilde{x}(k+i)$ is forced to belong to a set $\widetilde{X}(k+i) \subset X$ chosen so that, for any feasible disturbance sequence $w(k+j), j = 0, \ldots, i-1$, the real state still belongs to X. The sequence of sets $\widetilde{X}(k+i), i = 1, \ldots, N$ forms a "tube" where the predicted state is forced to remain. In addition, the auxiliary control law is usually chosen so that the terminal set X_f is a robustly positive invariant set for the corresponding closed-loop system. Once the "tube" has been computed and the additional constraints on the future states have been included into the problem formulation, minimization of the selected cost function can be performed on-line by resorting to the Receding Horizon principle. In the context of MPC for nonlinear systems, [28] describes an algorithm where only the open-loop terms $u_{op}(k+i)$ (5.11) are considered, while fixed closed-loop functions $\kappa_i(x(k+i))$ are used in [51].

- Methods where the whole min-max problem is solved on-line. In this case, the control law (5.11) is usually made only by the parameterized state-feedback control laws $\kappa_i(x)$. The computational burden turns out to be high, but less restrictive constraints must be a-priori imposed on the state evolution over the prediction horizon. An example of application of this approach is described in [31].

Once the feasibility problem has been solved, the stability issue must be considered. In case of persistent disturbances, it is not possible to require the asymptotic stability of the origin, but only "practical stability", i.e. convergence of the state trajectory to a robust positively invariant set containing the origin. The size of this set obviously depends on the worst feasible disturbance. Asymptotic stability of the origin can however be obtained for state dependent disturbances, provided that a suitable H_∞-type cost function is used in the optimization problem, see e.g. [31]. Concerning stability, in recent years it has been shown that the concept of Input to State Stability (ISS), see e.g. [59, 24], is the most appropriate tool for the analysis and synthesis of robust NMPC algorithms, see [32], [50].

The robust MPC methods now described can be applied quite easily to the problem of designing stabilizing regulators for distributed control structures. In fact, assuming that the overall system is made by a number of interconnected local subsystems, it is possible to see the mutual influences among subsystems as perturbation (disturbance) terms. In this perspective, it is advisable to design a local robust MPC law for any subsystem with robustness properties with respect to the perturbing actions performed by the other subsystems. This approach has been already followed in [49] where a completely decentralized and stabilizing MPC law has been designed under the main assumption that the interconnections are sufficiently weak. The proposed method heavily relies on the seminal results reported in [18], which provide well sounded theoretical foundations for the analysis of interconnected systems.
Also the distributed MPC methods relying on the exchange of information among the local subsystems regarding the future expected input or state trajectories (see the review paper [54] and the references therein), can be analyzed in the framework of robust MPC. In fact, it is possible to interpret the difference between the predicted and the real trajectories as disturbance terms to be suitably rejected. In order to apply the results of robust MPC in this perspective, it is mandatory to impose that any local subsystem's state and control trajectory does not differ too much with respect to the predicted one transmitted to the neighborhoods, so that the "equivalent disturbance term" is guaranteed to be bounded. This simple consideration motivates the introduction in some algorithms of additional constraints on the future input moves and state evolution, see for example [19] where however the MPC problem is not explicitly based on the robustness approach.

Chapter 6

Conclusions

The review of the literature on distributed state estimation reported in this deliverable has shown that new and efficient algorithms are required both for the *distributed estimation* and for the *partition-based estimation* problems, as they have been defined in the Introduction. In fact, distributed algorithms with guaranteed stability and convergence properties for linear and nonlinear systems are still largely missing. Moreover, most of the available results rely on the Kalman filtering approach, which cannot handle state and disturbance constraints, which are often to be considered in real problems. These considerations motivate the development of new Moving Horizon Estimation (MHE) schemes, which allow one to satisfy the two main requirements above mentioned, i.e. convergence of the estimates and disturbance estimation. These MHE algorithms have been already partially developed for linear systems within the project and will be extensively described in report D5.2. Further extensions will concern their extension to nonlinear systems as well as their applications to one or more benchmarks, such as the hydro power valley, object of Work Package 7.

As for distributed variance estimation, this is another topic of great interest. In fact, it is well known that the optimality properties of Kalman filters are based on the covariances of the noises affecting the state and measured variables. In MHE, these covariances are used to weight the terms to be minimized over a prescribed prediction horizon, namely the state disturbance and the estimation error over the considered time window. Many case studies have shown that a poor tuning of these weighting parameters lead to unsatisfactory results. Future research will concern the detailed analysis of the available algorithms in a number of significant cases as well as their extensions to the case of distributed estimation.

