



UNIVERSIDAD NACIONAL DE COLOMBIA

Evaluation of convergence and computational efficiency of distributed state estimation techniques for large-scale systems

Mario Alejandro Giraldo Vásquez

Universidad Nacional de Colombia
Facultad de Ingeniería y Arquitectura
Departamento Ingeniería Eléctrica, Electrónica y Computación
Manizales, Colombia
2013

Evaluation of convergence and computational efficiency of distributed state estimation techniques for large-scale systems

Mario Alejandro Giraldo Vásquez

Tesis o trabajo de grado presentada(o) como requisito parcial para optar al título de:
Magíster en Ingeniería - Automatización Industrial

Director(a):

Prof. Jairo José Espinosa Oviedo, Ph. D.

Línea de Investigación:

Matemáticas Aplicadas

Grupo de Investigación:

Grupo de Automática de la Universidad Nacional - GAUNAL

Universidad Nacional de Colombia

Facultad de Ingeniería y Arquitectura

Departamento Ingeniería Eléctrica, Electrónica y Computación

Manizales, Colombia

2013

Agradecimientos

Completar esta tesis de maestría no hubiera sido posible sin un grupo muy especial de gente, dotados de una calidad intelectual y humana muy difícil de encontrar y con quienes he tenido la fortuna de trabajar durante 2 años. Por ello, quiero agradecer a mi director de tesis Jairo José Espinosa, quien me asesoró y me guió de manera muy acertada durante la maestría, potenciando mi crecimiento académico y profesional, y mi formación personal. Su experiencia y buena disposición han sido fundamentales para la realización exitosa de este trabajo. Así mismo agradezco a mis compañeros del grupo GAUNAL, siempre dispuestos a ayudar, asesorar y celebrar.

A mi familia, especialmente mis padres y mi hermano, es invaluable todo lo que han hecho por mí y lo siguen haciendo. Es corto el agradecimiento, pues no hay palabras que lo expresen realmente. Gracias!

A mi novia y amigos, siempre a mi lado, después de tanto tiempo, muchos años en algunos casos, y aún me siguen aguantando. Son gran motivación para mí, pues son personas increíbles que me retan y me dan ejemplo. Y obvio, gracias por las risas que siempre nos acompañan.

Finalmente a la Universidad Nacional de Colombia, institución que me ha acogido en mis estudios de pregrado y posgrado y me apoyó económicamente a través de la beca estudiantes sobresalientes de posgrado.

Abstract

Large scale systems is a denomination born due to the increasing size and complexity of dynamical systems and models used for their representation. These systems have a large amount of embedded sensors and actuators. Measuring systems variables of interest is very important in order to implement control schemes or just because it is necessary to know the dynamical system evolution.

In order to simplify the complexity of the system, it is divided into a number of subsystems, each one smaller and simpler than the overall system. Some system partitioning techniques are explored. Distributed state estimation states that for each subsystem, a local state estimator is implemented, and then, the sum of each local estimation gives the overall state estimation.

Distributed state estimators based on Kalman filtering, particle filtering and moving horizon estimation are explained and implemented in simulation using a benchmark, and the attributes of interest are compared: computational cost, convergence and prediction error, this is done by dividing the system into different number of subsystems. These comparisons point out which could be the adequate technique to be implemented given a large scale system. Finally, the Hydro-Power Valley large scale system is used to verify the hints given by the comparisons previously made.

Keywords: Large scale systems, sensor networks, state estimation, distributed implementation.

Resumen

Sistemas a gran escala es una denominación que surge a raíz del crecimiento en tamaño y complejidad de los sistemas dinámicos y los modelos usados para representarlos. Estos sistemas cuentan además con una gran cantidad de sensores y actuadores embebidos. Medir variables de interés de los sistemas es muy importante si se quiere implementar técnicas de control o simplemente porque es necesario saber la evolución del sistema dinámico.

Para simplificar la complejidad del sistema, este se divide en un número determinado de subsistemas, cada uno más pequeño y sencillo que el sistema completo. Se exploran algunas técnicas para dividir sistemas. Se propone implementar un estimador de estado para cada subsistema y luego sumar la información de cada uno para obtener una estimación global. A este procedimiento se le conoce como estimación de estado distribuida.

Estimadores de estado distribuidos basados en filtro de Kalman, filtros de partículas y estimadores de horizonte deslizante son explicados e implementados en simulación en un

benchmark, y se comparan atributos de interés: costo computacional, convergencia y error de predicción, esto se dividiendo el sistema en diferente cantidad de subsistemas. Estas comparaciones dan un indicio de cual podría ser la técnica indicada a implementar dado un sistema dinámico a gran escala. Finalmente, el sistema a gran escala Hydro-Power Valley es usado para verificar las pistas arrojadas por las comparaciones hechas.

Palabras clave: Sistemas a gran escala, redes de sensores, estimación de estado, implementación distribuida.

Contents

Abstract	vi
Content	x
List of Figures	x
List of Tables	xii
Introduction	1
Chapter 1. State estimation for large-scale systems	4
1.1. Problem statement	5
1.2. Distributed state estimation	8
1.3. Systems partitioning methods. Brief review.	9
1.3.1. Partitioning techniques based on Relative Gain Array	9
1.3.2. Decomposition methods based on Gramians	11
1.3.3. Nodal transformation matrix	11
1.4. Discussion on distributed systems approach	12
Chapter 2. Distributed state estimation techniques	14
2.1. Benchmark description	14
2.2. Distributed Kalman filters	19
2.2.1. Kalman filtering	19
2.2.2. Distributed and Decentralized Kalman filter	21
2.2.3. Distributed Kalman filter with state consensus	22
2.2.4. Distributed Kalman filter with diffusion strategy	24
2.3. Distributed particle filter	26
2.3.1. Particle filtering	26
2.3.2. Distributed particle filter scheme	28
2.4. Distributed moving horizon estimator	29

2.4.1. Moving Horizon Estimation	29
2.4.2. Distributed moving horizon estimation scheme	31
2.5. Other distributed state estimators	32
Chapter 3. Distributed state estimation techniques comparison	34
3.1. Distributed Kalman filters unification	34
3.1.1. Modified distributed Kalman filter with state consensus	35
3.1.2. Modified distributed Kalman filter with diffusion strategy	36
3.2. Distributed state estimators implementations for different amount of nodes	37
3.2.1. 3 nodes system partitioning	37
3.2.2. 4 nodes system partitioning	40
3.3. Distributed state estimators performance analysis and comparison	43
3.3.1. Metrics for attributes evaluation	43
3.3.2. Attributes evaluation results	44
Chapter 4. Simulation results	48
4.1. Hydro Power Plant system description	48
4.1.1. System modelling	49
4.2. Hydro-Power Plant system partition	53
4.3. Distributed state estimation techniques simulations	56
Conclusions	59
Future work	60
Bibliography	61

List of Figures

1.1 Large-scale system with N interacting subsystems.	5
1.2 Node i and its neighbourhood N_i .	7
2.1 Spatial distribution of a rod.	15
2.2 Rod spatial distribution. Heat sources and sensors positions.	17
2.3 Evolution of temperature along the rod.	18
2.4 Dynamic response of the system.	18
2.5 Rod partitioning into two subsystems.	19
2.6 Nodal representation of two subsystems.	19
2.7 Dynamic response of the system with centralized Kalman filter.	20
2.8 Dynamic response of the system with DDKF.	22
2.9 Dynamic response of the system with DKF-SC.	24
2.10 Dynamic response of the system with DKF-DS.	26
2.11 Dynamic response of the system with centralized Particle filter.	28
2.12 Dynamic response of the system with distributed Particle filter.	29
2.13 Dynamic response of the system with moving horizon estimator.	31
2.14 Dynamic response of the system with distributed moving horizon estimator.	32
3.1 Dynamic response of the system with modified DKF-SC.	36
3.2 Dynamic response of the system with modified DKF-DS.	37
3.3 Nodal representation of 3 subsystems.	38
3.4 DDKF estimation for 3 nodes distribution.	38
3.5 Modified DKF-SC estimation for 3 nodes distribution.	38
3.6 Modified DKF-DS estimation for 3 nodes distribution.	39

3.7	Distributed Particle filter for 3 nodes distribution.	39
3.8	Distributed moving horizon estimator for 3 nodes distribution.	40
3.9	Nodal representation of 4 subsystems.	40
3.10	DDKF estimation for 4 nodes distribution.	41
3.11	Modified DKF-SC estimation for 4 nodes distribution.	41
3.12	Modified DKF-DS estimation for 4 nodes distribution.	42
3.13	Distributed Particle filter for 4 nodes distribution.	42
3.14	Distributed moving horizon estimator for 4 nodes distribution.	43
4.1	Hydro-Power Valley system overview [1].	49
4.2	Reach spatial discretization.	50
4.3	Hydro-Power Plant node connections.	53
4.4	Dam disposition.	55
4.5	HPV dynamic response.	56
4.6	Kalman filter estimation.	57
4.7	MHE estimation.	57

List of Tables

1.1 Common applications of distributed state estimation techniques.	7
2.1 Rod model parameters.	17
2.2 Classic Kalman filter algorithm.	20
2.3 Distributed and decentralized Kalman filter algorithm.	21
2.4 Distributed Kalman filter with state consensus algorithm.	23
2.5 Distributed Kalman filter with diffusion strategy algorithm (Information form).	25
2.6 Particle filter algorithm.	27
3.1 Attributes evaluation for centralized estimators.	44
3.2 Attributes evaluation for 2 nodes system partitioning.	45
3.3 Subsystem 1 computational cost evaluation for 2 nodes system partitioning.	45
3.4 Attributes evaluation for 3 nodes system partitioning.	46
3.5 Subsystem 1 computational cost evaluation for 3 nodes system partitioning.	46
3.6 Attributes evaluation for 4 nodes system partitioning.	47
3.7 Subsystem 1 computational cost evaluation for 4 nodes system partitioning.	47
4.1 Attributes evaluation for centralized estimators in HPV system.	58
4.2 Attributes evaluation for distributed estimators in HPV system.	58

Introduction

State estimation is one of the most useful concepts in control theory when there are variables in a dynamical system that are unknown and cannot be measured due to process nature, technological limitations or economical reasons. The knowledge of the states of a system is necessary in order to implement control strategies or just because the state information is of great interest. There are a large number of state estimation techniques that can retrieve the whole state vector of a system at any time using available input and output information of a system.

The complexity of the estimation techniques has increased to overcome the challenges encountered when trying to retrieve the variables of interest of more complex dynamical systems. For instance, the Kalman filter, originally intended for linear systems, has suffered several modifications since Rudolf E. Kalman proposed it in his seminal paper in 1960 [2]. An extended Kalman filter was proposed to deal with nonlinear systems, moving horizon state estimators were developed in order to consider dynamical systems and control systems constraints, just to name a few. An exhaustive review of state estimators is made in [3] comparing their basic formulation and how they were derived, focusing on nonlinear systems.

Recent challenges designing control systems and estimation structures are approaching dynamical systems with large number of states, inherent complex dynamics. Such systems are often referred as large-scale systems. Control systems for such systems require high computational resources due to the large amount of information to be handled and processed. It produces economical and performance impact, sometimes making infeasible the control system implementation.

Large-scale systems problem has been addressed partitioning the system into smaller and simpler subsystems. Decentralized and distributed processing of the information approaches have been studied in the last two decades [4, 5]. Instead of having one unique centralized

or global information processing unit, there are several subsystems processing less amount of information, and the sum of their contributions give, as result, the global system response. The main difference between decentralized and distributed processing is that in the distributed approach, each subsystem share its information with its neighbouring subsystems, meanwhile in the decentralized approach, each subsystem processes information on its own, not receiving nor sharing information from other subsystems.

The work in this thesis aims to review distributed estimation techniques proposed in the literature and compares their performance. Such comparison is made by evaluating attributes such as the convergence time of the estimation, the prediction error and the computational cost. Most of the works on distributed state estimators take a given distributed structure and modify it to achieve a better performance in terms of minimizing the deviation from the actual state to be estimated. So far, the impact of distributing a system into different number of subsystems and the implications on the aforementioned attributes have not been evaluated.

The general objective of this thesis is to implement in simulation and to evaluate, through attributes such as prediction error, convergence time of the estimation and computational cost, distributed state estimation techniques in order to establish selection criteria of distributed state estimators for large scale systems. The specific objectives established to accomplish the general objective are stated as follows:

- To identify the most commonly used distributed state estimation techniques proposed in the literature.
- To perform a literature review on available system partitioning techniques and how subsystems must be obtained accordingly to each identified distributed state estimation technique.
- To implement in simulation at least three of the identified distributed estimation techniques in a benchmark using at least three system partitions, with different amount of subsystems.
- To compare the performance of the implemented estimation techniques evaluating attributes such as: prediction error, convergence time of the estimation and computational cost.
- To establish selection criteria of the estimation technique that guarantees a reliable global estimation from the local estimations of each subsystem.
- To validate by means of simulation of the Hydro-Power Valley (HPV) system the established selection criteria.

This report is organized as follows: the first chapter presents a literature review and brief discussion on large-scale systems, distributed state estimators and system partitioning techniques. At the end of the chapter a benchmark is presented as test bed to implement in simulation distributed state estimation schemes. The second chapter presents a selection of distributed state estimations schemes, each scheme is described and simulated using the

benchmark. Algorithms of each scheme are presented, synthesised step by step, as a tutorial for future references. In the third chapter, the performance of the distributed estimators implemented in chapter two is compared. In the final chapter, the Hydro-Power Valley system model simulation is used to validate the results obtained in previous chapters.

CHAPTER 1

State estimation for large-scale systems

Large-scale systems are comprised by many interacting subsystems with highly complex dynamics and mutual influences, they are networked systems usually with many embedded sensors or actuators.

Technological and economical reasons motivate the development of process plants, manufacturing systems and traffic networks with an ever increasing complexity. These large-scale systems can be difficult to control with a centralized control structure due to the inherent computational complexity, since there are robustness and reliability problems and due to communication bandwidth limitations [6, 7, 8]. Control strategies aim to optimize the system operation which lead to optimization of the use of resources, economic benefits or precision augmentation.

Control systems such as Model Predictive Control (MPC) are widely used as control strategies for large-scale systems [9, 10]. MPC optimal performance, for instance, depends on an accurate estimation of information of the system, i.e. the estimation of the state vector or variables of interest of a system.

The analysis, characterization, modelling, control or state estimation of large-scale systems is simplified by partitioning the complete or centralized system into a number of subsystems, each one easier to handle than the complete system. Early contributions on the subject addressed the problem of data fusion [11], distributing a model when having multiple sensors.

The concepts of parallel and distributed state estimation are formalized in [4], where it is stated that distributed state estimation is a suitable solution for complex systems state estimation. Each subsystem could share information with the surrounding subsystems or

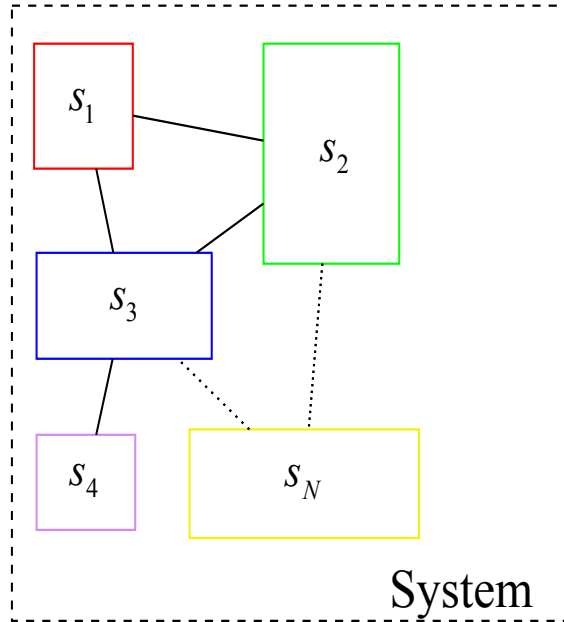


FIGURE 1.1. Large-scale system with N interacting subsystems.

with all of them. Once the information about the states is shared, a global estimation is achieved. Controllers can make decisions to meet process requirements or to optimize them, such controllers can be local as well.

1.1. Problem statement

For large-scale systems, complex by nature, distributed state estimators are implemented in order to retrieve the variables of interest, since the distributed and decentralized approaches address the computational scalability issue through decomposition of the process model into multiple subsystems that interact to produce identical state estimates as that of the centralized model.

As stated before, the distributed and decentralized approaches are proposed based on the assumption that smaller subsystems (e.g. less number of states) are easier to handle than a centralized system. And that the sum of the information of each subsystem leads to obtain the global behaviour of a system. But challenges arise when partitioning a system. The main challenge is to obtain a global reliable state estimation, even if the measurements performed at each subsystem are not sufficient to guarantee local observability (subsystem observability).

Consider a large-scale system described by a linear dynamic model as:

$$(1.1) \quad x(k+1) = Ax(k) + Bu(k)$$

$$(1.2) \quad y(k+1) = Cx(k)$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{n \times p}$ and n is large.

The distributed systems approach states that the system can be partitioned into N low order subsystems. Each subsystem can be represented as follows:

$$(1.3) \quad x_i(k+1) = A_i x_i(k) + B_i u_i(k)$$

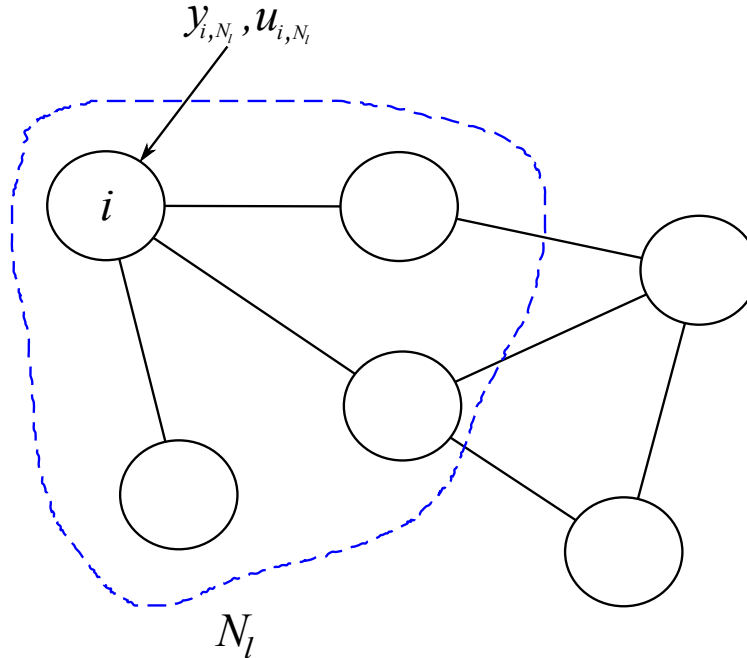
$$(1.4) \quad y_i(k) = C_i x_i(k)$$

Equations 1.3 and 1.4 fully represent a local dynamical system, therefore local control structures and state observers can be implemented. $x_i(k) \in \mathbb{R}^{n_i}$ is the state vector of a generic sub model. Index i identifies the i -th subsystem and $\text{rank}(x) > \text{rank}(x_i)$.

