UNIVERSITÀ
DEGLI STUDI
FIRENZE

Università degli Studi di Firenze
Dipartimento di Ingegneria dell'Informazione (DINFO)
Corso di Dottorato in Ingegneria dell'Informazione
Curriculum: Automatica, Ottimizzazione e Sistemi Elettrici
SSD ING-INF/04

Dynamic Field Estimation in Complex Environments

Candidate
Nicola Forti

Supervisors
Prof. Luigi Chisci

Prof. Giorgio Battistelli

PhD Coordinator
Prof. Luigi Chisci

ciclo XXIX, 2013-2016

Università degli Studi di Firenze, Dipartimento di Ingegneria dell'Informazione (DINFO).

To Corrin and my family

Acknowledgments

First and foremost, I would sincerely like to thank my supervisors Prof. Luigi Chisci and Prof. Giorgio Battistelli for giving me the great opportunity of working with them and for their constant and relentless support, patience, encouragement and guidance during my 3-year doctorate. Their invaluable expertise and enthusiasm for research have made my Ph.D. education a very challenging yet rewarding experience. I truly appreciated the constant interactions, fruitful discussions, conversations and advice they gave me, and I am very grateful that I had the chance to go abroad for a whole year.

I would also like to thank Prof. Bruno Sinopoli who gave me the opportunity to visit Carnegie Mellon University, and thus living one of the best experiences in my life. While at CMU, I greatly profited from my interaction with him both academic and personal. His generosity, guidance and advice helped me to grow as a researcher, improve and broaden my research activity.

I have been fortunate to work with other wonderful collaborators. I am particularly thankful to Prof. Stefano Selleri and Prof. Giuseppe Pelosi for introducing me to the wonders of FEM, and Alfonso Farina for sharing with me his expertise and passion for research.

Thanks to all my lab-mates in Florence, especially Stefano for the numerous stimulating discussions, his innovative ideas and enthusiasm driving our collaborations, Claudio for helping me during my first steps in research, along with Daniela and Daniele.

Thanks to all the people I met at CMU, in particular Elias, Stefanos, Javad, Anit and Ana for many great discussions and all the good times we spent together. I also want to thank the research group of Prof. Sinopoli, particularly Sean for the beneficial interactions. I wish them all the best. A special thanks to Walter and Giovanni who shared a memorable experience with me in Pittsburgh.

My thanks to my parents, my sister and all my friends who supported, helped and inspired me during my studies will never be enough.

My final but most important thanks go to Corrin for sharing with me all the important moments.

Abstract

This work addresses fundamental challenges underlying dynamic field estimation, i.e. the problem of estimating a spatially distributed time-varying *field* of interest from noisy measurements collected by a wireless sensor network deployed over an area to be monitored. This is clearly an infinite-dimensional estimation problem, which is intrinsically *dynamic* since spatio-temporal (non steady-state) dynamics are explicitly taken into account. Most physical phenomena are inherently spatially distributed systems governed by partial differential equations (PDEs). Particular focus is on distributed estimation of time-evolving and space-dependent fields for which fully scalable (with respect to the spatial domain) filters are proposed by suitably adapting the parallel Schwarz domain decomposition method. The original infinite-dimensional filtering problem is approximated into a, possibly large-scale, problem of finite dimension, through the finite element (FE) method. Combining the aforementioned key ingredients, a novel distributed finite element Kalman filter (FE-KF) has been proposed and stability results have been provided. The presence of unknown sources (e.g. of heat, polluting agents, etc.) can pose difficulties in reconstructing the target field. To this end, the source estimation problem is considered, which consists of detecting and localizing a concentrated diffusive source as well as estimating its intensity and induced field. Two field estimation strategies which are *robust* with respect to the presence of unknown moving and, respectively, motionless sources have been designed, by recasting the source localization as a multiple-model filtering problem and by using the FE method for space-discretization of the resulting field dynamics. The concept of *source identifiability* has also been defined and system-theoretic conditions in terms of rank tests have been derived. Furthermore, the challenging problem of performing low-cost, *energy-efficient*, dynamic field estimation adopting binary sensor networks, and thus with a minimal amount of available information, is addressed. Relying on the so-called *noise-aided* paradigm, a novel optimization strategy based on a *Moving-Horizon* (MH) approximation of the *Maximum A-posteriori Probability* (MAP) estimation is presented. A final challenge addressed is introduced by unprecedented *security* issues potentially targeting next-generation monitoring systems, subject to malicious cyber and physical attacks. Attack-resilient strategies for *secure* dynamic field estimation have been formulated and solved following a stochastic Bayesian approach.

Contents

List of Figures

Chapter 1

Introduction

1.1 The problem of dynamic field estimation

Recent technological advances in wireless sensor networks (WSNs) have enabled the deployment of a large number of low-cost networked sensors for real-time monitoring of spatially distributed physical processes over an area of interest. Spatially distributed systems are processes governed by *partial differential equations* (PDEs) modelling various real-world physical systems. Rapid, accurate and reliable estimation of such spatially-varying phenomena evolving over time is of paramount importance in a multitude of monitoring/control application domains.