Finally, the wide theme of disturbance modeling can be conjugated in different ways. First, it appears to be quite straightforward to extend the centralized approach for MPC with tracking properties by means of proper estimation of the disturbances also to distributed control structures. On the other hand, disturbance attenuation in robust centralized model predictive control has produced a number of methods and results which can be exploited to design new and efficient distributed schemes where the mutual interactions among locally controlled subsystems can be viewed as disturbances to be properly rejected.

Bibliography

[1] N. Abdel-Jabbar, C. Kravaris, and B. Carnaham. A partially decentralized state observer and its parallel computer implementation. *Industrial & Engineering Chemistry Research*, 37(7):2741–2769, 1998.

[2] B.M. Åkesson, J.B. Jørgensen, and S.B. Jørgensen. A generalized autocovariance least-squares method for covariance estimation. *Proc. of the American Control Conference, New York City, USA, 2007*, pages 3713 – 3714, 2007.

[3] B.M. Åkesson, J.B. Jørgensen, N.K. Poulsen, and S.B. Jø rgensen. A generalized autocovariance least-squares method for Kalman filter tuning. *Journal of Process Control*, 18:769–779, 2008.

[4] P. Alriksson and A. Rantzer. Distributed Kalman filtering using weighted averaging. In *Proc. of the 17th International Symposium on Mathematical Theory of Networks and Systems*, Kyoto, Japan, 2006.

[5] P. Alriksson and A. Rantzer. Experimental evaluation of a distributed Kalman filter algorithm. In *In Proc. 46th IEEE Conference on Decision and Control*, pages 5499 – 5504, 2007.

[6] P. Alriksson and A. Rantzer. Model based information fusion in sensor networks. In *In Proc. 17th IFAC World Congress*, pages 4150–4155, Seoul, Korea, 2008.

[7] D.L. Alspach. A parallel filtering algorithm for linear systems with unknown time varying noise statistics. *IEEE Trans. on Automatic Control*, 19(5):552–556, 1974.

[8] A. Averbuch, S. Itzikovitz, and T. Kapon. Radar target tracking - Viterbi versus IMM. *IEEE Trans. on Aerospace and Electronic Systems*, 27(3):550–563, 1991.

[9] P.R. Bélanger. Estimation of noise covariance matrices for a linear time-varying stochastic process. *Automatica*, 10:267–275, 1974.

[10] T.M. Berg and H.F. Durrant-Whyte. Model distribution in decentralized multi-sensor data fusion. In *Proc. of the 10th American Control Conference*, pages 2292 – 2293, 1991.

[11] T. Bohlin. Four cases of identification of changing systems. *In R.K. Mehra & D.G. Lainiotis (Eds.), System Identification: Advances and case studies (1st ed.). New York: Academic Press.*, 1976.

[12] J. Brewer. Kronecker products and matrix calculus in systems theory. *IEEE Trans. in Circuits and Systems*, 25(9):772–781, 1978.

[13] G.C. Calafiore and F. Abrate. Distributed linear estimation over sensor networks. *International Journal of Control*, 82(5):868 – 882, 2009.

[14] B. Carew and P.R. Bélanger. Identification of optimum filter steady-state gain for systems with unknown noise covariances. *IEEE Trans. on Automatic Control*, 18(6):582–587, 1973.

[15] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri. Distributed Kalman filtering based on consensus strategies. *IEEE Journal on Selected Areas in Communications*, 4:622 – 633, 2008.

[16] D.W. Clarke and P. Gawthrop. Self-tuning control. *Proc. IEE*, 126:633 – 640, 1979.

[17] D.W. Clarke, C. Mohtadi, and S. Tuffs. Generalized predictive control - part I. The basic algorithm. *Automatica*, 23:137 – 148, 1987.

[18] S. Dashkovskiy, B.S. Rffer, and F.R. Wirth. An ISS theorem for general networks. *Mathematics of Control, Signals, and Systems*, 19:93–122, 2007.

[19] W.B. Dunbar. Distributed receding horizon control of dynamically coupled nonlinear systems. *IEEE Trans. on Automatic Control*, 52:1249–1263, 2007.

[20] B.A. Francis and W. M. Wonham. The internal model principle of control theory. *Automatica*, 12:457 – 465, 1976.

[21] H.R. Hashemipour, S. Roy, and A.J. Laub. Decentralized structures for parallel Kalman filtering. *IEEE Trans. on Automatic Control*, 33(1):88 – 94, January 1988.