A subsystem is usually referred as a node. This denomination arises when the sensing procedure in a system is considered to be performed by a sensor network¹. A node can be comprised by one or more sensors. A subsystem can be obtained when at least one sensor takes measurements that can be related to the evolution of the subsystem according to a linear state space model such as 1.3 and 1.4. The term node is extended to refer a subsystem and its model representation.

A neighbourhood is a set of interacting nodes, since a node can exchange information with one or more nodes. Neighbourhood N_i of the i -th node of a system includes interacting nodes and itself.

¹A sensor network for large-scale systems is assumed to be a large number of interconnected sensors.

FIGURE 1.2. Node i and its neighbourhood N_i .

The increasing number of applications involving large-scale systems means that the availability of distributed state estimators for each application must be guaranteed. Several techniques have been proposed to address the distributed state estimation problem. The most common distributed approximations that have been explored are based on Kalman filters. Moving horizon estimation (MHE) and particle filters based distributed state estimation schemes are also proposed in the literature. Table 1.1 presents some references and possible applications (test bed systems for exemplification) of the three distributed state estimation approaches. Furthermore, within the approximations, different estimators schemes have been proposed aiming to improve the accuracy of the estimation and to reach global convergence.

Estimation technique	Application	References
Kalman filters	Wireless sensor networks, mobile localization, cascaded systems	[12, 13, 14, 15, 16, 17]
Moving horizon estimators	Industrial processes, manufacturing, robust control	[18, 19, 20, 21, 22, 23, 24]
Particle filters	Robot localization, wireless sensor networks	[25, 26, 27, 28]

TABLE 1.1. Common applications of distributed state estimation techniques.

1.2. Distributed state estimation

There are two ways to approach the distributed state estimation problem:

The first approach consists of schemes that perform the estimation of the whole state vector in each subsystem, and then perform a centralized consensus procedure. In this approach, any sensors of the network measure some variables, computes a local estimate of the overall state of the system and transmits only to its neighbours the computed state estimation and some other information, e.g. corresponding covariances, depending on the estimation scheme.

Each node should know the dynamic model of the overall system, therefore the subsystem model is:

$$(1.5) \quad x_i(k+1) = Ax_i(k) + Bu_i(k)$$

At each node, local measurements from the overall system are taken, and the local measurement model is:

$$(1.6) \quad y_i(k) = C_i x(k)$$

Then main challenge with this approach is reaching a reliable global estimation through a global consensus of the local estimations [27, 29, 30]. Another justification for the consensus strategy is to reduce their uncertainty of the estimation obtained by each local estimator.

The main drawbacks of this approach is that it is a full order problem, since each node must know the the complete dynamical model of the system, and the consensus must be achieved even if the measurements performed at each node are not sufficient to guarantee observability of the local vector state.

The second approach does not consider a global consensus, since it considers that each node estimates the local state vector. This approach is usually referred as partition-based state estimation [20, 21] and adjusts better to the statement made to justify using decentralized and distributed approaches for large-scale systems: it gives place to low order estimation problems relying in local dynamical and measurement models.

As the previous approach, only neighbouring subsystems share information, but in this case the subsystems can have overlapping states or not. This means that subsystems could not share states. In such case the local estimation is purely decentralized, but the local model dynamics consider the effect of the neighbouring states [12, 31, 32]:

$$(1.7) \quad x_i(k+1) = A_i x_i(k) + B_i u_i(k) + u_i^t(k)$$

where $u_i^t(k)$ represents the effect of states of surrounding subsystems. The local measurement model 1.4 holds, since the measurements are taken locally from the local state vector.

1.3. Systems partitioning methods. Brief review.

Designing state observers or control structures for large-scale systems is very difficult due to the complex nature of such systems, as stated before. The partitioning procedure is performed mainly with two purposes: first, the decomposition of the system into a number of smaller interacting subsystems. Second, to lower the computational burden at each subsystem compared to solve the original one (centralized).

The main issue regarding partitioning techniques is obtaining well conditioned subsystems. They need that the sensors and actuators give enough information about the system dynamics in order to reconstruct optimally the whole set of states of the system. A state estimation technique can perform poorly if the subsystem is ill-conditioned.

Formal studies on system partitioning are found in [33, 34, 35]. Some techniques and procedures proposed to perform the system partitioning are based on Gramians, relative gain array (RGA) and on the use of linear transformation matrices.

The partitioning procedure decomposes a large-scale system into multiple subsystems. Such subsystems interact to produce identical state estimates that would have been obtained using a centralized estimation technique.

1.3.1. Partitioning techniques based on Relative Gain Array. Relative Gain Array based techniques consider linear time-invariant systems, as the one described by equations 1.1 and 1.2. The transfer function of the system is $G(s) = C(sI - A)^{-1} + B$, with steady state gain $G_0 = G(0)$. Individual elements from G_0 are denoted by g_{ij} , with $i, j = 1, 2, \dots, m$. It is assumed that $G(s)$ has stable poles. RGA performs an input-output analysis in order to identify the relationships of such variables and according to how related they are, perform the system decomposition.

1.3.1.1. *Relative Gain Array* [36]. Relative Gain Array is defined as:

$$(1.8) \quad \Lambda = G_0 \times (G_0^{-1})^T$$

where the multiplication is made element by element. Some properties of matrix λ are:

- The sum of the elements of any row is equal to 1.

- The sum of the elements of any column is equal to 1.
- Λ is equal to the identity if $G(s)$ is a diagonal or triangular matrix.

The elements of the Λ matrix represent the ratio between the process gain for the pairing $y_i - u_j$ in an isolated loop and the process gain in the same loop when all other loops are closed. The variables selection in RGA method focuses on adequate pairing of inputs and outputs $y_i - u_j$, which aims for minimizing the complexity of a desired control structure. It is advisable to select those pairs that maintain approximately the same gain in open-loop and closed-loop configurations, i.e. $\lambda_{ij} \simeq 1$, pairings where $\lambda_{ij} < 0$ must be avoided.

The elements of matrix λ are defined as:

$$(1.9) \quad \lambda_{ij} = \frac{\left(\frac{\partial y_i}{\partial u_j} \right)_{u_i \neq j}}{\left(\frac{\partial y_i}{\partial u_j} \right)_{y_k \neq i}}$$

1.3.1.2. *Niederlinski Index.* It is possible that the analysis of the RGA can not provide an unique, dominating solution for pairing input and output variables, which means there are several equivalent solutions. When this problem arises, Niederlinski index is used [37]. The Niederlinski index (NI) considers that for a given choice of the input-output pairings, let \bar{G}_0 be the matrix obtained from G_0 by setting to zero all the elements that do not correspond to the selected $y_i - u_j$ input-output pairs. The NI $N_{\bar{G}_0}$ is:

$$(1.10) \quad N_{\bar{G}_0}(G_0) = \frac{\det(G_0)}{\det(\bar{G}_0)}$$

The following pairing criterion is proposed: among the possible sets of pairings selected after analysing RGA, choose the pairs with a positive NI. It can be proved that a decentralized control configuration corresponding to a positive value of the NI, has the potential to be Integral Controllable with Integrity (ICI).

1.3.1.3. *Partial Relative Gain.* This method has been proposed in [38] to guarantee that the control configuration is ICI and to derive rules where the pairing analysis from RGA gives multiple solutions. In order to introduce the Partial Relative Gain (PRG) index, it is assumed that matrices G_0 and Λ are:

$$(1.11) \quad G_0 = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}, \Lambda = \begin{bmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{bmatrix}$$

where G_{11} and Λ_{11} have the same dimensions, while G_{22} is assumed to be nonsingular. It is possible to show that

$$(1.12) \quad \Lambda_{11} = G_{11} \times (\bar{G}_{11}^{-1})^T, \bar{G}_{11} = G_{11} - G_{12}G_{22}^{-1}G_{21}$$

The PRG procedure can be summarized as follows:

- From RGA analysis choose input-output $y_i - u_j$ pairings.
- Reorder matrix G_0 as in 1.11 and make G_{22} corresponds to the selected input-output pair.
- Recompute the RGA for the subsystem G_{11} and obtain $\Lambda_P = \bar{G}_{11} \times (\bar{G}_{11}^{-1})^T$.
- Choose new pairing from Λ_P .
- Repeat procedure until all $u - y$ pairings have been chosen.

With this procedure, PRG is an attempt to consider closed-loop information related to the chosen decentralized control structure. Notice that the first step is not unique.

1.3.2. Decomposition

methods

based on Gramians. Gramians describe observability and controllability properties of a given stable linear system as stated in [39] and [40]. In such references, a definition of observability and controllability gramians is reported.

Definition. For a stable system, equations 1.1 and 1.2, the controllability gramian P and the observability gramian Q are symmetric non negative definite matrices which satisfy the Lyapunov equations

$$(1.13) \quad AP + PA^T + BB^T = 0$$

$$(1.14) \quad A^TQ + QA^T + C^TC = 0$$

Alternatively, matrices P and Q can be expressed as:

$$(1.15) \quad P = \int_0^\infty e^{At}BB^Te^{A^Tt}dt$$

$$(1.16) \quad Q = \int_0^\infty e^{A^Tt}C^TCe^{At}dt$$

The gramians quantify how hard is to control and to observe the system states.

1.3.3. Nodal transformation matrix. A nodal transformation matrix is presented in [41] as a mathematical tool used to perform a model distribution process. The main objective of the procedure is to construct reduced order models from a global (centralized) model by creating local state vectors which consist of locally relevant states. The procedure guarantees that there is a dynamic equivalence between local and global models. This procedure can be used for decentralized or distributed estimation when the system does not present a fully

connected topology.

Nodal state vector $x_i(k)$, local measurements $y_i(k)$ and local system dynamic matrix A_i , for instance, are related to the global state vector $x(k)$, global measurement $y(k)$ and system dynamic matrix A , accordingly, to

$$(1.17) \quad \begin{aligned} x_i(k) &= T_i x(k) \\ y_i(k) &= H_i y(k) \\ A_i &= T_i A T_i^+ \end{aligned}$$

where $C_i = H_i C T_i^+$, and H_i is the transformation for selecting local measurements from the global measurement vector. The superindex $+$ denotes the Moore-Penrose generalized inverse. $T_i^+ = T_i^{-1}$ for nonsingular T_i .

T_i is a linear nodal transformation matrix. Choosing T_i must satisfy that the local states can be locally estimated and controlled as optimally as they would be centrally, without having to propagate information between unconnected nodes. T_i picks states or combinations of states from the global vector to form a reduced order local state. This approach satisfy the need of propagating information between, for example, two unconnected nodes which have common state space observed by either or both nodes.

Given a dynamical system, the nodal transformation matrix T_i is dependent on the way states physically influence each other, that is, T_i is dependant on the state transition matrix. In [41] can be found how to define T_i according to the form that a state transition matrix is constructed given the state space model of a dynamical system.

1.4. Discussion on distributed systems approach

- Regarding systems partitioning.

None of the proposed methods has a marked implementation preference over the others. In fact, none of the reviewed works on distributed state estimation schemes discuss how the subsystems were obtained or which system partitioning technique was used.

The transformation matrix T_i is explicitly used in [41, 42], and seems to be a reasonable good choice for system partitioning. But the truth is that the success of this method relies on the knowledge one has on the system dynamics.

One possible solution for the system partitioning problem appears when it is considered first, which should be the correct or adequate location for the sensors. Optimal measurement methods and sensor positioning within a system is proposed in [43]. The idea is to allocate

sensors optimally so they give enough and reliable information from a system. This could lead to a trivial system partitioning based on the location of such sensors. However, this solution is not always possible, since usually it is intended to retrieve the states of an already functioning or designed system, in order to control and to optimize it. Besides, some variables may be too expensive or impossible to measure.

The general statement presented in the literature is that the partitioning procedure should give place to "*Well conditioned systems*". But, what is considered a well conditioned system, is an open discussion. It depends on which property interests the most. A system that guarantees full local observability is a well conditioned system? Is it that one in which the model uncertainties are reduced? There is no answer that can satisfy all the system partitioning procedures.

- Regarding distributed state estimators.

None of the reviewed works show a procedure to check or to guarantee local observability so a given distributed state estimation technique can be implemented. It is only stated that each subsystem must have at least one sensor. Even when several authors state that the distributed state estimation procedure calls for the availability of schemes that guarantee the asymptotic convergence of the local estimates. For instance, there is no established relationship that link noise constraints to convergence properties of the local estimations.

Estimating the states of a large-scale system by means of the distributed state estimation approach has to consider not only the system partitioning or the distributed state estimation scheme, but some other issues that have been previously exposed [9, 44, 45]. For example, problems related to communication protocols and synchronization of the information exchange; good performance of the local estimators requires information exchange that could be corrupted due to information loss or transmission frequency limitations. The discussed issues have yet not been solved from a distributed point of view.

In summary, the distributed approach is not a well conditioned problem since it lacks of formalization and unification of concepts. The proposed procedures and methods rely on heuristics and the knowledge a designer has over an observer structure and a given dynamical system.

CHAPTER 2

Distributed state estimation techniques

In order to compare the performance of some of the distributed state estimation schemes presented in table 1.1, a heat rod model is chosen as benchmark for the estimators implementation. The selection criteria of the schemes to be implemented obey to reasons such as the availability of convergence tests made by the authors, amount of references of the proposed scheme, or, as in the case of the moving horizon estimator based schemes, there is only one scheme proposed in the literature.

It is worth mentioning that the centralized or classical implementations of each algorithm is presented and implemented as well. In this chapter, the algorithms to implement the estimators are fully described and the estimated dynamic behaviour of the benchmark is compared with the reference dynamic behaviour of the temperature in the rod.

2.1. Benchmark description

The heat rod dynamical system has been used as benchmark for distributed state estimation schemes comparisons in works as the one presented in [46] and the *Hierarchical and distributed model predictive control of large-scale systems* (HD-MPC) project ². The features of the rod model that are attractive for distributed applications are the possibility of dividing the rod into any number of slices, therefore each slice represents a state. Also it is possible to consider any number of heat sources and temperature sensors that can be locate anywhere along the rod.

Consider a solid rod of length L where heat conduction and convection phenomena arise.

²Project sponsored by the European Union. Homepage: <http://www.ict-hd-mpc.eu/>

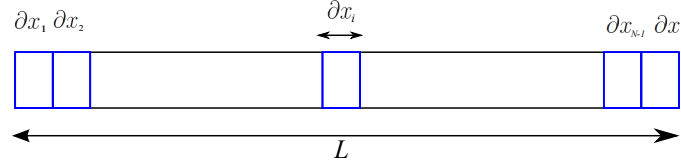


FIGURE 2.1. Spatial distribution of a rod.

Rod model.

$$(2.18) \quad \rho C_p (A_T \partial x) \frac{\partial T}{\partial t} = \left(\kappa \frac{\partial^2 T}{\partial x^2} + g(x, t) \right) A_T \partial x$$

where ρ is the density of the rod, C_p is the heat capacity per unit of mass, κ is the thermal conductivity, P is the perimeter of the cross-sectional circumference, ∂x is the width of the slice, $A_T = P \partial x$ is the area exposed to the environment, T is the temperature inside the slice, x and t are the spatial and temporal variables, and $g(x, t)$ is a generation function described as:

$$(2.19) \quad g(x, t) A_T \partial x = \dot{Q}(x, t) P \partial x + h (T_{env} - T(x, t)) P \partial x$$

with $\dot{Q}(x, t)$ the heater power per unit of area, h the convection coefficient and T_{env} the temperature of the environment.

Replacing 2.19 in 2.18, a final partial differential equation is obtained:

$$(2.20) \quad \frac{\partial T}{\partial t} = \frac{1}{\rho C_p} \left[\kappa \frac{\partial^2 T}{\partial x^2} + \frac{P}{A_T} \dot{Q} + \frac{hP}{A_T} (T_{env} - T) \right]$$

Model discretization. In order to solve numerically the previous model, partial derivatives are approximated. A straightforward way to perform such approximation is to apply the finite differences method. This method uses an approximation of the derivatives based on a truncation of the Taylor series at the first order term. Then the derivatives can be expressed as:

$$(2.21) \quad \begin{aligned} \frac{\partial u}{\partial h} &\approx \frac{u_{i+1} - u_i}{\Delta h} \\ \frac{\partial u}{\partial h} &\approx \frac{u_i - u_{i-1}}{\Delta h} \\ \frac{\partial u}{\partial h} &\approx \frac{u_{i+1} - u_{i-1}}{2\Delta h} \end{aligned}$$

where i is a spatial index, used as discretization variable. Equations in 2.21 are known as forward, backward and central approximations of the derivative, respectively. If the previous approximations are successively applied, the second order derivatives can be found as:

$$(2.22) \quad \begin{aligned} \frac{\partial^2 u}{\partial h^2} &\approx \frac{u_{i+1} - 2u_{i-1} + u_i}{\Delta h^2} \\ \frac{\partial^2 u}{\partial h^2} &\approx \frac{u_i - 2u_{i-1} + u_{i-2}}{\Delta h^2} \\ \frac{\partial^2 u}{\partial h^2} &\approx \frac{u_{i+1} - 2u_i + u_{i-1}}{2\Delta h^2} \end{aligned}$$

If the approximations are applied to the rod equation, following expression is considered for the heat rod modelling:

$$(2.23) \quad \frac{\partial T_i}{\partial t} = \frac{1}{\rho C_p} \left[\kappa \frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} + \frac{P}{A_T} \dot{Q}_i + \frac{hP}{A_T} (T_{env} - T_i) \right]$$

To model the ends of the rod, an additional assumption must be made: the conduction phenomena is only given in one dimension, across x -axis. Therefore, two more expressions must be considered to model the heat rod:

$$(2.24) \quad \begin{aligned} \frac{\partial T_i}{\partial t} &= \frac{1}{\rho C_p} \left[\kappa \frac{T_{i+1} - T_i}{\Delta x} + \frac{P}{A_T} \dot{Q}_i + \frac{hP}{A_T} (T_{env} - T_i) \right] \\ \frac{\partial T_i}{\partial t} &= \frac{1}{\rho C_p} \left[\kappa \frac{T_{i-1} - T_i}{\Delta x} + \frac{P}{A_T} \dot{Q}_i + \frac{hP}{A_T} (T_{env} - T_i) \right] \end{aligned}$$

Equations in 2.24 are used to model the ends of the rod, and equation 2.23 is applied in the middle of the rod.