Typical examples include, but are not limited to: i) weather analysis and prediction [2] that usually requires the solution of a very large data-fitting problem involving a set of partial differential equations that models the evolution of the atmosphere; ii) environmental monitoring (e.g., monitoring of pollutants [3], of wildfires in forests [4], of volcanic eruptions via seismic activity [5], forecasting of the triggering and propagation of landslides [6]); iii) oceanography [7]; iv) *smart* (energy-efficient) buildings [8] equipped with heating, ventilation and air conditioning (HVAC) control and occupancy estimation systems; v) smart grids [9] with partial differential equations representing the temperature distribution evolution of thermostatically controlled loads (TCLs); vi) traffic monitoring [10]; vii) water flow regulation [11]; viii) structural health monitoring [12]; ix) adaptive optics [13].

Based on the above practical motivations, the aim of this dissertation is to address dynamic field estimation, i.e. the problem of estimating a spa-

tially distributed time-varying *field* of interest (a physical quantity such as temperature, concentration, pressure) from noisy measurements collected by a wireless sensor network. This is clearly an infinite-dimensional estimation problem, which is intrinsically *dynamic* since non steady-state spatio-temporal dynamics are explicitly taken into account.

The problem of state estimation of systems distributed over a large geographical region has been extensively studied in the finite-dimensional literature, in the context of linear and nonlinear state estimation of large-scale systems [14–20]. Such systems are possibly (but not necessarily) originated from spatial discretization of PDEs. Hence, rather than considering the infinite-dimensional *field*, these works are only interested in a finite (possibly high-dimensional) collection of field samples, so that a preceding spatial discretization is only implicitly assumed. Indeed, by standard discretization of the PDE operator, the field of interest can usually be represented as a combination of basis function coefficients [17, 20] leading to a lumped-parameter system. In [21], the spatially distributed process are modeled as random fields, estimated via Kriging interpolation so that the mean function of the Gaussian process is a combination of basis functions. In other works, numerical methods, such as the Godunov scheme for traffic estimation [22] and the explicit Lax diffusive scheme for flow estimation [23], have been used for the approximation of the field.

For the very large non-linear systems arising in the environmental sciences such as weather forecasting and oceanography, characterized by multi-scale and usually unstable and/or chaotic dynamics, many traditional state-estimation techniques are not practicable and *data assimilation* schemes have been proposed [24], [25]. To deal with the huge dimensionality of the resulting state vector, appropriate *reduced-order* filtering techniques with lower computational load have been suitably developed [26].

Significantly less effort has been devoted to the more challenging case of distributed-parameter systems, with only few works dealing with *field* monitoring over sensor networks. Interesting contributions have focused on extending the design of centralized/distributed filters for systems modelling spatially distributed processes from the finite-dimensional literature (see e.g., [27], [28]). In this case, the design of state estimators/observers is usually carried out in an abstract infinite-dimensional framework with no interest on the finite-dimensional approximation that is, however, crucial to practical filters' implementation.

1.2 Field monitoring in complex environments

With the advent of novel wireless sensor network technologies and solutions, unprecedented fundamental challenges will be inevitably introduced in modern and next-generation monitoring systems for dynamic field estimation. In order to enhance the performance of such systems, in this work the complexity of the environment is considered by taking into account the following challenges:

- **scalability:** due to their spatially distributed nature and their very complex dynamics, it is typically more convenient (but clearly more challenging) to undertake a *decentralized* approach for state estimation of distributed-parameter systems, which allows for *scalability* of computation with respect to the problem size (spatial domain). This can be achieved by decomposing the original system into smaller subsystems which can be monitored locally within a restricted region of competence.

- **robustness:** the task of dynamic field estimation becomes much more challenging in the presence of unknown sources altering the field. This is the reason why it is relevant to design estimation strategies which are *robust* with respect to the presence of unknown inputs affecting the spatially distributed system under monitoring. Robust field estimators will be therefore capable of detecting and localizing the unknown source as well as estimate its intensity and the source-induced field.

- **energy-efficiency:** energy consumption is one of the core issues in WSNs, especially in applications involving numerous geographically dispersed sensors with limited power and hence communication resources. The most challenging solution to tackle the energy consumption problem is to adopt binary sensor networks that transmit binary measurements conveying a minimal amount (i.e. a single bit) of information. The greater energy-efficiency unavoidably translates into novel difficulties on how to fully exploit the minimum information content available by means of smart estimation methods.

- **security:** the breakthrough of cyber-physical systems (CPSs) is transforming field estimators into complex systems integrating computation capabilities and physical processes, tightly connected by a communication infrastructure. The increased interactions between cyber and

physical realms pose novel *security* issues on such systems employed in homeland security, situation awareness, environmental and industrial monitoring. Thus, there is a need for *secure* state estimation strategies that account for the new vulnerabilities introduced, e.g. malicious attacks on sensors, communication channels, and the possible presence of malicious sources in the field of interest.

The complexity of such problems offers fundamentally new challenges that led to the work of this dissertation, addressing the problem of dynamic field estimation.

1.3 Organization

The rest of the thesis is organized as follows:

Chapter 2 introduces the basic concepts related to partial differential equations modeling spatially distributed systems. Special attention is devoted to second-order parabolic PDEs for which the notions of initial and boundary conditions are presented, and a prototypical example concerning the heat equation is described. Furthermore, the weak or variational formulation for different types of parabolic problems is derived.

Chapter 3 reviews the fundamental principles of the finite-element approximation of spatially distributed systems. In particular, the Galerkin method is introduced, which approximates the weak form of a PDE in a subspace of finite dimension. Moreover, the implementation of the finite-element approach is presented following a standard step-by-step procedure. This includes the domain discretization into finite elements, the selection of the approximating subspace, and the derivation of the local (element) properties, that are subsequently assembled to model the overall spatio-temporal behavior of the approximated system.