[22] M.F. Hassan, G. Salut, M.G. Singh, and A. Titli. A decentralized computational algorithm for the global Kalman filter. *IEEE Trans. on Automatic Control*, 23(2):262 – 268, 1978.

[23] C.G. Hilborn and D.G. Lainiotis. Optimal estimation in the presence of unknown parameters. *IEEE Trans. on System Science and Cybernetics*, 5(1):38–43, 1969.

[24] Z.-P. Jiang and Y. Wang. Input-to-state stability for discrete-time nonlinear systems. *Automatica*, 37:857–869, 2001.

[25] M. Kamgarpour and C. Tomlin. Convergence properties of a decentralized Kalman filter. *Proc. 47^{th} IEEE Conference on Decision and Control*, pages 3205 – 3210, 2008.

[26] R.L. Kashyap. Maximum likelihood identification of stochastic linear systems. *IEEE Trans. on Automatic Control*, 15(1):25–34, 1970.

[27] U.A. Khan and J.M.F. Moura. Distributing the Kalman filter for large-scale systems. *IEEE Trans. on Signal Processing*, 56(10):4919 – 4935, October 2008.

[28] D. Limon, T. Alamo, and E. F. Camacho. Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties. In *Proc. of the 41st IEEE Conference on Decision and Control*, pages 4619–4624, Las Vegas, NV, USA, 2002.

[29] J. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, 2002.

[30] L. Magni, G. De Nicolao, and R. Scattolini. Output feedback and tracking of nonlinear systems with model predictive control. *Automatica*, 37:1601 – 1607, 2001.

[31] L. Magni, G. De Nicolao, R. Scattolini, and F. Allgower. Robust model predictive control of nonlinear discrete-time systems. *International Journal of Robust and Nonlinear control*, 13(3-4):229–246, 2003.

[32] L. Magni, D.M. Raimondo, and R. Scattolini. Regional input-to-state stability for nonlinear model predictive control. *IEEE Trans. on Automatic Control*, 51:1548–1553, 2006.

[33] L. Magni and R. Scattolini. Robustness and robust design of MPC for nonlinear systems. In R. Findeisen, L.T. Biegler, and F. Allgöwer, editors, *Nonlinear Predictive Control, Progress in Systems Theory Series*, volume 358, pages 239–254. Lecture Notes in Control and Information Sciences, Springer, 2007.

[34] R.K. Mehra. On the identification of variances and adaptive Kalman filtering. *IEEE Trans. on Automatic Control*, 15(2):175–184, 1970.

[35] R.K. Mehra. Approaches to adaptive filtering. *IEEE Trans. on Automatic Control*, 17:903–908, 1972.

[36] K.R. Muske and T.A. Badgwell. Disturbance modeling of offset-free linear model predictive control. *Journal of Process Control*, 12:617 – 632, 2002.

[37] A.G.O. Mutambara. *Decentralized estimation and control for multisensor systems*. CRC Press, 1998.

[38] A.G.O. Mutambara and H.F. Durrant-Whyte. Estimation and control for a modular wheeled mobile robot. *IEEE Trans. on Control Systems Technology*, 8(1):35 – 46, 2000.

[39] K.A. Myers and B.D. Tapley. Adaptive sequencial estimation with unknown noise statistics. *IEEE Trans. on Automatic Control*, 19(5):623–625, 1976.

[40] B.J. Odelson, M.R. Rajamani, and J.B. Rawlings. A new autocovariance least-squares method for estimating noise covariances. *Automatica*, 42(2):303–308, February 2006.

[41] R. Olfati-Saber. Distributed Kalman filter with embedded consensus filters. *Proc.* 44^{th} *IEEE Conference on Decision and Control - European Control Conference*, pages 8179 – 8184, 2005.

[42] R. Olfati-Saber. Distributed Kalman filtering for sensor networks. *Proc.* 46^{th} *IEEE Conference on Decision and Control*, pages 5492 – 5498, 2007.

[43] R. Olfati-Saber. Kalman-consensus filter: Optimality, stability and performance. *Proc.* 48^{th} *IEEE Conference on Decision and Control*, pages 7036 – 7042, 2009.

[44] R. Olfati-Saber and J. Shamma. Consensus filters for sensor networks and distributed sensor fusion. *Proc.* 44^{th} *IEEE Conference on Decision and Control - European Control Conference*, pages 6698 – 6703, 2005.