State space model. The model of the heat rod is:

$$(2.25) \quad A_R = \frac{\kappa}{\rho C_p \Delta x} \begin{bmatrix} -1 & 1 & & 0 \\ 1 & -2 & 1 & \\ & & \ddots & \\ & & & 1 & -2 & 1 \\ 0 & & & & 1 & -1 \end{bmatrix} - \frac{hP}{\rho C_p A_T} \begin{bmatrix} 1 & & & & & 0 \\ & 1 & & & & \\ & & \ddots & & & \\ & & & & 1 & \\ 0 & & & & & 1 \end{bmatrix}$$

Symbol	Parameter	Value	Units
ρ	Density	2700	$Kg \cdot m^{-3}$
C_p	Heat capacity per mass unit	900	$J \cdot K^{-1} \cdot Kg^{-1}$
κ	Thermal conductivity	230	$W \cdot m^{-1} \cdot K^{-1}$
h	Convection coefficient	10	$W \cdot m^{-2} \cdot K^{-1}$
\dot{Q}	Heat power	2000	W
T_{env}	Environmental temperature	298	K

TABLE 2.1. Rod model parameters.

$$(2.26) \quad B_R = \begin{bmatrix} \frac{P}{\rho C_p A_T} & 0 & \dots & 0 \\ 0 & \frac{P}{\rho C_p A_T} & & \vdots \\ \vdots & & \ddots & \\ & & & \frac{P}{\rho C_p A_T} & 0 \\ 0 & \dots & 0 & \frac{P}{\rho C_p A_T} & \frac{P}{\rho C_p A_T} \end{bmatrix}$$

The system states are the temperatures of each slice, $x = [T_1 T_2 \dots T_{N_i}]^T$, and the system inputs are the heat power

Simulation set up. To simulate the dynamic response of the system, a solid rod of 2 meters long is considered. The following considerations are made:

- The system is spatially partitioned into 20 slices.
- 2 heat sources are considered, located at slices 5 and 16.
- The measurements of the system are taken 4 temperature sensors, located at slices 5, 8, 13 and 16.

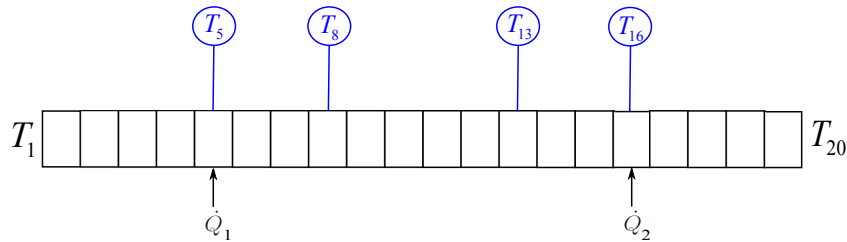


FIGURE 2.2. Rod spatial distribution. Heat sources and sensors positions.

The value of the parameters used for simulation are shown in next table.

Next figures present the dynamic response of the system. At the beginning of the simulation the temperature in the rod is the environment temperature and at 20000 seconds, the heat sources begin operation. A full description of the rod model and parameters values used for simulation are presented in [47].

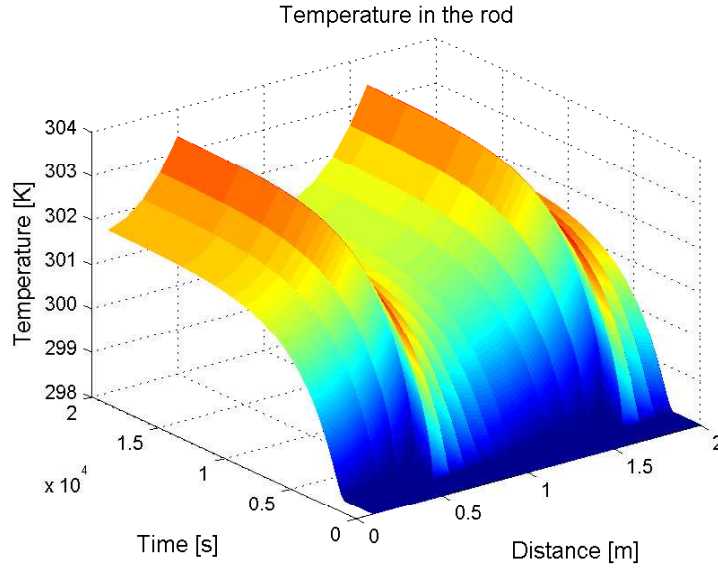
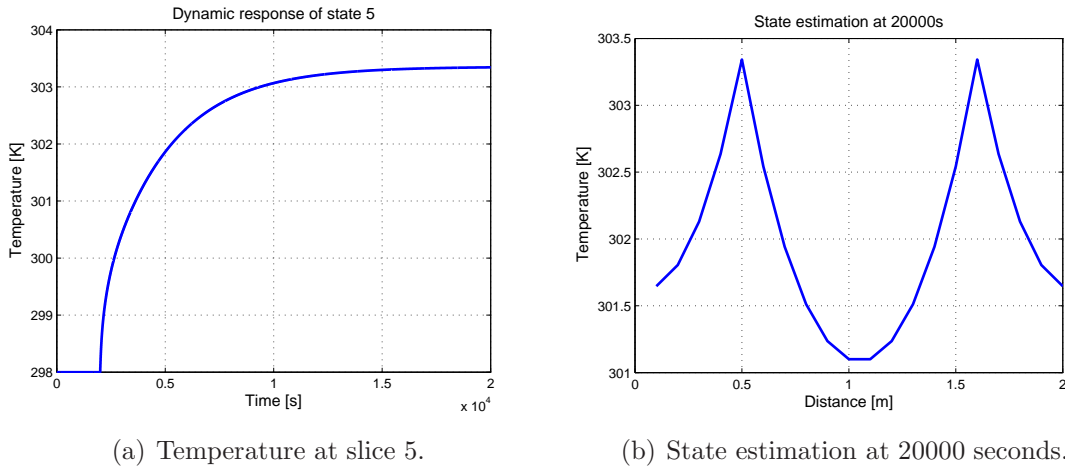


FIGURE 2.3. Evolution of temperature along the rod.



(a) Temperature at slice 5.

(b) State estimation at 20000 seconds.

FIGURE 2.4. Dynamic response of the system.

System partitioning. In this chapter, a first simple distribution of the rod model is made. Two subsystems are obtained partitioning the rod. Each partition has ten states and there are no overlapping states as shown in figure 2.5.

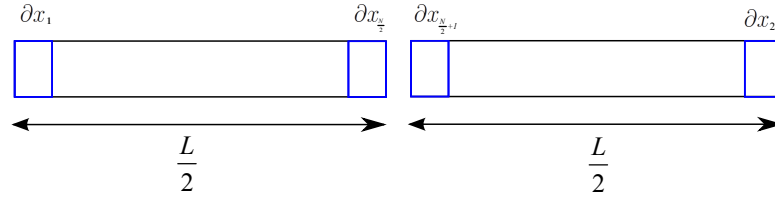


FIGURE 2.5. Rod partitioning into two subsystems.

The distributed state estimators will be dealing two reduced order models from the rod. Two nodes are obtained.

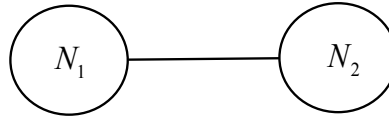


FIGURE 2.6. Nodal representation of two subsystems.

The neighbourhood N_i of node 1 comprises node 2 and itself, $N_1 = \{1, 2\}$. The same goes for node 2, $N_2 = \{1, 2\}$.

2.2. Distributed Kalman filters

A large number of distributed Kalman filters can be found in the literature. This is the most popular approach within the distributed state estimators schemes. Only three schemes were selected for implementation. The algorithms are explained in detail and some simulation results are presented.

2.2.1. Kalman filtering. The Kalman filter is a mathematical tool used as an estimator of the states of linear dynamical systems, from measurements corrupted with Gaussian white noise. Claimed to be one of the greatest estimation methods, it finds its most common applications in control theory, which needs to have knowledge of the states of a process, in order to know what it is doing, and then perform desired control actions over it [3, 48]. The Kalman filter addresses the problem of estimating the state or states of a system that is governed by a linear stochastic state equation and a measurement equation.

$$(2.27) \quad x(k) = Ax(k-1) + Bu(k-1) + w(k-1)$$

$$(2.28) \quad y(k) = Cx(k) + v(k)$$

where $w(k-1)$ is the process noise and $v(k)$ is the measurement noise and they are assumed to be Gaussian white noise. From now on, it is assumed that the systems behaviour obey to

dynamics such as modelled with equations 2.27 and 2.28.

2.2.1.1. *Centralized Kalman Filter.* The well known Kalman filter algorithm is presented in table 2.2 and implemented.

Algorithm: Centralized Kalman filter	
Initialize with	
$\hat{x}(0) = E(x(0))$	
$P(0) = E[(x - \hat{x})(x - \hat{x})^T]$	
<i>Prediction stage</i>	
(2.29)	$\hat{x}(k k-1) = A\hat{x}(k-1 k-1) + Bu(k-1)$ $\hat{z}(k) = C\hat{x}(k k-1)$ $P(k k-1) = AP(k-1 k-1)A^T + Q$
<i>Correction stage</i>	
(2.30)	$K(k) = P(k k-1)C^T[CP(k k-1)C^T - R]^{-1}$ $\hat{x}(k k) = \hat{x}(k k-1) + K(k)[z(k) - C\hat{x}(k k-1)]$ $P(k k) = [I - K(k)C]P(k k-1)$

TABLE 2.2. Classic Kalman filter algorithm.

And the estimated dynamic response of the system:

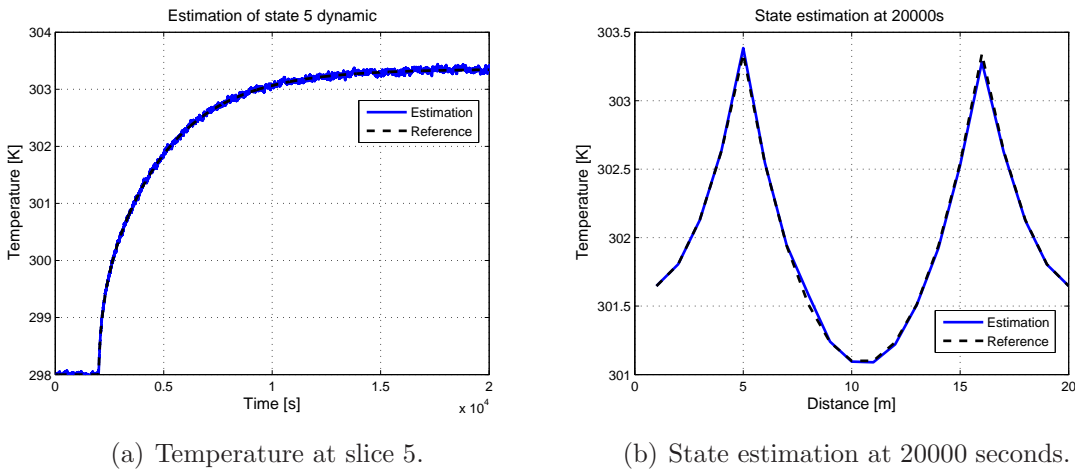


FIGURE 2.7. Dynamic response of the system with centralized Kalman filter.

The centralized approximation gives an accurate state estimation, which was expected. This performance will be the baseline of comparison for the distributed state estimation schemes.

2.2.2. Distributed and Decentralized Kalman filter. The distributed and decentralized Kalman filter (DDKF) was first presented in [41] and further explained in [42]. This algorithm can be classified as a partition based state estimator since it does not require that all nodes estimate the overall state vector.

Algorithm: Distributed and decentralized Kalman filter

Initialize with

$$\hat{x}_i(0) = E(x_i(0))$$

$$P_i(0) = E[(x_i - \hat{x}_i)(x_i - \hat{x}_i)^T]$$

Prediction stage

$$(2.31) \quad \begin{aligned} \hat{x}_i(k|k-1) &= A_i \hat{x}_i(k-1|k-1) + B_i u_i(k-1) \\ P_i(k|k-1) &= A_i P_i(k-1|k-1) A_i^T + Q_i \end{aligned}$$

Correction stage

- Local estimation

$$(2.32) \quad \begin{aligned} P_i(k|y_i(k)) &= [C_i^T R_i^{-1} C_i]^{-1} \\ i_i(y_i(k)) &= [C_i^T R_i^{-1}] y_i(k) \\ \hat{x}_i(k|y_i(k)) &= P_i(k|y_i(k)) i_i(y_i(k)) \end{aligned}$$

- Inter-nodal communication

$$(2.33) \quad \begin{aligned} P_i(k|y_j(k)) &= T_i [T_j^T P_j^{-1}(k|y_j(k)) T_j]^{-1} T_i^T \\ \hat{x}_i(k|y_j(k)) &= T_i T_j^+ \hat{x}_j(k|y_j(k)) \end{aligned}$$

- Assimilation

$$(2.34) \quad \begin{aligned} P_i(k|k) &= [P_i^{-1}(k|k-1) + \sum_{j=1}^N P_i^{-1}(k|y_j(k))]^{-1} \\ \hat{x}_i(k|k) &= P_i(k|k) [P_i^{-1}(k|k-1) \hat{x}_i(k|k-1) + \sum_{j=1}^N P_i^{-1}(k|y_j(k)) \hat{x}_i(k|y_j(k))] \end{aligned}$$

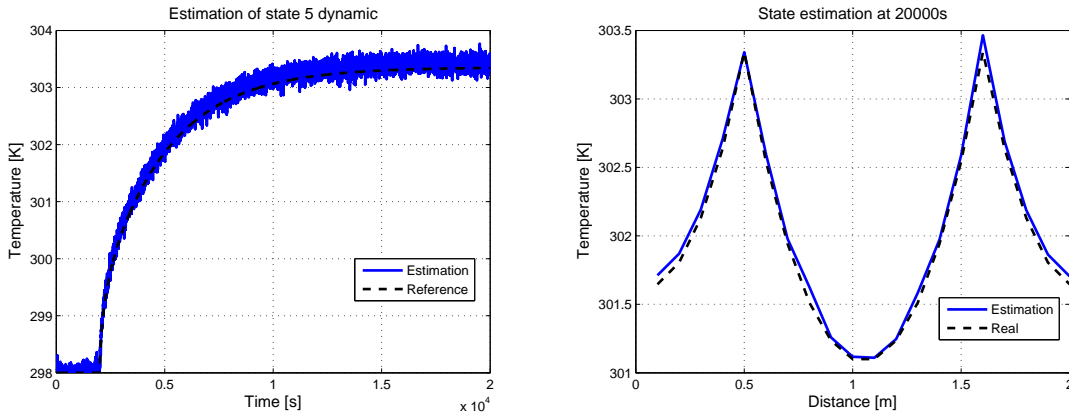
TABLE 2.3. Distributed and decentralized Kalman filter algorithm.

The prediction and correction stages of the centralized approach holds for the DDKF, but the prediction stage is performed locally by each node. The correction stage is divided into three steps. In the local update step, local covariance and state estimates are computed based on local measurements. In the internodal communication step, the information of

each node is communicated to relevant nodes (Neighbourhood) and transformed covariances and states estimates are obtained. Finally, in the assimilation step, the transformed states are assimilated locally to produce state and covariance estimates.

It is shown that the algorithm uses explicitly the transformation matrix T_i .

The estimated dynamic response of the system is:



(a) Temperature at slice 5.

(b) State estimation at 20000 seconds.

FIGURE 2.8. Dynamic response of the system with DDKF.

The reason this algorithm was chosen for implementation is that it was proposed as a solution for the distributed state estimation for large-scale systems within the aforementioned HD-MPC project. This algorithm was the start point and reference for the Kalman filter based distributed state estimators.

2.2.3. Distributed Kalman filter with state consensus. The distributed Kalman filter with state consensus (DKF-SC) was first presented in [29] and then improved in [49, 50]. This algorithm computes the overall system estimates at each node, and then performs a local state consensus by means of penalizing the difference of the system estimates of the neighbourhood using the correction factor ϵ .

Algorithm: Distributed Kalman filter with state consensus

Initialize with

$$\hat{x}_i(0) = E(x_i(0))$$

$$P_i(0) = E[(x_i - \hat{x}_i)(x_i - \hat{x}_i)^T]$$

Prediction stage

$$(2.35) \quad \begin{aligned} \hat{x}_i(k|k-1) &= A\hat{x}_i(k-1|k-1) + Bu_i(k-1) \\ P_i(k|k-1) &= AP_i(k-1|k-1)A^T + Q_i \end{aligned}$$

Correction stage

- Local update

$$(2.36) \quad l_i = C_i^T R_i^{-1} C_i, i_i = C_i^T R_i^{-1} y_i(k)$$

- Correction with neighbouring information

$$(2.37) \quad \begin{aligned} P_i^{-1}(k|k) &= P_i^{-1}(k|k-1) + \sum_{j \in N} l_j(k) \\ \hat{x}_i(k|k) &= P_i(k|k) [P_i^{-1}(k|k-1) \hat{x}_i(k|k-1) + \sum_{j \in N} i_j(k)] \end{aligned}$$

- Local consensus

$$(2.38) \quad \hat{x}_i(k|k) = \hat{x}_i(k|k) + \epsilon \sum_{j \in N} [\hat{x}_j(k|k) - \hat{x}_i(k|k)]$$

TABLE 2.4. Distributed Kalman filter with state consensus algorithm.

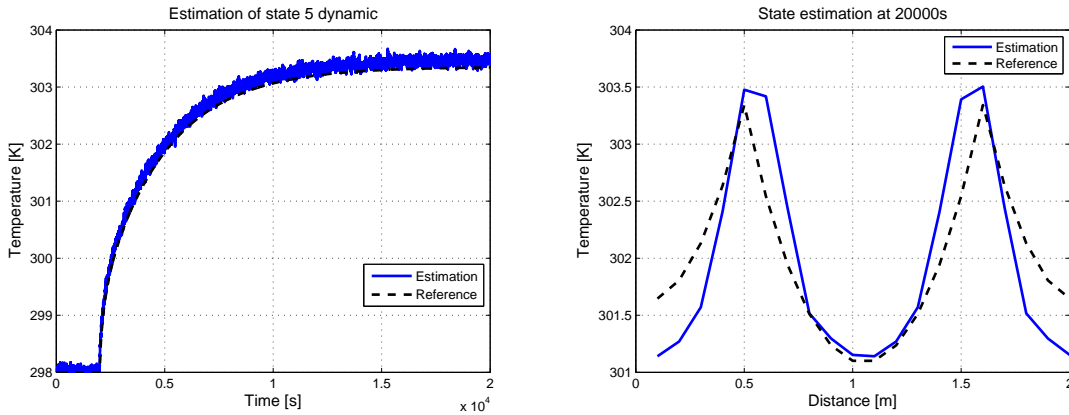
The steps presented in this algorithm are analogue to the ones presented in the DDKF algorithm in table 2.3. Local state and covariance are calculated and then corrected with neighbouring information. The final step, however, is different. In this case a local consensus of state estimation is reached using correction factor ϵ . If the value of ϵ increases, it could lead to instability of the algorithm, but if ϵ is zero, there is no consensus. Obtaining the right value for the correction factor is a tuning process.

In this case, the value chosen for ϵ was 0.1, since it is the value used in [46] where a rod was used as benchmark also.

One of the main drawbacks of this scheme is that it is necessary that all the nodes estimate the same amount of variables, otherwise, there are implementation issues. It can be seen in equation 2.37 that the covariances sum can not be performed if a neighbour node estimates

a different amount of states. In fact, the author assumes that all the measurement models are equivalent in its implementation example [50].

The estimated dynamic response of the system is:



(a) Temperature at slice 5.

(b) State estimation at 20000 seconds.

FIGURE 2.9. Dynamic response of the system with DKF-SC.

From the simulations results, another problem can be disguised. There are also implementation issues when the subsystems are not symmetric and a consensus is performed. For example, the heat source in subsystem 1 is located at its 5th partition, while the heat source in subsystem 2 is located at its 6th partition. The consensus procedure seems to replicate the heat source from one subsystem into the other.