Chapter 4 addresses the problem of centralized and distributed dynamic field estimation. By exploiting the finite-element approximation, it is shown how it is possible to design a centralized finite element Kalman filter for spatially distributed systems. Further, we illustrate how such a filter can be extended to the distributed setting by means of the parallel Schwarz domain decomposition method, and we analyze the numerical stability in terms of boundedness and convergence of the discretization errors. Finally, results on the exponential stability of the distributed finite element Kalman filter

are provided, while a numerical example related to the estimation of a bi-dimensional temperature field demonstrates its effectiveness.

Chapter 5 introduces the problem of dynamic field estimation in the presence of an unknown source altering the field of interest. After formulating the source estimation problem, a finite-element approximation of the original infinite-dimensional source diffusion model is derived. Moreover, the notion of source identifiability, i.e. the possibility of detecting the source and uniquely identifying its location and intensity, is analyzed in a system-theoretic framework. Finally, a multiple-model Kalman filtering approach to source estimation is presented, and its effectiveness is demonstrated by means of a numerical example concerning the transport of a contaminant in a fluid.

Chapter 6 introduces the problem of dynamic estimation of a diffusion field from binary pointwise-in-space-and-time field measurements. First, state estimation with binary measurements is formulated as a Maximum A-posteriori Probability (MAP) problem. Then, a *moving-horizon* (MH) approximation of MAP estimation, referred to as MH-MAP algorithm, is presented and the properties of the resulting optimization problem are analyzed. Finally, the proposed approach is formulated for the special case of dynamic field estimation, for which simulation results are presented in a diffusive field case-study.

Chapter 7 addresses the problem of dynamic field estimation in adversarial environments for next-generation monitoring systems potentially subject to cyber and physical attacks. The considered system and attack models are introduced and the necessary background is provided. Next, the joint attack detection and mode-state estimation problem is formulated and solved in the Bayesian framework. A possible Gaussian-mixture implementation of the proposed joint attack detector and mode-state estimator is described, while the effectiveness of the novel approach is demonstrated via a simulation example concerning a power network.

Chapter 8 summarizes the contributions of the thesis and discusses avenues for future research.

Chapter 2

Spatially distributed systems

Most physical systems are intrinsically *spatially distributed*, and for many of them, this distributed nature can be approximately modelled in terms of *partial differential equations* (PDEs). The purpose of this chapter is to recall the basic concepts related to *partial differential equations*. In particular, the aim is to briefly survey PDEs and their classification (Section 2.1.1), with a special focus on *partial differential equations* involving time (the so-called *evolution* equations), characterized by a solution that evolves in time from a given initial configuration. Second-order *parabolic* equations will be introduced in Section 2.1.2 and initial-boundary value problems (IBVPs) for this class of PDEs will be illustrated, by considering the different models of boundary and initial conditions discussed in Section 2.1.3. A special example of physical interest (the *heat equation*) will be described in Section 2.1.4. Finally, the weak or variational formulation for such parabolic IBVPs will be presented in Section 2.2.

2.1 Mathematical model: partial differential equations

Spatially distributed systems are modeled as infinite-dimensional systems, governed by *partial differential equations* (PDEs). A PDE is an equation involving an unknown function, its partial derivatives, and the (multiple) independent variables. The unknown function might represent quantities such as temperature, electrostatic potential, value of a financial security,

concentration of a substance, velocity of a fluid, displacement of an elastic material, population density of a biological species, acoustic pressure, etc. Typically these quantities depend on many variables, and one is interested in understanding the dependency of the unknown quantity on these variables. A partial differential equation can be usually derived from physical laws and/or modeling assumptions that specify the relationship between the unknown quantity and the variables on which it depends. In particular, we will consider equations in which one independent variable represents the time variable $t \in \mathbb{R}_+$, while the remaining variables $\mathfrak{p}_1, \ldots, \mathfrak{p}_d, d = 1, 2, 3, \ldots$ represent spatial variables. The spatial coordinate vector is denoted by $\mathbf{p} \in \mathbb{R}^d$. In this case, a PDE becomes an equation involving derivatives of the unknown function $x : \Omega \times \mathbb{R}_+ \to \mathbb{R}$, where Ω is an open subset of $\mathbb{R}^d$, taking the following form

$$\mathcal{F}(x, \theta) = \mathcal{F}\left(\mathbf{p}, t, x, \frac{\partial x}{\partial t}, \frac{\partial x}{\partial \mathbf{p}}, \frac{\partial^2 x}{\partial t^2}, \frac{\partial^2 x}{\partial \mathbf{p}^2}, ...; \theta\right) = 0, \tag{2.1}$$