[45] G. Pannocchia and A. Bemporad. Combined design of disturbance model and observer for offset-free model predictive control. *IEEE Trans. on Automatic Control*, 52:1048–1053, 2007.

[46] G. Pannocchia and J. B. Rawlings. Disturbance models for offset-free model-predictive control. *AIChE Journal*, 49:426 – 437, 2003.

[47] M. Perillo and W. Heinzelman. *Wireless Sensor Network Protocols, Fundamental Algorithms and Protocol for Wireless and Mobile Networks*. CRC Hall, 2005.

[48] D.M. Prett and C.E. Garcia. *Fundamental Process Control*. Butterworths, 1988.

[49] D. Raimondo, L. Magni, and R. Scattolini. Decentralized mpc of nonlinear systems: an input-to-state stability approach. *Int. Journal of Robust and Nonlinear Control*, 17:1651–1667, 2007.

[50] D.M. Raimondo, D. Limon, M. Lazar, L. Magni, and E.F. Camacho. Min-max model predictive control of nonlinear systems: a unifying overview on stability. *European Journal of Control*, 15:5–21, 2009.

[51] S.V. Rakovic, A.R. Teel, D.Q. Mayne, and A. Astolfi. Simple robust control invariant tubes for some classes of nonlinear discrete time systems. In *Proc. of the 45th IEEE Conf. on Decision and Control*, pages 6397–6402, 2006.

[52] B.S. Rao and H.F. Durrant-Whyte. Fully decentralised algorithm for multisensor Kalman filtering. *IEE Proc. on Control Theory and Applications*, *D*, 138(5):413 – 420, September 1991.

[53] N. Sandell Jr, P. Varaiya, M. Athans, and M. Safonov. Survey of decentralized control methods for large scale systems. *IEEE Trans. on Automatic Control*, 23(2):108 – 128, April 1978.

[54] R. Scattolini. Architectures for distributed and hierarchical model predictive control. *Journal of Process Control*, 19:723–731, 2009.

[55] S.R. Searle. *Matrix algebra useful for statistics*. New York: Wiley, 1982.

[56] J. Sijs, M. Lazar, P.P.J. Van Den Bosch, and Z. Papp. An overview of non-centralized Kalman filters. *Proc. 17th IEEE Conference on decision and Control*, pages 739–744, 2008.

[57] D.D. Šiljac. *Large-scale dynamic systems. Stability and structure*. North Holland, 1978.

[58] D.D. Šiljac and M.B. Vukčević. On decentralized estimation. *International Journal of Control*, 27(1):113–131, 1978.

[59] E.D. Sontag. Smooth stabilization implies coprime factorization. *IEEE Trans. on Automatic Control*, 34:435–443, 1989.

[60] D.P. Spanos, R. Olfati-Saber, and R.M. Murray. Approximate distributed Kalman filtering in sensor networks with quantifiable performance. *Fourth International Symposium on Information Processing in Sensor Networks*, pages 133 – 139, 2005.

[61] A. Speranzon, C. Fischione, K.H. Johansson, and Sangiovanni-Vincentelli A.L. A distributed minimum variance estimator for sensor networks. *IEEE Journal on Selected Areas in Communications*, 26(4):609–621, 2008.

[62] S.S. Stanković, M.S. Stanković, and D.M. Stipanović. Consensus based overlapping decentralized estimation with missing observations and communication faults. *Automatica*, 45:1397 – 1406, 2009.

[63] S.S. Stanković, M.S. Stanković, and D.M. Stipanović. Consensus based overlapping decentralized estimator. *IEEE Trans. on Automatic Control*, 54(2):410 – 415, February 2009.

[64] J. Stoustrup. Plug & play control: Control technology towards new challenges. *European Journal of Control*, 15(3-4):311–330, 2009.

[65] S. Tilak, N.B. Abu-Ghazaleh, and W. Heinzelman. A taxonomy of wireless micro-sensor network models. *Mobile Computing and Communications Review*, 6:28 – 32, 2002.

[66] R. Vadigepalli and F.J. Doyle III. A distributed state estimation and control algorithm for plantwide processes. *IEEE Trans. on Control Systems Technology*, 11(1):119 – 127, January 2003.

[67] R. Vadigepalli and F.J. Doyle III. structural analysis of large-scale systems for distributed state estimation and control applications. *Control Engineering Practice*, 11:895 – 905, 2003.

[68] B. Warneke, M. Last, B. Liebowitz, and K.S.J. Pister. Smart dust: communicating with a cubic-millimeter computer. *Computer*, 34:44 – 51, 2001.