2.2.4. Distributed Kalman filter with diffusion strategy. The distributed Kalman filter with diffusion strategy (DKF-DS) was presented in [51]. This algorithm computes the overall system estimates at each node, and then performs a local state consensus by means of weighting the state estimation performed by the nodes in the neighbourhood.

Algorithm: Distributed Kalman filter with diffusion strategy.

Initialize with

$$\hat{x}_i(0) = E(x_i(0))$$

$$P_i(0) = E[(x_i - \hat{x}_i)(x_i - \hat{x}_i)^T]$$

Prediction stage

$$(2.39) \quad \begin{aligned} \hat{x}_i(k|k-1) &= A\hat{x}_i(k-1|k-1) + Bu_i(k-1) \\ P_i(k|k-1) &= AP_i(k-1|k-1)A^T + Q_i \end{aligned}$$

Correction stage

- Incremental update

$$(2.40) \quad \begin{aligned} S_i(k) &= \sum_{j \in N_i} C_j^{-1} R_j^{-1} C_j \\ q_i(k) &= \sum_{j \in N_i} C_j^{-1} R_j^{-1} y_j(k) \end{aligned}$$

$$(2.41) \quad P_i^{-1}(k|k) = P_i^{-1}(k|k-1) + S_i(k)$$

$$(2.42) \quad \psi_i(k) = \hat{x}_i(k|k-1) + P_i(k|k)[q_i(k) - S_i(k)\hat{x}_i(k|k-1)]$$

- Diffusion update

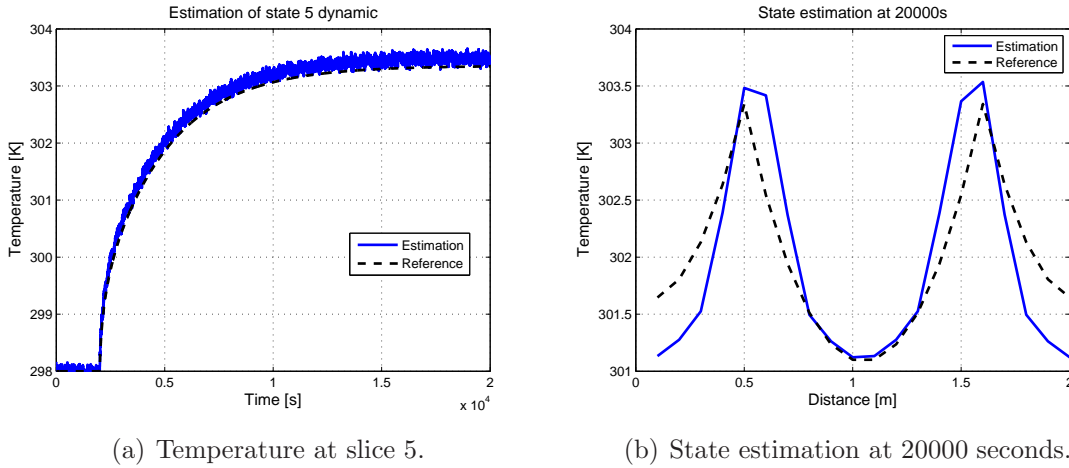
$$(2.43) \quad \hat{x}_i(k|k) = \sum_{j \in N_i} c_{j,i} \psi_i(k)$$

TABLE 2.5. Distributed Kalman filter with diffusion strategy algorithm (Information form).

The algorithm procedure does not differ much from the previous algorithms development. Local state and covariance are calculated and then corrected with neighbouring information. The consensus, however, does not penalize the difference between estimations, but weights the estimation provided by each neighbouring node. The consensus term $c_{j,i}$ must satisfy $\sum_{j \in N_i} c_{j,i} = 1$ and $c_{j,i} \geq 0 \forall i, j$.

The consensus proposed in equation 2.43 is a convex combination of the estimates of the neighbours, and the author claims it outperforms the one proposed in equation 2.38 of the DKF-SC scheme, since it is a mere averaging of the estimates.

The estimated dynamic response of the system is:



(a) Temperature at slice 5.

(b) State estimation at 20000 seconds.

FIGURE 2.10. Dynamic response of the system with DKF-DS.

This algorithm presents the same implementation issues exposed for DKF-SC in the previous section. From equation 2.41 it can be seen that if all the neighbouring subsystems do not measure the same amount of states, the matrices dimensions mismatch. From the simulations results, it can be seen that the consensus strategy also seems to replicate the heat source from one subsystem into the other. Then, the implementation and good performance of the algorithm is restricted.

2.3. Distributed particle filter

The conception of the particle filter as a completely nonlinear state estimator and its inherent computational burden [52], makes this approach not a likely candidate for addressing the large-scale system state estimation problem, even more when dealing with linear large-scale models. However, the literature reports several developments on the subject to be considered [28, 53, 54].

2.3.1. Particle filtering. The main idea behind particle filtering is to consider a set of particles or guesses instead of only one particle or expected value as in the initialization of the Kalman filter. The evolution of the states is modelled as a simple Markov process, specified by their state transition probabilities. Observations about states are modelled by their likelihood probabilities. The aim of the algorithm is to estimate the posterior probability density function (*pdf*). The posterior *pdf* is represented by a set of weighted samples or particles. Due to the usage of weighted samples to approximate *pdf*, particle filter is computationally expensive.

The performance of the filter when addressing nonlinear problems surpasses the extended Kalman filter and even the unscented Kalman filter. The price that must be paid for the high performance of the filter is an increased level of computational cost. That precisely is

the main obstacle to a more widespread use.

The algorithm for a general and centralized particle filter implementation is presented in table 2.6.

Algorithm: Centralized Particle filter

Initialization

Draw N particles $x^n(0)$ with $n = 1, \dots, N$ and the initial weight $w^n(0)$

$$(2.44) \quad w^n(0) = \frac{1}{N}$$

Importance sampling step

- Sample N particles $x^n(k)$ from $p(x(k)|x^n(k-1))$.
- Perform the time propagation step to obtain a priori particles $x^n(k)$ using 2.27.
- Update the importance weights

$$(2.45) \quad w^n(k) = w^n(k-1)p(y(k)|x^n(k))$$

- Normalize the important weights

$$(2.46) \quad w^n(k) = \frac{w^n(k)}{\sum_{n=1}^N w^n(k)}$$

Selection step

- Resample N particles $x^n(k)$ according to $w^n(k)$ to obtain a posteriori particles

TABLE 2.6. Particle filter algorithm.

The estimated dynamic response of the system is:

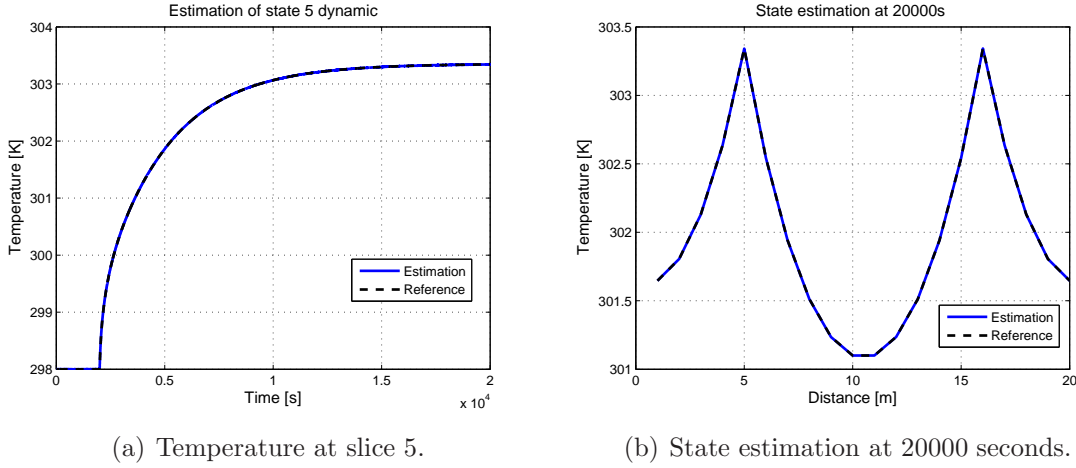


FIGURE 2.11. Dynamic response of the system with centralized Particle filter.

2.3.2. Distributed particle filter scheme. The proposed distributed particle filters schemes has mainly two approaches: the distributed nature is achieved by either transmitting local statistics of particles to a centralized unit (consensus based distributed particle filters) or using a message passing method or similar neighbouring communication methods [30]. Transmitting local statistics of particles to a centralized unit is not an efficient approach. Failure of the centralized unit is a treat to the entire network. In the message passing method, the algorithms construct or have defined a path through and between the nodes. Once the information runs through such paths, the important sampling and selection steps are performed locally.

In [31], a distributed particle filter for large-scale systems is presented that can be classified among the message passing methods. This scheme is selected for implementation due to its simplicity in the formulation and two important features: first, it considers low order dynamical systems. Second, it considers reduced amount of information exchange between neighbouring nodes. The way such features are obtained is by partitioning the system into subsystems modelled as in equation 1.7. The only exchange of information is the effect of relevant states from the neighbours.

The algorithm presented in table 2.6 holds for each subsystem, considering local variables $x_i^n(k)$ and $w_i^n(k)$ with state transition $p(x_i(k)|x_i^n(k-1))$ and likelihood $p(y_i(k)|x_i^n(k))$ distributions. And in the importance sampling step, the time propagation step to obtain a priori particles $x_i^n(k)$ is done using equation 1.7.

The estimated dynamic response of the system is:

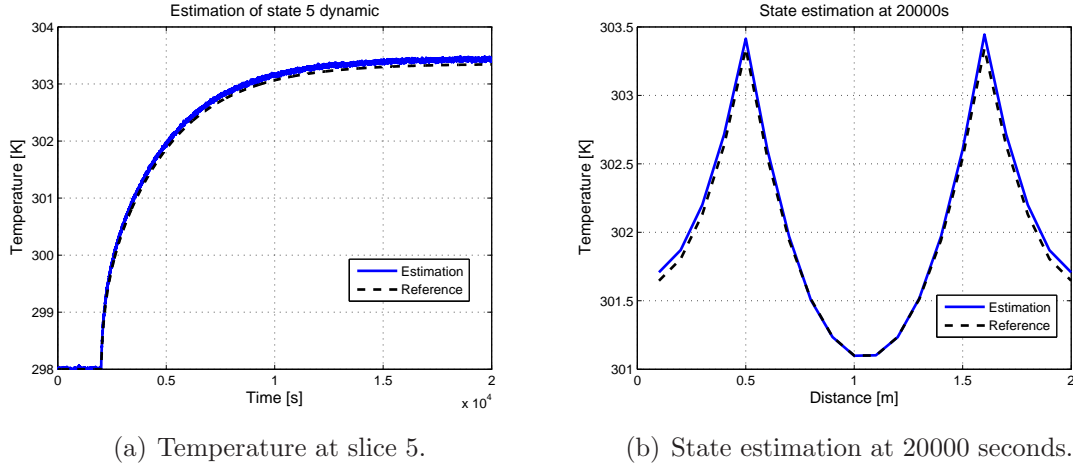


FIGURE 2.12. Dynamic response of the system with distributed Particle filter.

2.4. Distributed moving horizon estimator

Model predictive controllers popularity has been increasing in industrial applications [55] and its implementation has motivated the development of state observers such as the moving horizon estimator (MHE). Since the industrial processes complexity and size have been increasing, it is natural that large-scale approaches for both, MPC and MHE, have been proposed.

Distributed moving horizon estimation schemes is one of the most recent approaches for addressing the large-scale systems state estimation problem. It has been motivated for the development of novel MPC structures applied to large-scale systems. Among them, it is worth mentioning completely decentralized structures, distributed control systems and hierarchical structures.

2.4.1. Moving Horizon Estimation. Moving horizon estimation approach was formulated as an estimation strategy capable of considering inequality constraints of a process. The Kalman filter is the standard choice for estimating the state of a linear system when the measurements are noisy. However, if we add inequality constraints to the problem formulation, recursive solutions as the Kalman filter are unavailable. The strategy adopted to retrieve an optimal state estimate is to reformulate the estimation problem as a quadratic program. This formulation allows the addition of inequality constraints [56]. Unconstrained linear systems can be addressed with MHE as well.

Consider a system as the one presented in equations 2.27 and 2.28. The constrained linear state estimation problem is formulated as the solution of the following quadratic problem [57]:

$$(2.47) \quad \phi_t^* = \min_{x(0), \{w(k)\}_{k=0}^{t-1}} \phi_t$$

subject to the constraints $x(k) \in \mathbb{X}$, $w(k) \in \mathbb{W}$ and $v(k) \in \mathbb{V}$. The objective function is defined by ³

$$(2.48) \quad \phi_t(x(0), \{w(k)\}) = \sum_{k=0}^{t-1} (v^T(k)Rv(k) + w^T(k)Qw(k)) + (x(0) - \hat{x}(0))^T P(0)(x(0) - \hat{x}(0))$$

The pair $(\hat{x}(0), P(0))$ summarizes the prior information $t = 0$. This formulation is known as a full information estimator since all measurements are considered. The problem of solving the linear quadratic program grows without bound as more measurements are collected, leading to a very high computational burden. Moving horizon estimation solves this using a basic strategy, using a fixed amount of data, while approximately summarizing the old data. The size of the quadratic program is bounded since a moving, fixed-size estimation window is used.

The time interval is divided into two parts: $t_1 = \{0 \leq k \leq t - N - 1\}$ and $t_2 = \{t - N \leq k \leq t - 1\}$, where N is the amount of measurements to be considered or the window size.

The objective function of equation 2.48 is rearranged according to the previous division and the moving horizon estimation is the solution of the following quadratic problem:

$$(2.49) \quad \hat{\phi}_t^* = \min_{y, \{w(k)\}_{k=t-N}^{t-1}} \hat{\phi}_t(y, \{w(k)\})$$

subject to the constraints $x(k) \in \mathbb{X}$, $w(k) \in \mathbb{W}$ and $v(k) \in \mathbb{V}$. The objective function is defined as:

$$(2.50) \quad \hat{\phi}_t(y, \{w(k)\}) = \sum_{k=t-N}^{t-1} (v^T(k)Rv(k) + w^T(k)Qw(k)) + (y - \hat{x}(t-N))^T P(t-N)(y - \hat{x}(t-N))$$

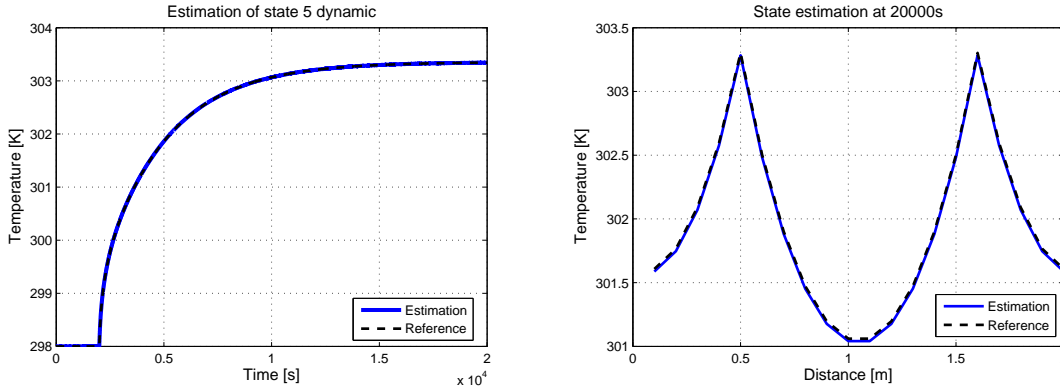
³There are many formulations for the objective function. Equation 2.48 is one example but each author can propose a variation according to which variable wishes to penalize.

Where the term $(y - \hat{x}(t - N))^T P(t - N)(y - \hat{x}(t - N))$ is called the arrival cost. This term approximately summarizes the old data from the interval $t_1 = \{0 \leq k \leq t - N - 1\}$, in other words, it summarizes the effect of data $\{y(k)\}_{k=0}^{t-N-1}$ on the state x_{t-N} .

The matrix $P(t)$ is calculated through the Kalman filter covariance update formula:

$$(2.51) \quad P(t) = Q + AP(t-1)A^T - AP(t-1)C^T(R + CP(t-1)C)^{-1}CP(t-1)A^T$$

The estimated response using the moving horizon approach is:



(a) Temperature at slice 5.

(b) State estimation at 20000 seconds.

FIGURE 2.13. Dynamic response of the system with moving horizon estimator.

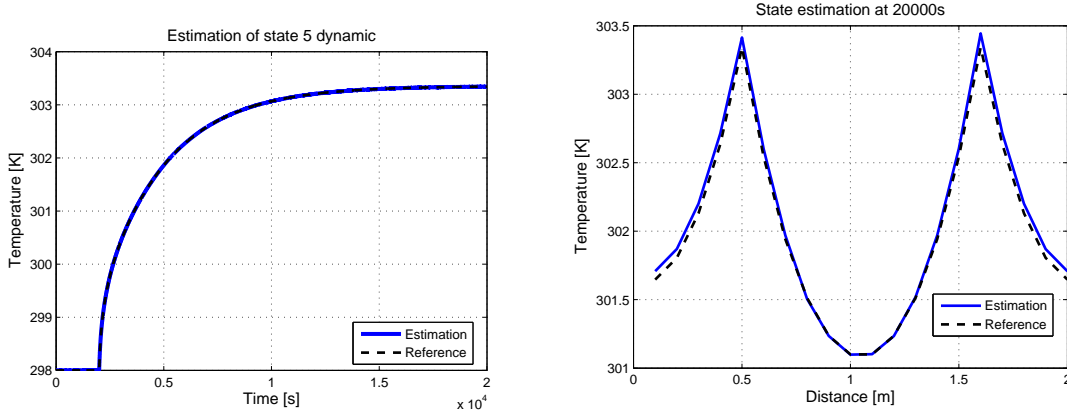
2.4.2. Distributed moving horizon estimation scheme. Distributed state estimation is proposed as a reliable state estimation solution when addressing the estimation of states of large-scale systems. Distributed MHE scheme was proposed and theoretically described in [18, 19]. The approach is simple: given a system partitioned into a number of subsystems, local sensors collect measurements that are modelled as in equation 1.6 and the local dynamics are modelled by expression 1.7. Then, each node solves a MHE problem to retrieve an optimal local state.

Local constraints are handled by each MHE, one of the main features of this approach, which is an extension of one of the earliest justifications for MHE development. Convergence of the approach is discussed and proved in [20, 21]. The first numerical examples are also presented.

For a given estimation horizon $N \geq 1$, each node $i \in N_l$ at time k solves a MHE problem. A local quadratic problem is described by:

$$(2.52) \quad \hat{\phi}_{t,i}^* = \min_{y_i, \{w_i(k)\}_{k=t-N}^{t-1}} \hat{\phi}_{t,i}(y_i, \{w_i(k)\})$$

The estimated dynamic response of the system for the distributed MHE is:



(a) Temperature at slice 5.

(b) State estimation at 20000 seconds.

FIGURE 2.14. Dynamic response of the system with distributed moving horizon estimator.