where θ denotes a vector of parameters on which the equation depends. The *order* of a PDE is the degree of the highest order derivatives appearing in the equation. Note that equations of order higher than fourth rarely occur, and the most important PDEs are the second-order ones. Moreover, a PDE is said to be *linear* if (2.1) depends linearly on the unknown x and on its derivatives. In the special case where the derivatives having maximal order only appear linearly (with coefficients which may depend on lower-order derivatives), the equation is said to be *quasi-linear*. It is said to be *semi-linear* when it is quasi-linear and the coefficients of the maximal order derivatives only depend on x and t, and not on the solution $\overline{x}$. A function $\overline{x} = \overline{x}(\mathbf{p}, t)$ is a solution of (2.1) if, substituting $\overline{x}$ and its derivatives in (2.1), one obtains $\mathcal{F}(\overline{x}, \theta) = 0$. Common examples of linear equations are the Laplace equation, the Poisson equation, the heat equation, Dupire's equation, Black-Scholes equation, the wave equation, and the transport equation. The reaction-diffusion equation is semi-linear. Burgers' equation is a quasi-linear equation. The Hamilton-Jacobi equation is *fully nonlinear*, i.e. it depends in a nonlinear way on the highest order derivatives. Finally, if the equation contains no terms which are independent of the unknown function x, the PDE is called *homogeneous*. Otherwise it is called *inhomogeneous*. For example, the Laplace equation is homogeneous, while the Poisson equation is the inhomogeneous variation. In general, it is not possible to obtain a solution of (2.1) in closed (explicit) form.

2.1.1 Classification of second-order PDEs

Partial differential equations can be classified into three different categories [29]: *elliptic, parabolic*, and *hyperbolic* equations. We restrict our attention to the case of a linear second-order PDE with constant coefficients of the form

$$A\frac{\partial^2 x}{\partial v_1^2} + B\frac{\partial^2 x}{\partial v_1 \partial v_2} + C\frac{\partial^2 x}{\partial v_2^2} + D\frac{\partial x}{\partial v_1} + E\frac{\partial x}{\partial v_2} + Fx = G, \tag{2.2}$$

where $A, B, C, D, E, F, G \in \mathbb{R}$, and v_1, v_2 represent two of the $d+1$ independent variables in (2.1), one of which being the time variable. The classification is based on the sign of the discriminant $\Delta = B^2 - 4AC$, i.e.

- if $\Delta < 0$, the equation is called *elliptic*,
- if $\Delta = 0$, the equation is called *parabolic*,
- if $\Delta > 0$, the equation is called *hyperbolic*.

The above classification depends exclusively on the coefficients of the highest derivatives, and the names assigned to each class of partial derivative operator recall the three types of conic section in the Euclidean plane. As a matter of fact, the quadratic algebraic equation

$$Av_1^2 + Bv_1v_2 + Cv_2^2 + Dv_1 + Ev_2 + F = G,$$

represents an ellipse, a parabola or a hyperbola in the Cartesian plane (v_1, v_2) depending on whether the discriminant Δ is negative, null or positive. The three types of PDE exhibit different features. Hyperbolic equations are most commonly associated with convection or transport, parabolic equations are most commonly associated with diffusion and elliptic equations model stationary situations, with no evolution in time. Thus, they are most commonly associated with steady states of either parabolic or hyperbolic problems. The canonical examples of the three families listed above are the Laplace, heat, and wave (or Helmholtz) equations, respectively [30].

2.1.2 Parabolic equations

Parabolic PDEs describe evolution systems where the field of interest varies not only in space, but also in time. A prototypical example of parabolic PDE is the *diffusion* model, governing, for instance, the transport of a substance

due to the molecular motion of the surrounding medium. In this case, x represents the concentration of a polluting material or of a solute in a liquid or a gas. More generally, a parabolic PDE describes generic time-dependent *transport* phenomena that include the effects of *diffusion*, *advection* and *reaction*, according to the following linear second-order partial differential equation [31], [32]

$$\frac{\partial x}{\partial t} + \mathcal{L}(x) = f, \qquad x \in \Omega, \qquad t \in \mathbb{R}_+, \tag{2.3}$$

where $\Omega \subset \mathbb{R}^d$ is the *bounded* domain within which the unknown field $x = x(\mathbf{p}, t)$ evolves at each time instant $t > 0$. The exogenous term on the right-hand side of (2.3) is a given (possibly time-space varying) real-valued function $f = f(\mathbf{p}, t)$, while $\mathcal{L}$ is a generic *linear differential elliptic operator* acting on the unknown function x. We can write the operator in *divergence form*[1] as

$$\mathcal{L}(x) = -\nabla \cdot (\mathbf{\Lambda} \nabla x) + \mathbf{v} \cdot \nabla x + g\, x \tag{2.4}$$

where $\mathbf{\Lambda} = \mathbf{\Lambda}(\mathbf{p}, t) \in \mathbb{R}^{d \times d}$, $\mathbf{v} = \mathbf{v}(\mathbf{p}, t) \in \mathbb{R}^d$, and $g = g(\mathbf{p}, t) \in \mathbb{R}$. In this case, (2.3) is referred to as the *advection-diffusion-reaction* equation, which is said to be *parabolic* if $\mathbf{\Lambda}$ in (2.4) is a *positive definite* matrix, i.e.

$$\sum_{i,j=1}^{d} \lambda_{ij}(\mathbf{p}, t) w_i w_j > 0 \quad \forall \mathbf{p} \in \Omega, \quad \forall t > 0, \quad \forall \mathbf{w} \in \mathbb{R}^d, \quad \mathbf{w} \neq \mathbf{0}.$$

The terms appearing in (2.4) admit the following physical interpretation

- the diffusion term $-\nabla \cdot (\mathbf{\Lambda} \nabla x)$ is due to a nonuniform spatial distribution of the unknown x, where the rate of diffusion is given by the *diffusivity* (matrix of diffusion coefficients) $\mathbf{\Lambda}$;
- the advection term $\mathbf{v} \cdot \nabla x$ is due to a transport process proportional to the gradient of the unknown field, where $\mathbf{v}$ is a velocity field;
- the reaction term $g\, x$ accounts for linear growth or decay of the unknown function.

[1] Let $\mathbf{p} = [\xi\ ,\ \eta]^T$ be the position vector in a two-dimensional coordinate system, and f a scalar field. The *gradient* of f is defined as the vector $\nabla f = [\frac{\partial f}{\partial \xi}\ ,\ \frac{\partial f}{\partial \eta}]^T$. Furthermore, let $\mathbf{F} = [F_\xi\ ,\ F_\eta]^T$ be a vector field, the *divergence* of $\mathbf{F}$, is defined as the scalar field $\nabla \cdot \mathbf{F} = \frac{\partial F_\xi}{\partial \xi} + \frac{\partial F_\eta}{\partial \eta}$. Finally, the *Laplacian* of a scalr field is the divergence of the gradient, i.e. $\triangle f = \nabla \cdot (\nabla f)$. These definitions continue to be valid also for higher dimensions.

Hence, with $\mathcal{L}(x)$ given by (2.4), the general second-order parabolic PDE (2.3) takes the form

$$\frac{\partial x}{\partial t} - \nabla \cdot (\mathbf{\Lambda} \nabla x) + \mathbf{v} \cdot \nabla x + g\, x = f, \qquad x \in \Omega, \qquad t \in \mathbb{R}^+ \tag{2.5}$$

which is a set of *conservation laws* arising from a balance of the quantity x with the advective and conductive fluxes entering and/or leaving a control volume. In the context of heat transfer, equation (2.5) describes the evolution of a temperature field x under the combined effects of thermal diffusion, advection, reaction, and external heat sources f.

2.1.3 Initial and boundary conditions

Since the same PDE may describe a wide variety of sytem dynamics, some additional information is required to complete the problem statement. In practical applications, the processes to be investigated take place in a concrete geometry (e.g., in turbines, chemical reactors, heat exchangers, car engines etc.) during a finite interval of time. The choice of the domain and of the time interval to be considered is dictated by the nature of the problem at hand, by the objectives of the analytical or numerical study, and by the available resources. Another relevant aspect is that a PDE has to be supplemented by suitable initial and boundary conditions to give a well-posed problem with a unique solution. Let $\Omega \subset \mathbb{R}^d$ be a bounded domain and $[0, T]$ be the time interval of interest. The initial conditions, also called *Cauchy conditions*, model the spatial distribution, or *profile*, of the unknown field at some initial time t_0. In order to determine a unique evolution, we also need information about the behaviour of the solution x at the domain boundary $\partial\Omega$, and hence suitable boundary conditions must be added. There are three broad classes of boundary conditions [29]:

- *Dirichlet boundary conditions.* The value of the field x is specified on the boundary, i.e.

$$x(\mathbf{p}, t) = \mu(\mathbf{p}, t) \quad \forall \mathbf{p} \in \partial\Omega, \quad \forall t \in [0, T]; \tag{2.6}$$

- *Neumann boundary conditions.* The normal derivative (*flux*) of the field $\nabla x \cdot \mathbf{n} = \partial x / \partial \mathbf{n}$ is imposed on the boundary, i.e.

$$\frac{\partial x}{\partial \mathbf{n}}(\mathbf{p}, t) = \gamma(\mathbf{p}, t) \quad \forall \mathbf{p} \in \partial\Omega, \quad \forall t \in [0, T] \tag{2.7}$$

$\mathbf{n}$ being the outward pointing unit normal vector of $\partial\Omega$, while γ is the prescribed flux distribution, that is, in general a function of position and time. If $\gamma > 0$ an *incoming flux* is assigned, while $\gamma < 0$ corresponds to an *outgoing flux*;

- *Robin boundary conditions.* The value of a linear combination of the field and its normal derivative is specified on the boundary, i.e.

$$\alpha(\mathbf{p},t)\frac{\partial x}{\partial \mathbf{n}}(\mathbf{p},t) + \beta(\mathbf{p},t)x(\mathbf{p},t) = \gamma(\mathbf{p},t) \quad \forall \mathbf{p} \in \partial\Omega,\, \forall t \in [0,T] \quad (2.8)$$

where α and β are in general space-time dependent, but usually constant. Robin conditions are also called *generalized Neumann boundary conditions.* Indeed, a Robin condition with $\beta = 0$ simplifies to a Neumann condition. When $\mu = 0$ and $\gamma = 0$, the boundary conditions are said to be *homogeneous.* In the case of homogeneous Dirichlet conditions this is equivalent to specify a zero field on the boundary, or, for Neumann conditions, to assume there is no flux across the boundary, i.e. the domain is insulated from the surrounding environment (no-flux condition). Boundary conditions are, therefore, imposed by suitably specifying the functions μ, α, β, and γ on $\partial\Omega$. Finally, different types of conditions can be assigned to different portions of the boundary of the considered domain. In such a case, the associated boundary conditions are said to be *mixed.* The combination of a PDE with assigned initial conditions and boundary conditions is referred to as an *initial boundary value problem* (IBVP).