2.5. Other distributed state estimators

There are a considerable number of Kalman filter based distributed state estimators proposed in the literature, and only three have been presented and discussed. Next, there is a list of some extra approaches where some features are highlighted for future references.

Modified Decentralized Kalman Consensus (MDKC) [58]:

- Promise: Convergence to an unbiased estimate.
- Limitations: good results are obtained only when the addressed system can be treated as a strongly connected network.

Decentralized Kalman filter Decentralized Kalman filter (DKF) [11, 59]:

- Promise: Alleviate computational burden when all nodes exchange information among each other.
- Limitations: Communication increases and information can mislead the estimation. Estimation accuracy is diminished.

Decoupled Hierarchical Kalman filter (DHKF) [46]:

- Promise: Computational saving and improvement of the estimation accuracy respect DKF.
- Limitations: Poor performance if nodes can not transmit accurate information with each others, due to communication limitations or corruption.

Distributed Kalman filter with weighted averaging (DKF-WA) [46]:

- Promise: Only neighbouring nodes communicate, avoiding information corruption and decreased computational burden.
- Limitations: requires off-line calculations and is sensitive to non-linearities.

Distributed Kalman filter with bipartite fusion graphs (DKF-BF) [60]:

- Promise: Information loss decreased fusing the observations that are common among the local KF using bipartite fusion graphs and consensus averaging algorithms
- Limitations: There is the need of taking the inverse of high order matrices making the approach sensitive to ill-conditioned systems.

CHAPTER 3

Distributed state estimation techniques comparison

In the previous chapter, some distributed state estimation techniques were fully described and an implementation example was presented to show the estimated dynamic response for a system. It is of great interest to compare the performance of each technique when a system is partitioned into a different number of subsystems under the same conditions, including its sensors and actuators. Furthermore, if the comparisons are made by evaluating attributes of interest of an algorithm, as the mentioned computational cost, convergence and prediction error.

But first, it is proposed a modification of the distributed Kalman filter with state consensus and the distributed Kalman filter with diffusion strategy. It is shown that it can be proposed a unification of the distributed Kalman filter approach. The proposed unification is one of the main contributions of this thesis.

3.1. Distributed Kalman filters unification

Two main drawbacks were found when the distributed Kalman filter with state consensus and the distributed Kalman filter with diffusion strategy were implemented: first, the same number of variables must be measured at each subsystem, and second, the subsystems must be symmetric.

It is proposed the use of transformation matrix T_i to modify DKF-SC and DKF-DS algorithms in order to obtain distributed schemes able to address subsystems with different number of measured states or non symmetric. As exposed before, matrix T_i is used explicitly in the distributed and decentralized Kalman filter algorithm to choose the relevant states of a node, but it is also used to communicate information from one node to another as seen in

the internodal communication step in equation 2.33 of table 2.3.

3.1.1. Modified distributed Kalman filter with state consensus. The inclusion of matrix T_i is simple, as will be further shown. But first, the following equivalences are pointed out:

Replacing equation 2.33 in equation 2.36 yields:

$$(3.53) \quad \begin{aligned} l_i &= C_i^T R_i^{-1} C_i = P_i^{-1}(k|y_i(k)) \\ i_i &= C_i^T R_i^{-1} y_i(k) = i_i(y_i(k)) \end{aligned}$$

For the error covariance matrix estimate, from equations 2.37 and 3.53:

$$(3.54) \quad \sum_{j \in N} l_i(k) = \sum_{j \in N} P_i^{-1}(k|y_j(k))$$

Matrix T_i is used to transform the error covariance matrix in order to communicate information from one node to its neighbouring nodes:

$$(3.55) \quad P_i(k|y_j(k)) = T_i [T_j^T P_j^{-1}(k|y_j(k)) T_j]^{-1} T_i^T$$

which yields the following expression:

$$(3.56) \quad \begin{aligned} P_i^{-1}(k|k) &= P_i^{-1}(k|k-1) + \sum_{j \in N} P_i^{-1}(k|y_j(k)) \\ P_i(k|k) &= \left[P_i^{-1}(k|k-1) + \sum_{j \in N} P_i^{-1}(k|y_j(k)) \right]^{-1} \end{aligned}$$

It is the exact same expression as equation 2.34 in the assimilation step of distributed and decentralized Kalman filter.

For the state estimate, from equations 2.37 and 3.53:

$$(3.57) \quad \sum_{j \in N} i_j(k) = \sum_{j \in N} i_i(y_j(k))$$

Matrix T_i is used again as:

$$(3.58) \quad i_i(y_j(k)) = T_i T_j^+ i_j(y_j(k))$$

Now, the consensus procedure of equation 2.38 can be carried out by applying the state transformation presented in equation 2.33.

The dynamic response of the modified algorithm is:

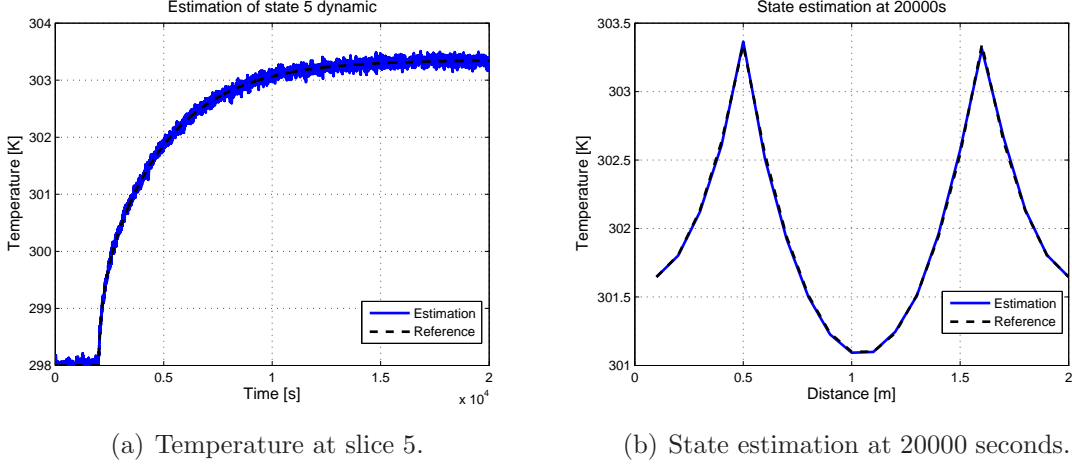


FIGURE 3.1. Dynamic response of the system with modified DKF-SC.

The results show an improvement of the dynamic response respect the DKF-SC proposed in [50]. Each node considers only one source of heat, correcting the undesired effect seen in figure 2.9.

3.1.2. Modified distributed Kalman filter with diffusion strategy. The modification procedure of the distributed Kalman filter with diffusion strategy is similar as the one presented for the distributed Kalman filter with state consensus. Replacing equation 2.33 in equation 2.40 yields:

$$\begin{aligned}
 (3.59) \quad S_i(k) &= \sum_{j \in N_i} C_j^{-1} R_j^{-1} C_j = \sum_{j \in N_i} P_j^{-1}(k | y_j(k)) \\
 q_i(k) &= \sum_{j \in N_i} C_j^{-1} R_j^{-1} y_j(k) = \sum_{j \in N_i} i_j(y_j(k))
 \end{aligned}$$

For the error covariance matrix estimate, matrix T_i is used to transform the error covariance matrix in order to communicate information from one node to its neighbouring nodes:

$$(3.60) \quad P_i(k | y_j(k)) = T_i [T_j^T P_j^{-1}(k | y_j(k)) T_j]^{-1} T_i^T$$

With this consideration, and replacing equation 3.59: in equation 2.40:

$$\begin{aligned}
 (3.61) \quad P_i^{-1}(k|k) &= P_i^{-1}(k|k-1) + S_i(k) \\
 P_i^{-1}(k|k) &= P_i^{-1}(k|k-1) + \sum_{j \in N} P_i^{-1}(k | y_j(k)) \\
 P_i(k|k) &= \left[P_i^{-1}(k|k-1) + \sum_{j \in N} P_i^{-1}(k | y_j(k)) \right]^{-1}
 \end{aligned}$$

Which is the exact same expression as equation 2.34 in the assimilation step of distributed and decentralized kalman filter.

For the state estimate, equation 2.42 holds to obtain a local estimate, the consensus procedure is performed with the next modification:

$$(3.62) \quad \hat{x}_i(k|k) = \psi_i(k) + \sum_{j \in N_i} c_{j,i} T_i T_j^+ \psi_j(k)$$

The dynamic response of the modified algorithm is:

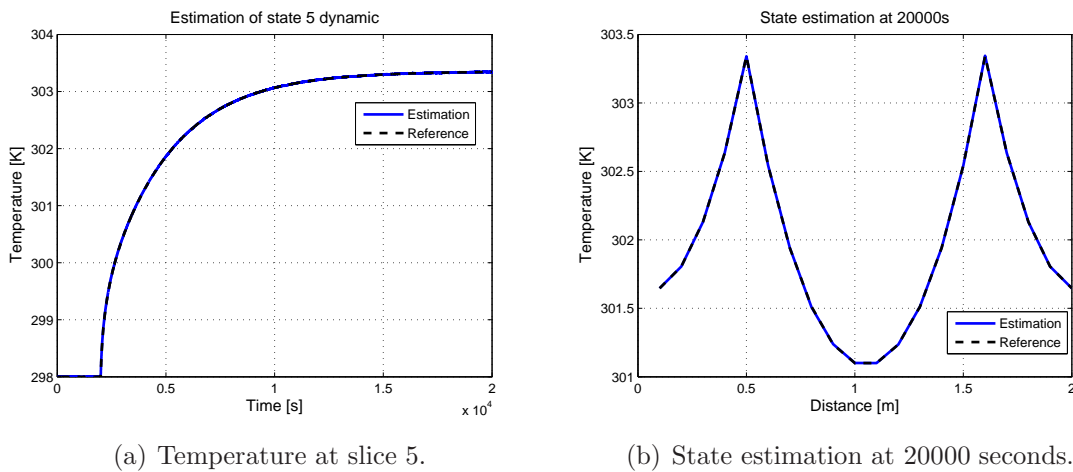


FIGURE 3.2. Dynamic response of the system with modified DKF-DS.

The results show an improvement of the dynamic response respect the DKF-DS proposed in [51]. Each node considers only one source of heat, correcting the undesired effect seen in figure 2.10.

3.2. Distributed state estimators implementations for different amount of nodes

It is a common practice within the literature that the comparison made for distributed state estimation techniques, especially Kalman filter based, the use of only one system partition. In this section, the rod system is partitioned into 3 and 4 subsystems, and the chosen distributed state estimation techniques are implemented accordingly to the description made in chapter 2 and the modified algorithms presented in section 3.1.

3.2.1. 3 nodes system partitioning. The system is partitioned into 3 subsystems. The first node comprises states 1 to 6, the second nod comprises states 7 to 14 and the third node comprises states form 15 to 20. The subsystems are not symmetric and the number of measured states are different according to the sensors placement shown in figure 2.2.

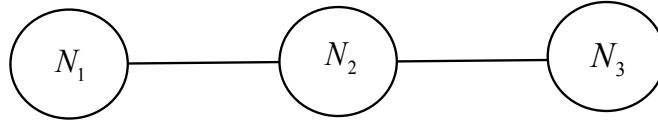
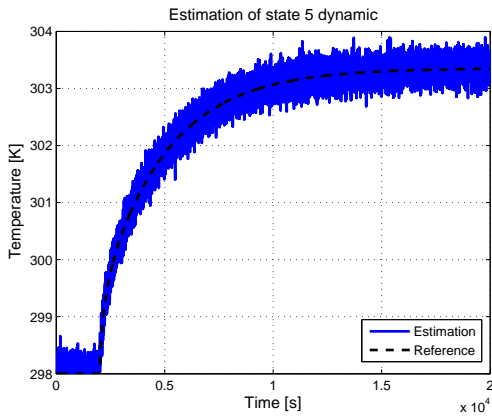


FIGURE 3.3. Nodal representation of 3 subsystems.

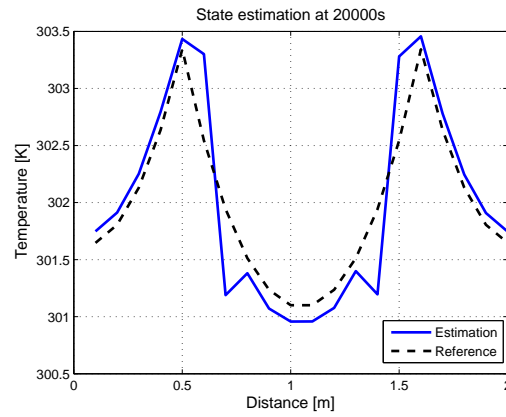
The neighbourhoods are defined as:

- $N_1 = \{N_1, N_2\}$
- $N_2 = \{N_1, N_2, N_3\}$
- $N_3 = \{N_2, N_3\}$

Following figures show the dynamic response of each distributed state estimation technique for a 3 nodes system partitioning:

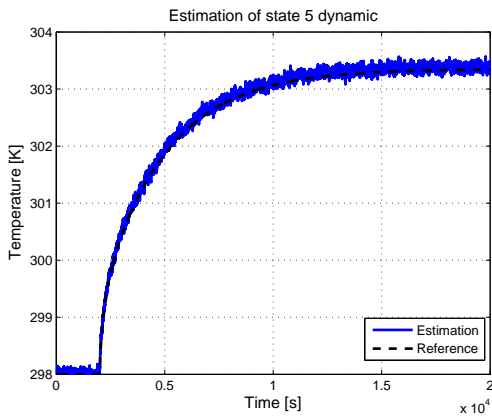


(a) Temperature at slice 5.

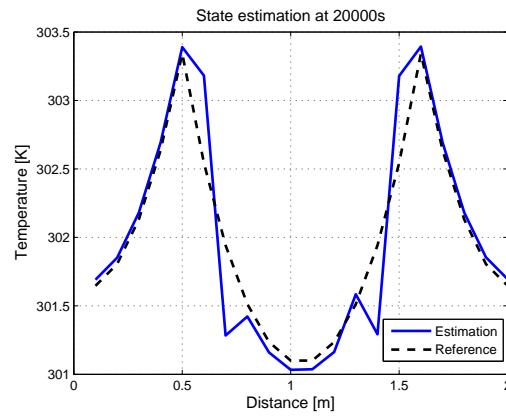


(b) State estimation at 20000 seconds.

FIGURE 3.4. DDKF estimation for 3 nodes distribution.

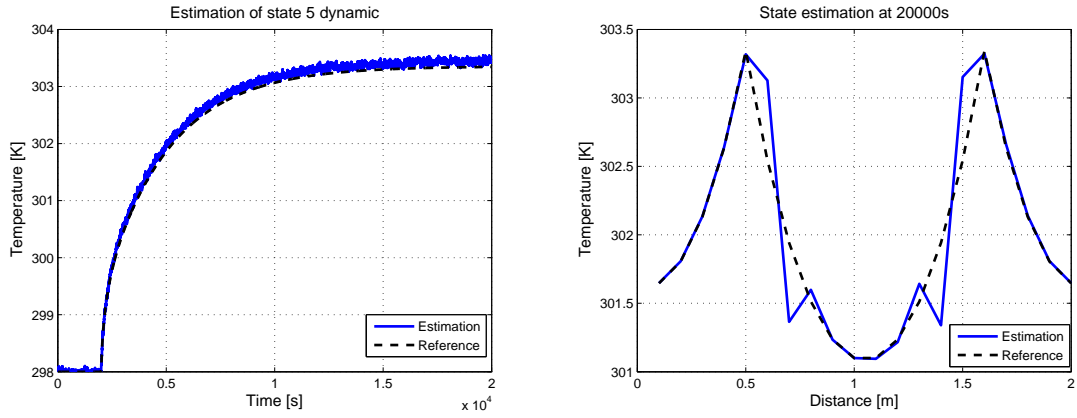


(a) Temperature at slice 5.



(b) State estimation at 20000 seconds.

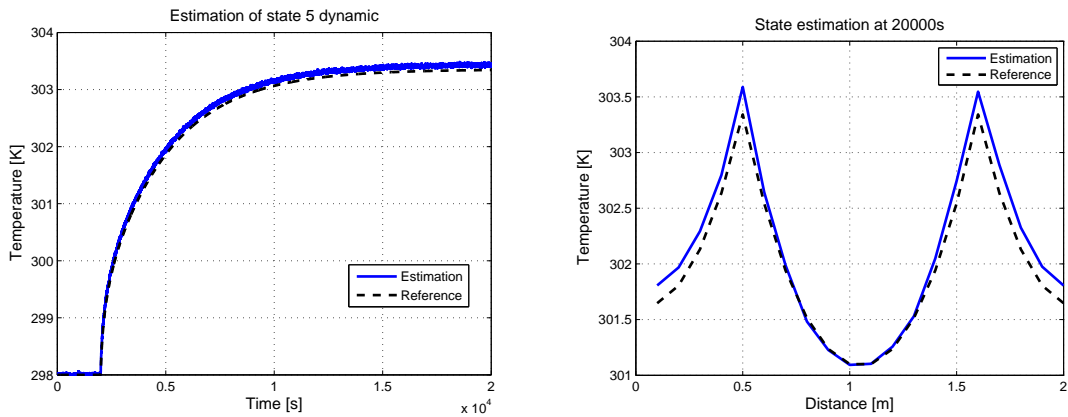
FIGURE 3.5. Modified DKF-SC estimation for 3 nodes distribution.



(a) Temperature at slice 5.

(b) State estimation at 20000 seconds.

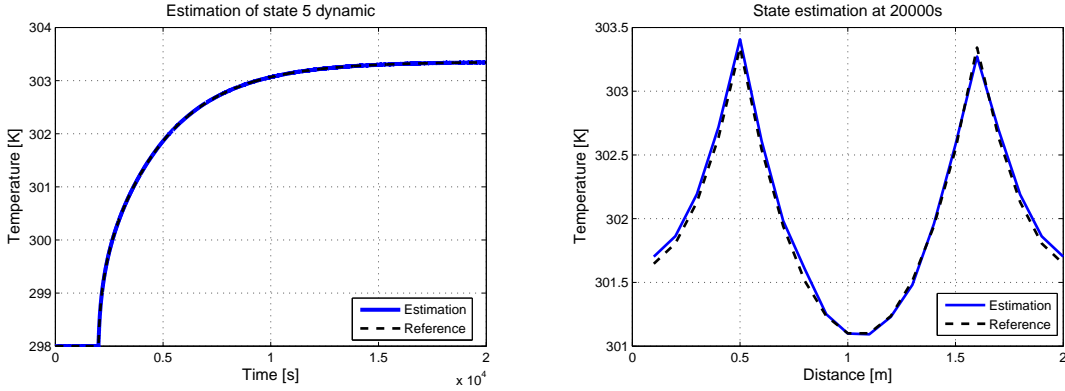
FIGURE 3.6. Modified DKF-DS estimation for 3 nodes distribution.



(a) Temperature at slice 5.

(b) State estimation at 20000 seconds.

FIGURE 3.7. Distributed Particle filter for 3 nodes distribution.



(a) Temperature at slice 5.

(b) State estimation at 20000 seconds.

FIGURE 3.8. Distributed moving horizon estimator for 3 nodes distribution.