2.1.4 Example: the heat equation

The prototype of a parabolic PDE (2.4) with $\mathcal{L}(x) = -\nabla \cdot (\lambda \nabla x)$ is the so called *heat equation* [31], [32]

$$\frac{\partial x}{\partial t} - \nabla \cdot (\lambda \nabla x) = f, \qquad x \in \Omega, \qquad t \in \mathbb{R}_+ \qquad (2.9)$$

which provides the basic tool for heat conduction analysis. When the *thermal diffusivity* λ is space-independent, we can rewrite $\mathcal{L}(x) = -\lambda\nabla^2 x$, where $\nabla^2 = \nabla \cdot \nabla$ denotes the Laplacian operator. The heat equation describes the evolution in time of the temperature $x(\mathbf{p},t)$ in homogeneous and isotropic heat-conducting media occupying the region Ω. Modeling the heat conduction process requires to apply thermodynamics of energy conservation along

with *Fourier's law* of heat conduction that for a homogeneous medium takes the form [33], [34]

$$q = -\kappa A \frac{\partial x}{\partial \mathbf{p}} \tag{2.10}$$

Fourier's law models the rate of heat transfer which depends on a physical property of the medium, the *thermal conductivity* κ. A is the cross-sectional area normal to direction of heat flow. The minus sign in (2.10) is a consequence of the second law of thermodynamics, requiring that, whenever a temperature gradient exists, heat must flow in the direction from higher to lower temperature. Heat is transferred through a complex submicroscopic mechanism in which atoms interact by elastic and inelastic collisions to propagate the energy from regions of higher to regions of lower temperature. Thanks to Fourier's law, the complexities of the molecular mechanisms can be neglected, and the rate of heat propagation can be directly evaluated through (2.10). We consider the heat conduction conditions in solids and structures, i.e. we assume the material particles of the body of interest are at rest. The thermal diffusivity $\lambda\,(m^2/s)$, which appears on the left-hand side of (2.9), is defined as $\lambda = \kappa/\rho c$, ρ being the mass density (kg/m^3) and c the specific heat $(J/kg \cdot K)$, both assumed constant. The physical significance of thermal diffusivity is associated with the speed of propagation of heat into the medium during changes of temperature. The higher the thermal diffusivity, the faster is the response of a medium to thermal perturbations, and the faster such changes propagate throughout the medium.

Heat transfer problems are classified according to the variables that influence the temperature. If the temperature is a function of time, the problem is classified as *unsteady* or *transient*. If the temperature is independent of time, the problem is called a *steady-state* problem. If the temperature is a function of a single space coordinate, the problem is said to be *one-dimensional*.

Note that equation (2.9), also known as the *diffusion* equation, describes a much more general model, where *diffusion* not only concerns *heat* but also, for instance, *mass transfer*, i.e. the transport of a substance due to the molecular motion of the surrounding medium. In this case, x may represent the chemical concentration in a liquid or a gas and equation (2.10) is known as the *Fick's law* of diffusion [33].

As we have mentioned in Section 2.1.3, the governing equations in a mathematical model have to be supplemented by additional information in order to obtain a well posed problem, i.e. a problem that has exactly one solution, depending continuously on the data. To determine the temperature

distribution in a medium, it is necessary to solve the appropriate form of the heat equation. However, such a solution depends on the physical conditions existing at the boundaries of the medium and, since the situation is time-dependent, on conditions existing in the medium at some initial time. The heat equation is second-order in the spatial coordinates, hence two boundary conditions must be expressed for each coordinate in order to describe the system. Moreover, the equation being first-order in time, one initial condition must be specified.

In heat transfer problems, the three types of boundary conditions introduced in Section 2.1.3 can be derived by considering conservation of energy at the surface as follows.

- The *Dirichlet* condition corresponds to a situation where the boundary surface is maintained at a fixed temperature, i.e.

$$x = T \quad \text{on } \partial\Omega \tag{2.11}$$

  where T is a prescribed temperature (which in general can be a function of position and time). This condition is closely approximated, for instance, when the surface is in contact with a melting solid or a boiling liquid. In both cases, there is heat transfer at the surface, while the surface remains at the temperature of the phase change process.

- The *Neumann* condition corresponds to the existence of a fixed heat flux at the boundary surface. This heat flux is related to the temperature gradient at the surface by Fourier's law

$$-\kappa \frac{\partial x}{\partial \mathbf{n}} = \gamma \quad \text{on } \partial\Omega \tag{2.12}$$

  Boundary conditions of this type may physically correspond to heaters (e.g., thin electric strip heaters) attached to the surface. A special case of this condition corresponds to the perfectly insulated, or *adiabatic*, surface with zero heat flux for which

$$\frac{\partial x}{\partial \mathbf{n}} = 0 \quad \text{on } \partial\Omega \tag{2.13}$$

- The *Robin* boundary condition corresponds to the existence of *convection* heating (or cooling) at the surface and is obtained from the *Newton's law* of cooling describing the convective heat flux

$$q_{\text{conv}} = \nu \left(x - x_e\right) \tag{2.14}$$

where x_e is the so called *external* or *reference* temperature of the surrounding ambient fluid (e.g., liquid or gas) and ν is the *convection heat transfer coefficient* of units $W/(m^2 \cdot K)$. Conservation of energy at the surface boundary takes the form

$$-\kappa \frac{\partial x}{\partial \mathbf{n}} = \nu\,(x - x_e) \quad \text{on } \partial\Omega \tag{2.15}$$

Positive convective heat flux is considered in the direction of the surface normal (i.e., away from the surface).

A Robin (*convection*[2]) boundary condition is physically different than Dirichlet (prescribed temperature) or Neumann (prescribed flux) boundary conditions in that the temperature gradient within the considered body at the surface is now coupled to the convective flux at the body-fluid interface. Neither the flux nor the temperature are prescribed, but rather, a balance between conduction and convection is forced, with the exact surface temperature and surface heat flux determined by the combination of convection coefficient, thermal conductivity, and ambient fluid temperature. Clearly, Dirichlet and Neumann boundary conditions can be obtained from the Robin condition as special cases if κ and ν are treated as coefficients. For example, by setting $\nu\, x_e = \gamma$ and then letting $\nu = 0$ in the first term of the right-hand side, (2.15) reduces to (2.12). From a physical point of view, Robin boundary conditions are the most common in practice in that many actual systems are governed by a natural energy balance between conduction and convection.

2.2 Weak formulation of parabolic problems

In this section we present the *weak* or *variational* form of initial-boundary value problems for second-order parabolic equations, introduced in Section 2.1.2, of the form (2.5)

$$\frac{\partial x}{\partial t} + \mathcal{L}(x) = f, \qquad x \in \Omega, \qquad t \in \mathbb{R}_+ \tag{2.16}$$

satisfying an initial (or *Cauchy*) condition

$$x(\mathbf{p}, 0) = x_0(\mathbf{p}) \quad \text{in } \Omega \tag{2.17}$$

[2]Convection is the process in which a physical property is propagated (i.e., convected) through space by the motion of the medium occupying the space. Fluid flow is a common example of convection [29].

and one of the conventional boundary conditions (Dirichlet, Neumann, Robin or mixed)

$$\mathcal{B}(x) = 0 \quad \text{on } \partial\Omega \tag{2.18}$$

In (2.16) Ω is a bounded domain in $\mathbb{R}^n$ and the linear operator is given by (2.4), i.e.

$$\mathcal{L}(x) = -\lambda\nabla^2 x + \mathbf{v}\cdot\nabla x + g\,x \tag{2.19}$$

where we used $\mathbf{\Lambda} = \lambda\mathbf{I}, \mathbf{I} \in \mathbb{R}^n$ being the identity matrix and $\lambda > 0$. The matrix of diffusion coefficients is therefore assumed constant in time and spatially uniform, so that the diffusive term can be rewritten using the Laplacian operator as follows

$$-\nabla\cdot(\mathbf{\Lambda}\nabla x) = -\nabla\cdot(\lambda\mathbf{I}\,\nabla x) = -\lambda\nabla\cdot\nabla x = -\lambda\nabla^2 x$$

A function $x \in C^2(\Omega)\cap C(\overline{\Omega})$ satisfying (2.16)-(2.18) is called a *classical solution* of this problem. The space $C^k(\Omega)$ is defined to be the set of all real-valued functions x defined on Ω with the property that x and its partial derivatives up to order k are all continuous on Ω. The *closure* $\overline{\Omega}$ of Ω is the union of Ω and $\partial\Omega$. We know from the theory of partial differential equations [32] that (2.16)-(2.18) has a unique classical solution, provided that $\partial\Omega, f$, and the coefficients of $\mathcal{L}$ are sufficiently smooth. However, in many applications one has to consider equations where these smoothness requirements are no longer met, and, hence, the classical theory is inappropriate. In order to overcome these limitations and to be able to deal with PDEs with non-smooth quantities, we generalize the notion of solution by weakening the differentiability requirements on x. To begin with, let us suppose that x is a classical solution of (2.16)-(2.18). Then, for any sufficiently smooth generic weight function $\psi \in C_0^1(\Omega)$, $C_0^1(\Omega) = \{\psi \in C^1(\Omega) : \psi = 0 \text{ on } \partial\Omega\}$, multiplying both sides of the PDE by ψ and integrating over Ω yields

$$\int_\Omega \frac{\partial x}{\partial t}\psi\,d\mathbf{p} - \int_\Omega \lambda\nabla^2 x\psi\,d\mathbf{p} + \int_\Omega \mathbf{v}\cdot\nabla x\psi\,d\mathbf{p} + \int_\Omega gx\psi\,d\mathbf{p} = \int_\Omega f\psi\,d\mathbf{p} \tag{2.20}$$

In this context, ψ is referred to as a *test function*. The idea is to check whether the PDE holds in the weighted average sense over Ω using the test function ψ to define the weights in the average. Clearly, the fact that (2.20) holds for a particular test function ψ does not mean that (2.16) holds. However, if (2.20) holds for all test functions from a sufficiently large set, then

(2.16) must hold. Note that the integral form (2.20) still involves second order derivatives of the unknown field. In order to obtain lower-order derivatives, the next step is to apply *Green's identity* to the diffusive term in the left-hand side of (2.20):

$$\int_\Omega \nabla^2 x\psi \, d\mathbf{p} = \int_{\partial\Omega} \frac{\partial x}{\partial \mathbf{n}} \psi \, d\mathbf{p} - \int_\Omega \nabla x \cdot \nabla \psi \, d\mathbf{p} \; . \tag{2.21}$$