3.2.2. 4 nodes system partitioning. The system is partitioned into 4 subsystems. Each subsystem has the same number of states, the first node comprises states 1 to 5, the second node comprises states from 6 to 10, the third node comprises states from 11 to 15 and the fourth node comprises states from 16 to 20.

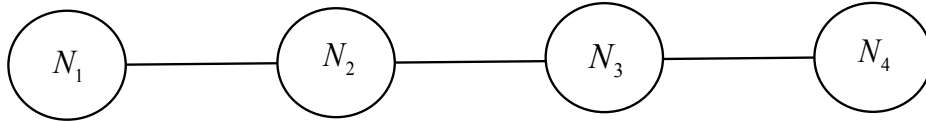
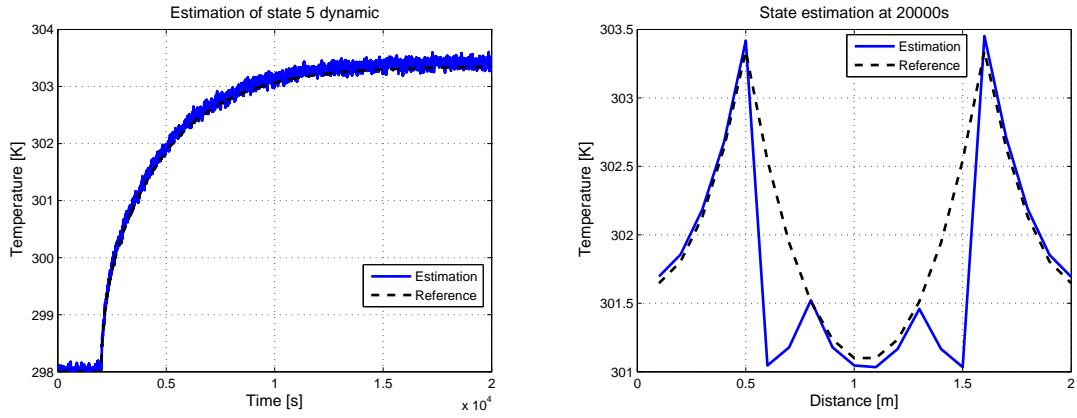


FIGURE 3.9. Nodal representation of 4 subsystems.

The neighbourhoods are defined as:

- $N_1 = \{N_1, N_2\}$
- $N_2 = \{N_1, N_2, N_3\}$
- $N_3 = \{N_2, N_3, N_4\}$
- $N_4 = \{N_3, N_4\}$

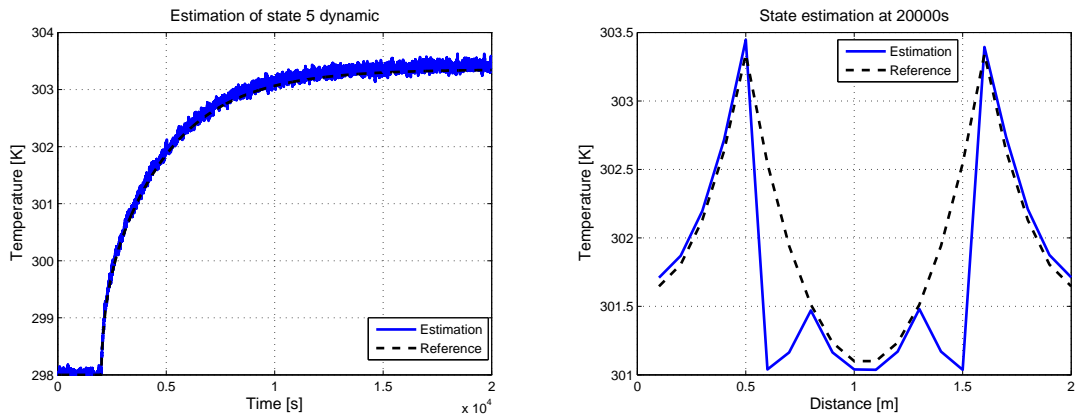
Following figures show the dynamic response of each distributed state estimation technique for a 4 nodes system partitioning:



(a) Temperature at slice 5.

(b) State estimation at 20000 seconds.

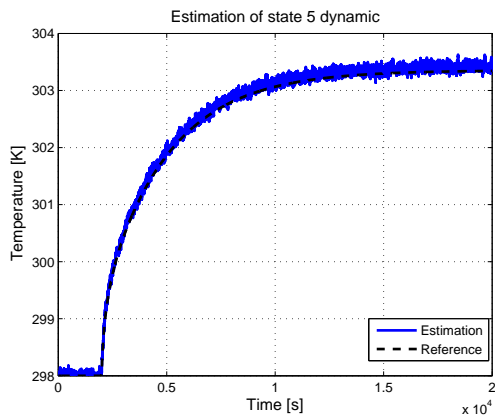
FIGURE 3.10. DDKF estimation for 4 nodes distribution.



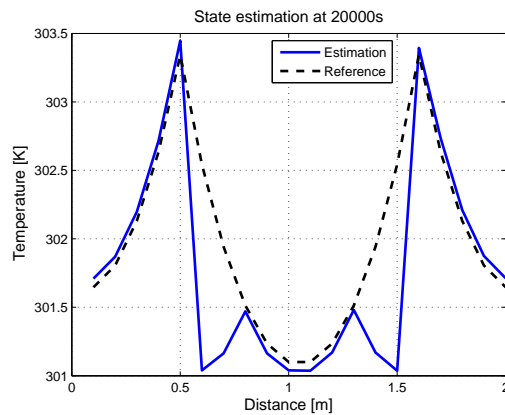
(a) Temperature at slice 5.

(b) State estimation at 20000 seconds.

FIGURE 3.11. Modified DKF-SC estimation for 4 nodes distribution.

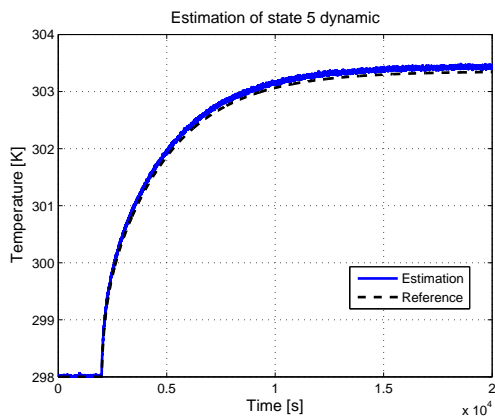


(a) Temperature at slice 5.

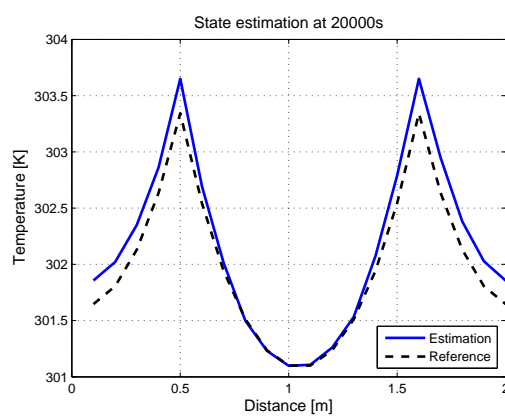


(b) State estimation at 20000 seconds.

FIGURE 3.12. Modified DKF-DS estimation for 4 nodes distribution.

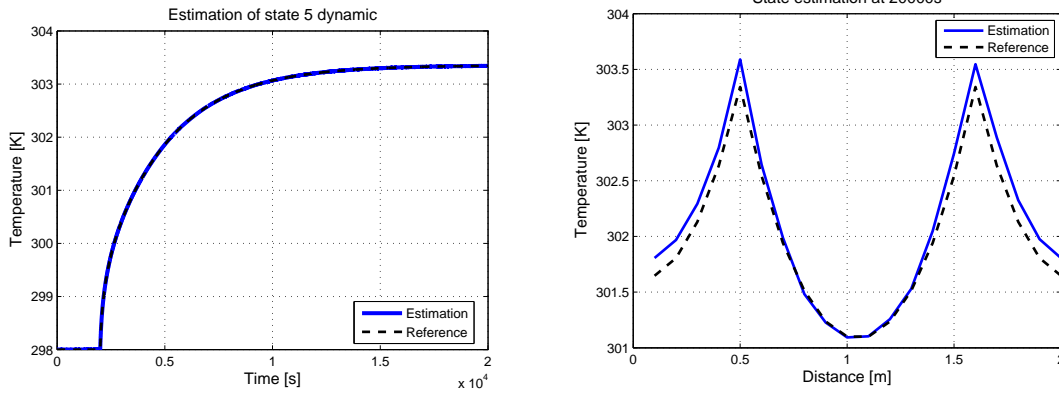


(a) Temperature at slice 5.



(b) State estimation at 20000 seconds.

FIGURE 3.13. Distributed Particle filter for 4 nodes distribution.



(a) Temperature at slice 5.

(b) State estimation at 20000 seconds.

FIGURE 3.14. Distributed moving horizon estimator for 4 nodes distribution.

3.3. Distributed state estimators performance analysis and comparison

Computational cost, convergence and prediction error are the attributes to be evaluated and compared for each distributed state estimation implementation. The computational cost is an attribute that must be evaluated, since the justification for a distributed approach to solve the control and state estimation of large-scale systems is the reduction of the system complexity and the scalability of the computational burden. Convergence must be guaranteed to obtain reliable local state estimations. The prediction error is a very common evaluation made when implementing state estimators and is a familiar metric from which performance analysis can be easily made.

The set of results obtained should give hints on which is the most suitable distributed state estimation scheme to be implemented for a large-scale system. It is necessary to define metrics that can give quantitative evaluation of the attributes.

3.3.1. Metrics for attributes evaluation.

Computational cost. The metric used to evaluate the computational cost is the amount of time the algorithm takes to execute. The evaluation of this attribute was made in a computer with the following characteristics:

- Intel Core i7 CPU U640 1.20GHz
- 8.00 GB RAM
- Windows operating system 64 bits

It must be considered that the distributed implementations are not executed by an according number of processors, but by only one computer. The time reported is the total time in which the estimation was achieved performing local estimators from one processor.

Convergence. The selected metric to evaluate the convergence of the distributed implementations is the trace norm of the error covariance matrix. Early works on the

evaluation of convergence of recursive state estimators used the trace norm as metric of convergence performance, such studies are reported in [61, 6]. In [62], an exhaustive study of convergence properties of distributed and decentralized Kalman filters schemes, was presented, it was stated that the trace norm was an adequate metric to check the approximate convergence of local state estimators. The trace norm of a matrix M is defined as:

$$(3.63) \quad \|P\| = \text{trace}\sqrt{P^T P}$$

If the trace norm of a local estimator approximates to the value that a centralized estimation gives, the approximated convergence of the local state estimator is guaranteed. This is not a definitive proof of convergence of a distributed state estimation technique. The global asymptotic convergence of local estimations is still an open problem [9].

Prediction error. The selected metric to evaluate the prediction error is the root-mean-square deviation (RMSD). It is a frequently used measure of the differences between values predicted an estimator and the actual values. It is useful to compare the estimated errors within a dataset, as in this case. The dataset includes the reference values given by the benchmark simulation. RMSD is defined as:

$$(3.64) \quad RMSD = \sqrt{\frac{\sum_{i=1}^n (x_i - \hat{x}_i)^2}{n}}$$

Distributed implementations works as the ones presented in [51, 50] have used RMSD as performance metric of the proposed distributed schemes based on the performance analysis procedures of [63, 64].

3.3.2. Attributes evaluation results. The results of the attributes evaluation are summarized in the following tables:

	Computational cost	Prediction error	Convergence
Centralized KF	18s	0.0250	0.7825
Centralized PF	824s	0.0098	-
Centralized MHE	136s	$1.0305 \cdot 10^{-9}$	0.0010

TABLE 3.1. Attributes evaluation for centralized estimators.

The centralized implementations results are the performance baseline to be taken as a reference in comparison for the distributed implementations. The best possible performance is achieved by the centralized implementation, since it has all the information. Particle filtering has an excessive amount of computational burden. It makes it a less interesting candidate to be implemented in a large-scale system, even more, if it can be represented by

a linear model, as in this case.

It is difficult to find a metric to evaluate the convergence of the particle filter, even more, a metric that can be compared to the trace norm metric. However, the high computational cost of the particle filter is highlighted as the attribute to be considered.

	Computational cost	Prediction error	Convergence
DDKF	34s	0.1243	1.5125
DKF-SC	91s	0.1339	38.4990
DKF-DS	58s	0.1500	25.3099
Modified DKF-SC	78s	0.0566	1.5432
Modified DKF-DS	76s	0.0285	1.5003
Distributed PF	1275s	0.0862	-
Distributed MHE	238s	$1.2854 \cdot 10^{-9}$	0.0014

TABLE 3.2. Attributes evaluation for 2 nodes system partitioning.

The time reported in table 3.2 to measure computational cost is the overall time in which the algorithm was executed. Prediction error and convergence are evaluated at subsystem 1, and following results are reported also for subsystem 1 despite the number of subsystems. Using the algorithm profiler tool of Matlab, it is possible to calculate the time of execution of each subsystem portion of algorithm.

	Computational cost
DDKF	17s
DKF-SC	46s
DKF-DS	23s
Modified DKF-SC	38s
Modified DKF-DS	36s
Distributed PF	683s
Distributed MHE	117

TABLE 3.3. Subsystem 1 computational cost evaluation for 2 nodes system partitioning.

Since the two subsystem partition gives 2 reduced order models almost equal, it was expected that the time of execution of each algorithm was close to the half of the overall algorithm execution. Notice that the times reported are lower than the reported for centralized implementation.

The modified DKF-SC and DKF-DS overcome the performance of the original implemented ones. The estimated dynamic response showed it, but it was necessary to quantify the real improvement. The main contribution of the modified algorithms are the reduction of the prediction error and improved the convergence of the algorithm. Also notice that the convergence evaluation gives as result similar values, almost the same, which was to be expected since it was demonstrated that the error covariance matrix calculation was the same as shown in equations 2.34, 3.56 and 3.61.

Distributed Kalman filter implementations are the most suitable solution for the large-scale state estimation problem. However, we must consider that these implementations does not consider constraints as the moving horizon estimation approach.

Next, evaluations for 3 subsystems and 4 subsystems are presented, it is expected that the results lead to the same observations just reported.

	Computational cost	Prediction error	Convergence
DDKF	79s	0.1597	46.4541
Modified DKF-SC	94s	0.1155	45.3480
Modified DKF-DS	98s	0.0654	45.2528
Distributed PF	1504s	0.0933	-
Distributed MHE	365s	$7.4205 \cdot 10^{-7}$	0.0094

TABLE 3.4. Attributes evaluation for 3 nodes system partitioning.

	Computational cost
DDKF	22s
Modified DKF-SC	30s
Modified DKF-DS	31s
Distributed PF	487s
Distributed MHE	132s

TABLE 3.5. Subsystem 1 computational cost evaluation for 3 nodes system partitioning.

Although the subsystems state vectors size diminished, the computational effort increased. The amount of information to be processed by each node is lower, but such information must be transmitted to other subsystems. In case of subsystem 1, it only exchanges information with subsystem 2. But subsystem 2 has to transmit information to subsystems 1 and 3, and process the information received from them. It is difficult to know if the information transmitted, for example, from subsystem 1 to subsystem 3, is relevant for the local state

estimation procedure of subsystem 3.

Taken the case of distributed and decentralized Kalman filter, assimilation step takes longer to be computed, since the calculation of error covariance matrix is larger due to the existence of more subsystems from which receive information, the sum size increases in equation 2.34.

	Computational cost	Prediction error	Convergence
DDKF	102s	0.1606	36.9345
Modified DKF-SC	118s	0.9024	36.8720
Modified DKF-DS	125s	0.0770	36.7981
Distributed PF	1712s	0.0983	-
Distributed MHE	538s	$2.007 \cdot 10^{-6}$	0.0202

TABLE 3.6. Attributes evaluation for 4 nodes system partitioning.

	Computational cost
DDKF	24s
Modified DKF-SC	28s
Modified DKF-DS	32s
Distributed PF	435s
Distributed MHE	135s

TABLE 3.7. Subsystem 1 computational cost evaluation for 4 nodes system partitioning.

In case of four subsystems, the amount of information that must be processed by each subsystem is even lower, but the communication between subsystems increases. However the difference between 3 and four partitions is not considerable. Subsystem 1 has six states for 3 subsystems partition, and five states for 4 subsystems partition.

CHAPTER 4

Simulation results

In this chapter, a large-scale system will be described and used as validation system for the analysis made of the distributed state estimation techniques. The rod benchmark was useful to implement and understand the distributed state estimation techniques, but the results obtained must be extended to a system with a size and complexity deserves the qualification of a large-scale system. If the attributes evaluation and performance of the algorithms resemble the performance of the large-scale system, such results could be concluding and can lead to the path of generalization in the, so far, wide and heuristic distributed state estimation area.

Next section presents, the large-scale system, Hydro-Power Valley (HPV) system is presented and described. The model of the dynamics governing the system behaviour are detailed and a partition into subsystems is also given.

4.1. Hydro Power Plant system description

Hydro-Power Plant system is used as a large-scale system benchmark to implement distributed control and state estimation systems [65]. The system overview is depicted in figure 4.1.

The elements that comprise the system are listed next:

- 3 lakes: L_1 , L_2 and L_3
- A river which is divided into 6 reaches R_1 to R_6 . The river is fed by the flows q_{in} and $q_{tributary}$
- Dams equipped with turbines at the end of each reach D_1 to D_6
- A duct U_1 connecting lakes 1 and 2
- Ducts equipped with turbines T_1 and T_2
- Ducts equipped with turbines and pumps C_1 and C_2

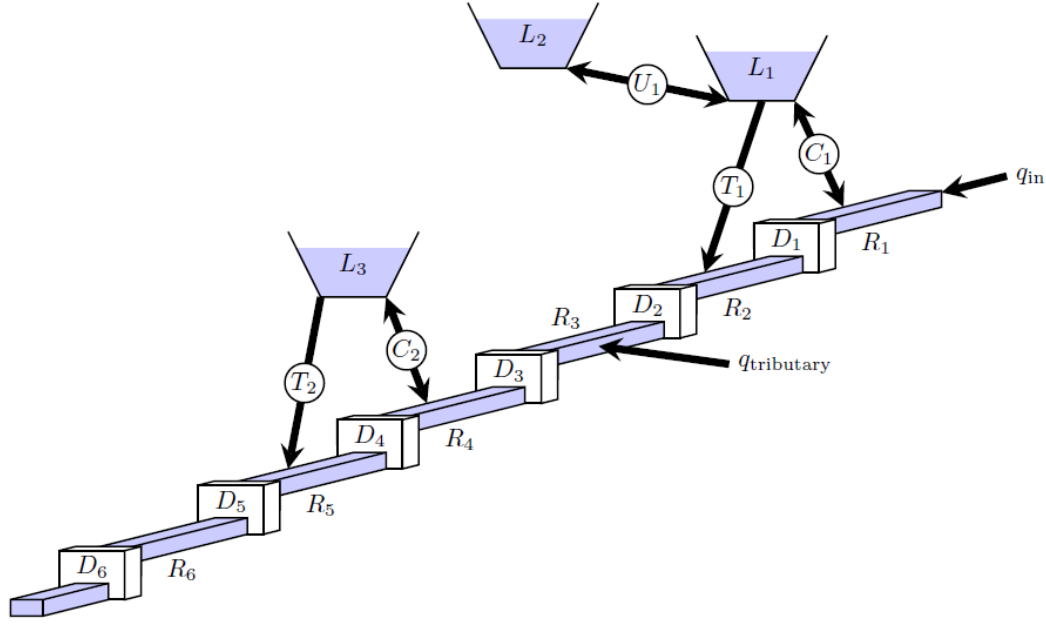


FIGURE 4.1. Hydro-Power Valley system overview [1].

The considerations made to simplify the system modelling are:

- The ducts are connected at the bottom of the lakes or to the bottom of the river bed.
- The cross section of the rivers and the lakes are rectangular.
- The width of the reaches varies linearly along them.
- The river slope is constant along every reach.

The main objective of the system is producing electricity using the turbines inside the dams and the ducts connecting the lakes and the reaches. The power produced by the turbines and consumed by the pumps is modelled. The power production and consumption are the controlled variables in this benchmark. The variables for state estimation are the water levels in the lakes and reaches and the water input and output flows in the reaches.

4.1.1.1. System modelling.

4.1.1.1.1. *Reach model.* The model of the reaches is based on the one-dimensional Saint Venant partial differential equations to express the mass and momentum balances:

$$(4.65) \quad \frac{\partial q(x, t)}{\partial x} + \frac{\partial s(x, t)}{\partial x} = 0$$

$$\frac{1}{g} \frac{\partial}{\partial t} \left(\frac{q(x, t)}{s(x, t)} \right) + \frac{1}{2g} \frac{\partial}{\partial x} \left(\frac{q^2(x, t)}{s^2(x, t)} \right) + \frac{\partial h(x, t)}{\partial x} + I_f(x, t) - I_0(x) = 0$$

where x is the spatial variable, which increases along the flow main direction, $q(x, t)$ is the river flow or discharge, $s(x, t)$ is the wetted surface, $h(x, t)$ is the water level with respect to the river bed, g is the gravitational acceleration, $I_f(x, t)$ is the friction slope and I_0 is the river bed slope which is assumed to be constant.

Since the cross section of the river is assumed to be rectangular, the following equations are considered:

$$(4.66) \quad s(x, t) = w(x)h(x, t)$$

and

$$(4.67) \quad I_f(x, t) = \frac{q^2(x, t) (w(x) + 2h(x, t))^{4/3}}{k_{srt}^2 (w(x)h(x, t))^{10/3}}$$

where $w(z)$ is the river width and k_{srt} is the Gauckler-Manning-Strickler coefficient ⁴.

Lateral inflows per space units $q_l(x)$ are considered:

$$(4.68) \quad \frac{\partial q(x, t)}{\partial x} + \frac{\partial s(x, t)}{\partial x} = q_l(x)$$

Discretized model. The partial differential equation 4.65 can be converted into an ordinary differential equation dividing the reach into sections as it was done with the rod in chapter 2. The reach is divided into N sections of length dx . $q_i(t)$ is the value of the discharge in the middle of the section i and $h_i(t)$ is the value of the water level at the beginning of section i . h_{N+1} corresponds to the water level at the end of the reach.

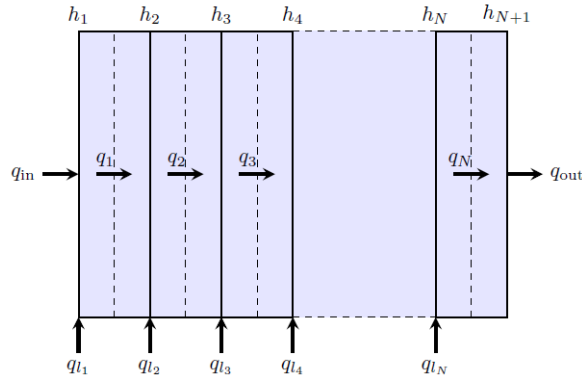


FIGURE 4.2. Reach spatial discretization.

⁴ k_{srt} coefficient value is assigned according the river bed surface and changes according to the river bed variations. In this case it is assumed is constant

$q_{in}(t)$ and $q_{out}(t)$ are the water input at the beginning of the reach and the water output at the end of the reach, respectively. The following set of differential equations are obtained:

$$(4.69) \quad \begin{aligned} \frac{\partial h_i}{\partial t} &= -\frac{1}{w_i} \frac{q_i - q_{i-1} - q_{li}}{dx} \\ \frac{\partial q_i}{\partial t} &= \frac{q_i}{w_i h_i} \frac{q_{li}}{dx} - \frac{2q_i}{w_i h_i} \frac{q_i - q_{i-1}}{dx} + \left[\frac{1}{w_i} \left(\frac{q_i}{h_i} \right)^2 - g w_i h_i \right] \frac{h_{i+1} - h_i}{dx} \\ &\quad + g w_i h_i I_0 - g w_i h_i \left[\frac{q_i (w_i + 2h_i)^{4/3}}{k_{srt}^2 (w_i h_i)^{10/3}} \right] \end{aligned}$$

and the water level at the end of the reach:

$$(4.70) \quad \frac{\partial h_{N+1}}{\partial t} = -\frac{1}{w_{N+1}} \frac{q_{out} - q_N}{dx}$$

where q_{li} is the total lateral inflow of section i .

4.1.1.2. *Lake model.* For each lake there are defined the variables water input flow $q_{in}(t)$ and water output flow $q_{out}(t)$. The volume of water inside the lake varies according to the following expression:

$$(4.71) \quad \frac{\partial v(t)}{\partial t} = q_{in}(t) - q_{out}(t)$$

But the cross section of the lake is assumed to be constant with lake surface area S and lake water level $h(t)$:

$$(4.72) \quad \frac{\partial h(t)}{\partial t} = \frac{q_{in}(t) - q_{out}(t)}{S}$$

4.1.1.3. *Duct model.* The flow inside the duct U_1 can be modelled using Bernoulli's law. Assuming that the duct section is much smaller than the lake surface, the flow from lake L_1 to lake L_2 can be expressed as:

$$(4.73) \quad q_{U_1}(t) = S_{U_1} \text{sign}(h_{L_2}(t) - h_{L_1}(t) + h_{U_1}(t)) \sqrt{2|h_{L_2}(t) - h_{L_1}(t) + h_{U_1}(t)|}$$

denoting $z = h_{L_2}(t) - h_{L_1}(t) + h_{U_1}(t)$, equation 4.73 can be written as $S_{U_1} \text{sign}(z) \sqrt{|z|}$. And the following approximation is considered so the function is differentiable for $z = 0$:

$$(4.74) \quad \text{sign}(z) \sqrt{|z|} \approx \frac{z}{(z^2 + \epsilon^4)^{1/4}}$$

For $\epsilon = 0$ the two functions are equivalent.

4.1.1.4. *Turbine model.* For each turbine, it is assumed that the discharge can be manipulated. The power produced is given by the following equation:

$$(4.75) \quad p_t(t) = k_t q_t(t) \Delta h_t(t)$$

Where k_t is the turbine coefficient, q_t is the turbine discharge and $\Delta h_t(t)$ is the turbine head.

4.1.1.5. *Pump model.* Pumps can be modelled similarly to turbines. The power absorbed by a pump is given by

$$(4.76) \quad p_p(t) = k_p q_p(t) \Delta h_p(t)$$

Equation 4.75 was applied to the pump.

4.1.1.6. *Modelling ducts C_1 and C_2 .* The ducts C_1 and C_2 are equipped with a turbine and a pump and therefore equations 4.75 and 4.76 to express the amount of power generated or absorbed. However, turbines and pumps do not operate simultaneously. A simplified model, called double flow model, is used to consider this system restriction. For example, for C_1 :

$$(4.77) \quad q_{C_1}(t) = q_{C_{1t}}(t) - q_{C_{1p}}(t)$$

and

$$(4.78) \quad p_{C_1}(t) = (k_{t_{C_1}} q_{C_{1t}}(t) k_{p_{C_1}} q_{C_{1p}}(t)) \Delta h_{C_1}(t)$$

4.2. Hydro-Power Plant system partition

The system is partitioned into 8 subsystems:

- Subsystem 1: comprised by L_1 , L_2 , U_1 , T_1 and C_1 .
- Subsystem 2: comprised by L_3 , T_2 and C_2 .
- Subsystem 3: comprised by R_1 and D_1
- Subsystem 4: comprised by R_2 and D_2
- Subsystem 5: comprised by R_3 and D_3
- Subsystem 6: comprised by R_4 and D_4
- Subsystem 7: comprised by R_5 and D_5
- Subsystem 8: comprised by R_6 and D_6

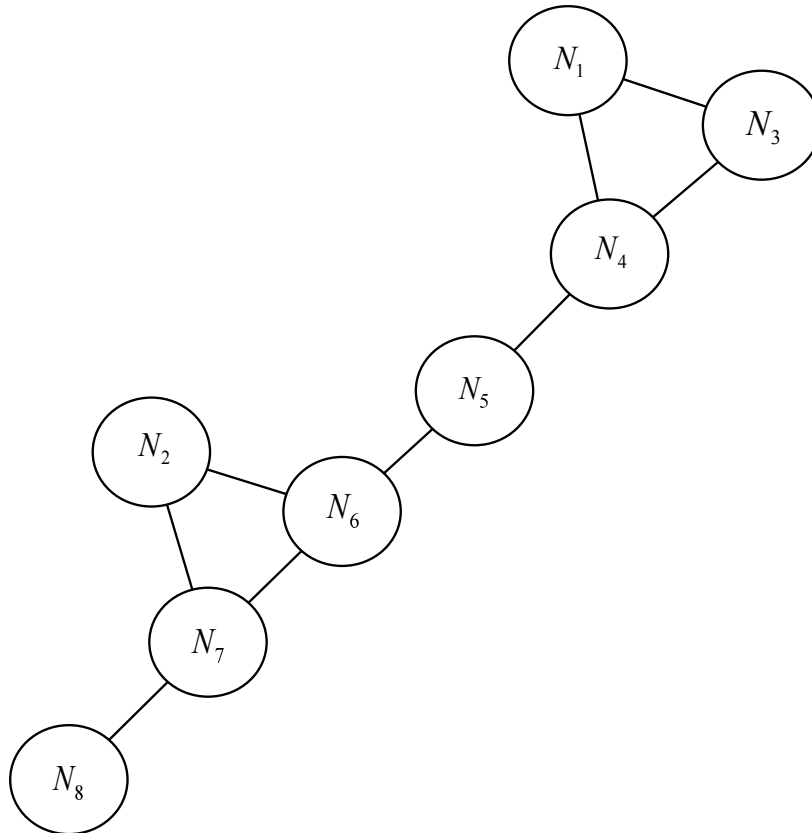


FIGURE 4.3. Hydro-Power Plant node connections.

The equations governing the subsystem behaviour can be derived using the equations described in the previous section.

4.2.0.7. *Subsystem 1.*

- Lake L_1 and lake L_2 respectively.

From equation 4.72:

$$(4.79) \quad \begin{aligned} q_{in}(t) &= q_{L_1}(t) + q_{U_1}(t) \\ q_{out}(t) &= q_{T_1}(t) + q_{C_1}(t) \end{aligned}$$

$$(4.80) \quad \begin{aligned} q_{in}(t) &= q_{L_1}(t) + q_{U_1}(t) \\ q_{out}(t) &= q_{T_1}(t) + q_{C_1}(t) \end{aligned}$$

- Turbine T_1 .

The turbine head:

$$(4.81) \quad \Delta h_t(t) = h_{T_1} + h_{L_1}(t) - h_{R_2, T_1}(t)$$

- Duct equipped with turbine and pump C_1 .

The duct head:

$$(4.82) \quad \Delta h_{C_1}(t) = h_{C_1} + h_{L_1}(t) - h_{R_1, C_1}(t)$$

The flows q and the heads h are constrained by minimum and maximum values.

4.2.0.8. *Subsystem 2.*

- Lake L_3 .

From equation 4.72:

$$(4.83) \quad \begin{aligned} q_{in}(t) &= q_{L_3}(t) + q_{C_{2p}}(t) \\ q_{out}(t) &= q_{T_2}(t) + q_{C_{2t}}(t) \end{aligned}$$

- Turbine T_2 .

The turbine head:

$$(4.84) \quad \Delta h_t(t) = h_{T_2} + h_{L_3}(t) - h_{R_5, T_2}(t)$$

- Duct equipped with turbine and pump C_2 .

The duct head:

$$(4.85) \quad \Delta h_{C_2}(t) = h_{C_2} + h_{L_3}(t) - h_{R_4, C_1}(t)$$

Subsystems 3, 4, 5, 6, 7 and 8. Subsystems 3 to 8 are comprised by a reach and a dam. Figure 4.4 depicts the structure of the dams. All the flow going through the dams is used by the turbine to produce electricity. The head of the turbines inside the dams can be expressed as difference of the water level before and after the dam.

The water levels at the end of each reach h_R and each dam discharge q_D , which go to the dams turbines, are constrained by minimum and maximum values.

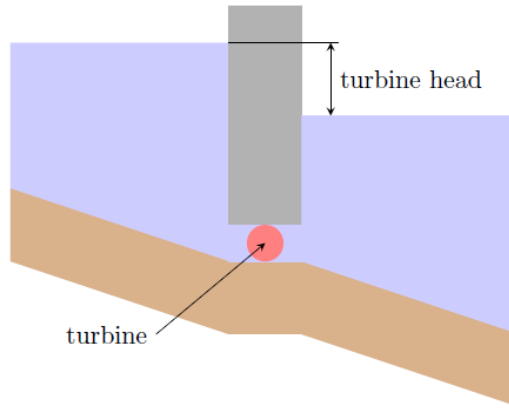


FIGURE 4.4. Dam disposition.

System variables. The system has 249 states. The first three states are the levels of the lakes, the next 246 states correspond to 20 flows and 21 water levels of the spatially distributed cells of each reach. The manipulated variables are the input flows of the the ducts and the reaches: $q_{T_1}, q_{T_2}, q_{C_{1t}}, q_{C_{1p}}, q_{C_{2t}}, q_{C_{2p}}, q_{R_1}, q_{R_2}, q_{R_3}, q_{R_4}, q_{R_5}$ and q_{R_6} . The measured variables are the lakes water levels, the water level at the end of the reaches, the power in the ducts (produced or consumed) and the power of the turbines of the dams: $h_{L_1}, h_{L_2}, h_{L_3}, h_{R_1}, h_{R_2}, h_{R_3}, h_{R_4}, h_{R_5}, h_{R_6}, P_{D_1}, P_{D_2}, P_{D_3}, P_{D_4}, P_{D_5}$ and P_{D_6} .

For the simulation of the system, it is considered the water discharge from the lakes and the nominal input flows of the ducts and the reaches. The dynamic response of some states of the plant is:

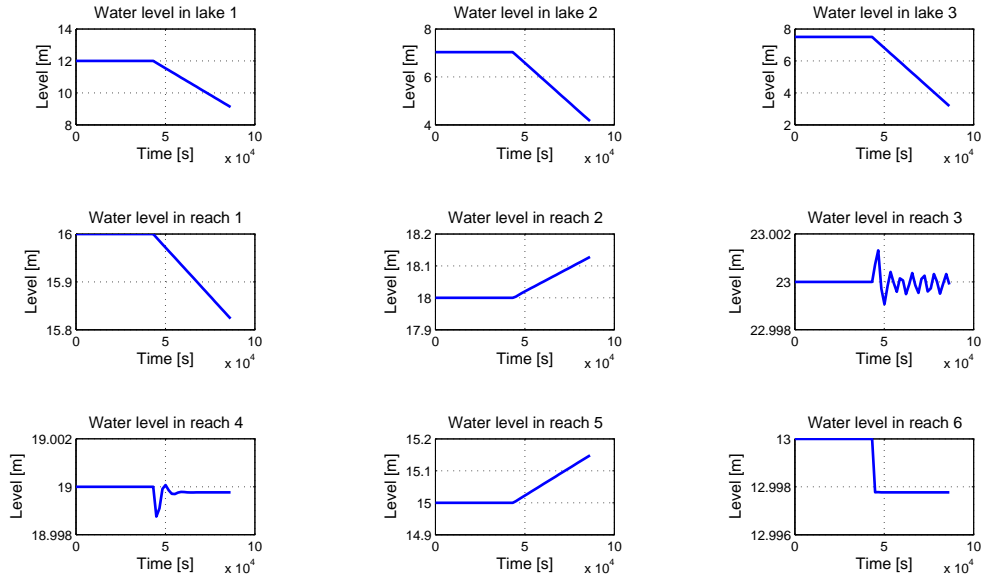


FIGURE 4.5. HPV dynamic response.

4.3. Distributed state estimation techniques simulations

In chapter 2, several distributed state estimation techniques were presented, explained and some qualitative estimation results were presented. In chapter 3, an attempt of improvement for DKF-SC and DKF-DS led not only to the possibility of using such algorithms when subsystems are not symmetric, but also it was found that all the distributed Kalman filter approach formulation is the same and the error covariance propagation is the same. The difference is made in the way the states consensus is made.

Since the modified DKF-DS, presented in section 3.1.2., proposed a convex combination of the states leading to a better convergence of the algorithm, it is chosen for implementation of the HPV. Also, due the high computational burden of the particle filter, it is not implemented for the large-scale system.

The state estimation of the system simulation is performed under the same conditions exposed in the previous section. Figure 4.6 presents the results of the estimation of the centralized Kalman filter and the modified DKF-DS from algorithm presented in equations 3.59 to 3.62. Notice that the modified DKF-DS uses a subsystem dynamical model given by equation 1.7.

Figure 4.7 presents the results of the estimation of the centralized moving horizon estimator and the distributed moving horizon estimator presented in section 2.4.2.

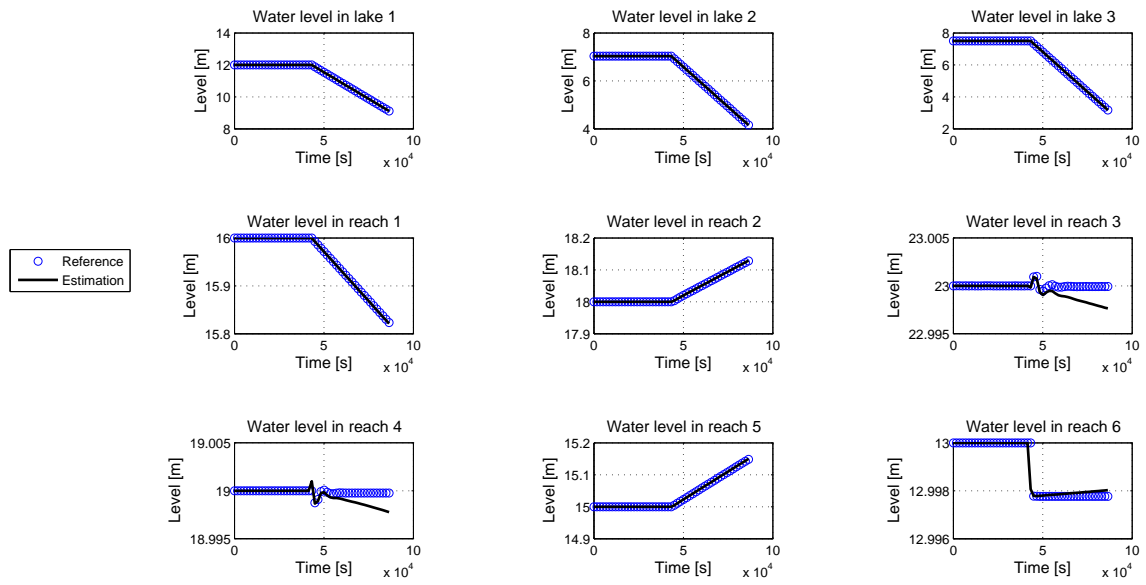


FIGURE 4.6. Kalman filter estimation.

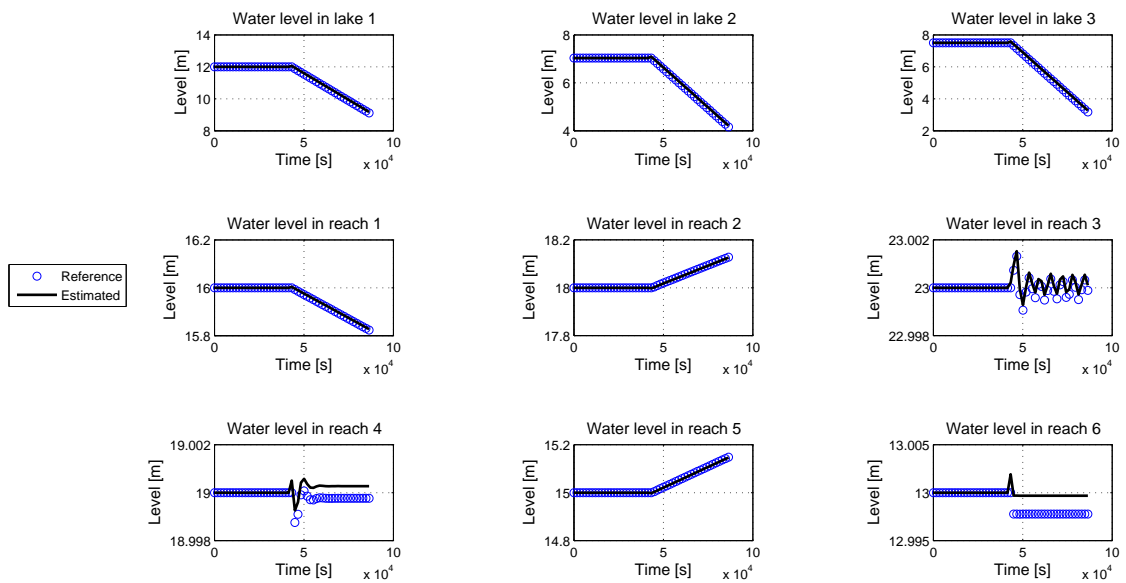


FIGURE 4.7. MHE estimation.

The response of the distributed state estimators is good. It can be seen that the estimation using distributed moving horizon approach can keep up with the slightly changes of the system dynamics. The centralized attributes evaluation are presented in next table. The values reported are the reference to check if the

	Computational cost	Prediction error	Convergence
KF	273s	$0.845 \cdot 10^{-4}$	1.2890
MHE	593s	$0.0414 \cdot 10^{-4}$	1.0589

TABLE 4.1. Attributes evaluation for centralized estimators in HPV system.

The attributes evaluation for subsystem 3 are presented in the next table.

	Computational cost	Prediction error	Convergence
DKF	273s	$9.6 \cdot 10^{-4}$	3.6340
DMHE	593s	$1.5414 \cdot 10^{-4}$	2.7850

TABLE 4.2. Attributes evaluation for distributed estimators in HPV system.

Two main results can be observed. First, the computational burden is alleviated for the subsystem. This observation was not obvious for the benchmark implementation, but for the large-scale system it takes relevance. Second, the prediction error values and convergence values are not significantly deviated from the centralized approach, which confirm that the distributed approach is a reliable solution for large-scale systems state estimation.

The distributed Kalman filter outperforms the distributed moving horizon approach in terms of computational cost. It does not imply that DKF should be always chosen over DMHE. For highly constrained systems, DMHE must be implemented. Prediction error and convergence values for DMHE indicate its estimation is more reliable, and the computational cost of a subsystem is still better than the observed for centralized approaches.

Conclusions and future work

Distributed state estimation techniques are a suitable solution for the state estimation problem for large scale systems. The computational cost is reduced at each processing unit within a subsystem.

The computational cost is the key attribute to be considered when evaluating distributed state estimation techniques. Any metric used for its evaluation, i.e. number of operations or time spent for the information to be processed, applies for all algorithms. Convergence was achieved since the distributed results approached the centralized ones. Prediction error was low in all cases. Besides, convergence and prediction error values did not show significant changes, except for the improvement made with the modified distributed Kalman filters.

It was achieved an unification of Kalman filter based distributed state estimation techniques. The way such techniques communicate information of the local uncertainty among nodes is similar. It indicates two areas in which the improvements can be made. First, a hardware improvement; the speed at which that information is transmitted must be increased and information loss must be lowered when the transmission takes place. Second, develop an optimal state consensus strategy. Distributed Kalman filter with diffusion strategy showed better performance because the consensus was made using a convex combination of the neighbouring estimates.

The distributed particle filtering approach proved to give good and reliable local state estimates since the prediction error was very low. But the computational cost is too high, making it not worth of implementation seeking good estimates. A large scale system with highly nonlinear dynamics, and therefore, non linear model representation could be suitable for implementation to check if the trade off between computational cost and prediction error is affordable.

Distributed Kalman filters and distributed moving horizon estimators are the state estimation techniques to be used for large scale systems. MHE approaches are becoming very popular within manufacturing and chemical processes, motivated by the ever increasing use of model predictive control structures, which have analogue procedures to be designed. Distributed MHE approach is the most recent distributed approach, it showed good performance and, due to its mentioned popularity, it calls for further development and improvement, to take full advantage of handling local constraints.

Future work

The distributed approach for large scale systems is an ill-conditioned problem that needs formalization. It calls for a system partitioning technique that can demonstrate that a subsystem is well conditioned in terms of observability or an indicator that says that the subsystem input and output information is enough to achieve a reliable state estimation. The literature does not propose many alternatives to approach the problem, but it is intended to address the problem as an optimization problem in which the relationship of the subsystems can be evaluated through generalized measure of association GMA statistical test, which is currently used to analyse the structural dependence of the human brain.

Formalization of the distributed Kalman filter modified algorithms is intended. As demonstrated, the formulation of three Kalman filter based distributed state estimators is exactly the same as using nodal matrix transformation, it is sought to formalize this procedure, extend it to other DKF approaches and propose optimal state consensus.

Bibliography

- [1] Damien Faille. Control specifications for hydro-power valleys. Technical report, Hierarchical and Distributed Model Predictive Control for Large-Scale Systems (HD-MPC), Electricite de France, September 2009.
- [2] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [3] F. Daum. Nonlinear filters: Beyond the kalman filter. *IEEE A&E systems magazine*, 20(8):57 – 69, August 2005.
- [4] D.M. Falcao, Felix F. Wu, and Liam Murphy. Parallel and distributed state estimation. *IEEE Transactions on Power Systems*, 10(2):724–730, 1995.
- [5] Yaoyu Wang and Yixin Yu. Distributed asynchronous state estimation algorithm for distributed energy management system (DEMS). *Zhongguo Dianji Gongcheng Xuebao/Proceedings of the Chinese Society of Electrical Engineering*, 16(3):160–164, 1996.
- [6] D.D. Siljak and M.B. Vukcevic. Large-scale systems: stability, complexity, reliability. *Journal of the Franklin Institute*, 301(12):49–69, January 1976.
- [7] A.N. Venkat, J.B. Rawlings, and S.J. Wright. A framework for integrating model predictive controllers to control large-scale systems. In *AICHE Annual Meeting, Conference Proceedings*, pages 6146–6148, 2005.
- [8] M.F. Godwin, S. Spry, and J.K. Hedrick. Distributed collaboration with limited communication using mission state estimates. In *Proceedings of the American Control Conference*, volume 2006, pages 2040–2046, 2006.
- [9] R. Scattolini. Architectures for distributed and hierarchical model predictive control ? a review. *Journal of Process Control*, 19(5):723 – 731, 2009.
- [10] A.N. Venkat, I.A. Hiskens, J.B. Rawlings, and S.J. Wright. Distributed output feedback mpc for power system control. In *Proceedings of the IEEE Conference on Decision and Control*, pages 4038–4045, 2006.
- [11] T.M. Berg and H.F. Durrant-Whyte. Model distribution in decentralized multi-sensor data fusion. In *Proceedings of the 1991 American Control Conference, June 26, 1991 - June 28, 1991*, volume 3 of *Proceedings of the American Control Conference*, pages 2292 – 2293, Boston, MA, USA, 1991. American Automatic Control Council.
- [12] K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Computer Networks.*, 43(3):499 – 518, 2003.
- [13] M. A Demetriou. Estimation of spatially distributed processes using mobile spatially distributed sensor network. *SIAM Journal on Control and Optimization*, 48(1):266 – 291, 2009. cited By (since 1996) 12.

-
- [14] R. Madhavan, K. Fregene, and L.E. Parker. Distributed heterogeneous outdoor multi-robot localization. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 1, pages 374–381, 2002.
- [15] T.M. Rasheed, L. Reynaud, and K. Al Agha. A stable clustering scheme for large scale mobile ad hoc networks. In *2005 IEEE/Sarnoff Symposium on Advances in Wired and Wireless Communication*, volume 2005, pages 121–124, 2005.
- [16] T.M. Rasheed, L. Reynaud, Y. Gourhant, and K.A. Agha. Reliability of weight metric based clustering models in large scale mobile ad hoc networks. In *Proceedings of the IASTED International Conference on WIRELESS SENSOR NETWORKS, Part of the Sixth IASTED International Multi-Conference on WIRELESS AND OPTICAL COMMUNICATIONS*, volume 2006, 2006.
- [17] S. Xu and R. Kumar. Distributed state estimation in discrete event systems. In *Proceedings of the American Control Conference*, pages 4735–4740, 2009.
- [18] M. Farina, G. Ferrari-Trecate, and R. Scattolini. A moving horizon scheme for distributed state estimation. In *Proceedings of the IEEE Conference on Decision and Control*, pages 1818 – 1823, 2009.
- [19] M. Farina, G. Ferrari-Trecate, and R. Scattolini. Distributed moving horizon estimation for linear constrained systems. *Automatic Control, IEEE Transactions on*, 55(11):2462 – 2475, November 2010.
- [20] Marcello Farina, Giancarlo Ferrari-Trecate, and Riccardo Scattolini. Moving-horizon partition-based state estimation of large-scale systems. *Automatica*, 46(5):910 – 918, 2010.
- [21] M. Farina, G. Ferrari-Trecate, and R. Scattolini. State estimation for large-scale partitioned systems: a moving horizon approach. In *American Control Conference (ACC), 2010*, pages 3180 – 3185. IEEE, July 2010.
- [22] J. Garca and J. Espinosa. Moving horizon estimators for large-scale systems. *Journal of Control Engineering and Applied Informatics*, 11(3):49–56, September 2009.
- [23] W. Lei and W. Chenglin. Distributed model predictive control based on cooperation. In *Chinese Control and Decision Conference, 2008, CCDC 2008*, pages 3526–3531, 2008.
- [24] K. Menighed, C. Aubrun, and J.-J. Yam. Distributed state estimation and model predictive control: Application to fault tolerant control. In *2009 IEEE International Conference on Control and Automation, ICCA 2009*, pages 936–941, 2009.
- [25] Anwer Bashi, Vesselin Jilkov, X. Rong Li, and Huimin Chen. Distributed implementations of particle filters. In *In Proc. Sixth International Conference of Information Fusion*, pages 1164 – 1171, 2003.
- [26] T. Misu, A. Matsui, M. Naemura, M. Fujii, and N. Yagi. Distributed particle filtering for multiocular soccer-ball tracking. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, volume 3, pages 937 – 940, 2007.
- [27] S.N. Nargess and A. Afshar. Consensus-based distributed particle filters in sensor networks. In *2009 Chinese Control and Decision Conference, CCDC 2009*, pages 4333–4338, 2009.
- [28] Mark Coates. Distributed particle filters for sensor networks. In *Proceedings of the 3rd international symposium on Information processing in sensor networks, IPSN '04*, pages 99 – 107, New York, NY, USA, 2004. ACM.
- [29] R. Olfati-Saber. Distributed kalman filter with embedded consensus filters. In *44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05*, pages 8179– 8184. IEEE, December 2005.
- [30] Dongbing Gu, Junxi Sun, Zhen Hu, and Hongzuo Li. Consensus based distributed particle filter in sensor networks. In *International Conference on Information and Automation, 2008. ICIA 2008*, pages 302 –307, June 2008.
- [31] A. Mohammadi and A. Asif. Distributed particle filtering for large scale dynamical systems. In *Multitopic Conference, 2009. INMIC 2009. IEEE 13th International*, pages 1 –5, December 2009.

- [32] L. Chai, B. Hu, and P. Jiang. Distributed state estimation based on quantized observations in a bandwidth constrained sensor network. In *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, pages 2411–2415, 2008.
- [33] R. Vadigepalli and F.J. Doyle III. Structural analysis of large-scale systems for distributed state estimation and control applications. *Control Engineering Practice*, 11(8):895–905, 2003.
- [34] G. Yang and C. Rong. Performance analysis of partition algorithms for parallel solution of nonlinear systems of equations. In *Parallel and Distributed Computing, Applications and Technologies, PDCAT Proceedings*, pages 684–687, 2003.
- [35] P. Morillo, S. Reuda, J.M. Orduna, and J. Duato. A latency-aware partitioning method for distributed virtual environment systems. *IEEE Transactions on Parallel and Distributed Systems*, 18(9):1215–1226, 2007.
- [36] Sigurd Skogestad and Ian Postlethwaite. *Multivariable Feedback Control: Analysis and Design*. Wiley, 1 edition, 1996.
- [37] A. Niederlinski. A heuristic approach to the design of linear multivariable interacting control systems. *Automatica*, 7(6):691–701, November 1971.
- [38] Kurt E Haeggblom. Partial relative gain: A new tool for control structure selection. *AICHE ANNUAL MEETING*, 1997.
- [39] Bjorn Wittenmark and Mario E. Salgado. Hankel-norm based interaction measure for input-output pairing. pages 1623–1623, July 2002.
- [40] Mario E. Salgado and Arthur Conley. MIMO interaction measure and controller structure selection. *International Journal of Control*, 77(4):367–383, 2004.
- [41] Arthur G.O. Mutambara. *Decentralized Estimation and Control for Multisensor Systems*. CRC Press, 1 edition, January 1998.
- [42] R. Vadigepalli and F. J. Doyle. A distributed state estimation and control algorithm for plantwide processes. *IEEE Transactions on Control Systems Technology*, 11(1):119 – 127, January 2003.
- [43] Dariusz Ucinski. *Optimal Measurement Methods for Distributed Parameter System Identification*. CRC Press, 1 edition, February 2009.
- [44] K.K. Somasundaram and J.S. Baras. Performance improvements in distributed estimation and fusion induced by a trusted core. In *2009 12th International Conference on Information Fusion, FUSION 2009*, pages 1942–1949, 2009.
- [45] S. Zheng, T. Jiang, and J.S. Baras. Robust state estimation under false data injection in distributed sensor networks. In *GLOBECOM - IEEE Global Telecommunications Conference*, 2010.
- [46] J. Sijs, M. Lazar, P. P.J van den Bosch, and Z. Papp. An overview of non-centralized kalman filters. In *IEEE International Conference on Control Applications, 2008. CCA 2008*, pages 739–744. IEEE, September 2008.
- [47] Christopher James Rapson. Spatially distributed control: Heat conduction in a rod. Master’s thesis, Lulea University of Technology, 2008.
- [48] M. Grewal and A. Andrews. *Kalman Filtering - Theory and Practice Using MATLAB*. John Wiley & Sons, Inc., New York, United States of America, third edition, 2008.
- [49] R. Olfati-Saber and J. S Shamma. Consensus filters for sensor networks and distributed sensor fusion. In *44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05*, pages 6698– 6703. IEEE, December 2005.
- [50] R. Olfati-Saber. Distributed kalman filtering for sensor networks. In *2007 46th IEEE Conference on Decision and Control*, pages 5492–5498. IEEE, December 2007.
- [51] Federico S Cattivelli, Cassio G Lopes, and Ali H Sayed. Diffusion strategies for distributed kalman filtering: formulation and performance analysis. IN *PROCEEDINGS OF THE IAPR WORKSHOP ON COGNITIVE INFORMATION PROCESSING*, 2008.
- [52] D. Simon. *Optimal State Estimation*. John Wiley and Sons, Inc., United States of America, 2006.

-
- [53] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188, February 2002.
- [54] M. Bolic, P.M. Djuric, and Sangjin Hong. Resampling algorithms and architectures for distributed particle filters. *IEEE Transactions on Signal Processing*, 53(7):2442–2450, July 2005.
- [55] S. Joe Qin and Thomas A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764, July 2003.
- [56] James B. Rawlings and Bhavik R. Bakshi. Particle filtering and moving horizon estimation. *Computers & Chemical Engineering*, 30(1012):1529–1541, September 2006.
- [57] Christopher V. Rao, James B. Rawlings, and Jay H. Lee. Constrained linear state estimation - a moving horizon approach. *Automatica*, 37(10):1619–1628, October 2001.
- [58] M. Alighanbari and J. P. How. An unbiased kalman consensus algorithm. In *American Control Conference, 2006*. IEEE, June 2006.
- [59] H. F Durrant-Whyte, B. Y.S Rao, and H. Hu. Toward a fully decentralized architecture for multi-sensor data fusion. In *, 1990 IEEE International Conference on Robotics and Automation, 1990. Proceedings*, pages 1331–1336 vol.2. IEEE, May 1990.
- [60] U. A Khan and J. M.F Moura. Distributing the kalman filter for Large-Scale systems. *IEEE Transactions on Signal Processing*, 56(10):4919–4935, October 2008.
- [61] L. Ljung. Asymptotic behavior of the extended kalman filter as a parameter estimator for linear systems. *IEEE Transactions on Automatic Control*, 24(1):36–50, February 1979.
- [62] M. Kamgarpour and C. Tomlin. Convergence properties of a decentralized kalman filter. In *47th IEEE Conference on Decision and Control, 2008. CDC 2008*, pages 3205–3210, December 2008.
- [63] L. Shi, K.H. Johansson, and R.M. Murray. Kalman filtering with uncertain process and measurement noise covariances with application to state estimation in sensor networks. In *Proceedings of the IEEE International Conference on Control Applications*, pages 1031–1036, 2007.
- [64] T. Damak. Procedure for asymptotic state and parameter estimation of nonlinear distributed parameter bioreactors. *Applied Mathematical Modelling*, 31(7):1293–1307, 2007. cited By (since 1996) 5.
- [65] C. Savorgnan and D. Moritz. Control benchmark of a hydro-power plant. Technical report, Katholieke Universiteit Leuven, 2011.