Finally, by substituting (2.21) into (2.20), we obtain the *weak* form of the initial-boundary vale problem (2.16)

$$\begin{aligned} \int_\Omega \frac{\partial x}{\partial t} \psi \, d\mathbf{p} - \int_{\partial\Omega} \lambda \frac{\partial x}{\partial \mathbf{n}} \psi \, d\mathbf{p} + \int_\Omega \lambda \nabla x \cdot \nabla \psi \, d\mathbf{p} \\ + \int_\Omega \mathbf{v} \cdot \nabla x \psi \, d\mathbf{p} + \int_\Omega g x \psi \, d\mathbf{p} = \int_\Omega f \psi \, d\mathbf{p} \end{aligned} \tag{2.22}$$

which is valid for any test function ψ. Clearly if x is a classical solution of (2.16)-(2.18), then it is also a weak solution. However, the converse is not true. If (2.16)-(2.18) admit a weak solution, this may not be smooth enough to be a classical solution. The original PDE (2.16) suggests the solution should have partial derivatives up to order two, i.e. x should be twice differentiable, and f should be continuous over $\overline{\Omega}$. Nonetheless, the variational form (2.22) involves only the first derivatives of x, and it is only necessary that f be integrable. This is the reason why (2.22) is referred to as the *weak form* of the original IBVP which, by contrast, can be called the *strong form*. Notice that the weakest possible assumptions should be made on the functions involved, so as to include as many cases as possible. To this end, we introduce the *Sobolev* spaces [35] below. First of all, it is convenient to define the space of *square-integrable* functions

$$L^2(\Omega) = \left\{ f : \Omega \to \mathbb{R} \;\middle|\; \int_\Omega \mid f \mid^2 \, d\Omega < \infty \right\} \; . \tag{2.23}$$

The righ-hand side of the PDE will be required to belong to $L^2(\Omega)$. Moreover, if $\Omega \in \mathbb{R}^2$, then the solution x of (2.22) and the test functions must satisfy

$$x, \frac{\partial x}{\partial \xi}, \frac{\partial x}{\partial \eta} \in L^2(\Omega) \tag{2.24}$$

where (ξ, η) are the spatial coordinates of the position vector $p \in \Omega$. The above conditions define the *Sobolev* space $H^1(\Omega)$:

$$H^1(\Omega) = \left\{ f \in L^2(\Omega) \;\middle|\; \frac{\partial f}{\partial \xi} \in L^2(\Omega) \; , \; \frac{\partial f}{\partial \eta} \in L^2(\Omega) \right\} \tag{2.25}$$

Both the solution and the test functions must also satisfy the boundary conditions. In particular, for a Dirichlet condition it is convenient to introduce

$$H_0^1(\Omega) = \left\{ f \in H^1(\Omega) \mid f = 0 \text{ on } \partial\Omega \right\} \tag{2.26}$$

which is another example of a *Sobolev* space. The assumptions on the functions appearing in the variational form (2.22) can be stated in terms of *weak derivatives*, therefore x and ψ are assumed to be only weakly differentiable. Moreover, the weak form only requires weak derivatives of order one, whereas in the strong form (2.16) x must have continuous derivatives up to order two. This is certainly a substantial relaxation of the requirements on x. One of the main advantages of extending the class of solutions of a IBVP from classical solutions with continuous derivatives to weak solutions with weak derivatives is that it is easier to prove the existence of weak solutions. Once established the existence of weak solutions, one may then study their properties, such as uniqueness and regularity, and try to prove under appropriate assumptions that the weak solutions are, in fact, classical solutions.

It is important to note that different boundary conditions in (2.18) will lead to different weak formulations of (2.22). In particular, since we look for a weak solution in the space of test functions, Dirichlet conditions will be explicitly imposed in the weak form, while Neumann and Robin conditions will be only implicitly contained in (2.22). This is why Dirichlet conditions are usually called *essential* boundary conditions, whereas Neumann/Robin conditions are said to be *natural*.

In the next sections we will examine in more detail how the variational formulation (2.22) may vary according to the different type of boundary condition under consideration.

2.3 Dirichlet problem

2.3.1 Homogeneous case

Let us now consider the weak formulation of parabolic problems with homogeneous Dirichlet boundary conditions (2.6)

$$\begin{aligned} \frac{\partial x}{\partial t} - \lambda \nabla^2 x + \mathbf{v} \cdot \nabla x + g x &= f && \text{in } \Omega \\ x(\mathbf{p}, 0) &= x_0(\mathbf{p}) && \forall \mathbf{p} \in \Omega \\ x &= 0 && \text{on } \partial\Omega \end{aligned} \tag{2.27}$$

where $\lambda > 0$ and $f \in L^2(\Omega)$. The Dirichlet conditions are essential as they appear explicitly in the weak form, through the requirement that $x \in H_0^1(\Omega)$ and hence $\psi \in H_0^1(\Omega)$. In this case, the integral term defined over the boundary $\partial\Omega$ appearing in the left-hand side of (2.22) cancels out, since from (2.26) $\psi = 0$ on $\partial\Omega$. The weak form of the Dirichlet homogeneous problem (2.27) is thus defined as follows.

Find $x \in H_0^1(\Omega)$ *such that* $\forall \psi \in H_0^1(\Omega)$

$$\int_\Omega \frac{\partial x}{\partial t}\psi \, d\mathbf{p} + \int_\Omega \lambda \nabla x \cdot \nabla \psi \, d\mathbf{p} + \int_\Omega \mathbf{v} \cdot \nabla x \psi \, d\mathbf{p} + \int_\Omega g x \psi \, d\mathbf{p} = \int_\Omega f \psi \, d\mathbf{p} \quad (2.28)$$

where $f \in L^2(\Omega)$.

2.3.2 Inhomogeneous case

The inhomogeneous Dirichlet problem is described by the following IBVP:

$$\begin{aligned} \frac{\partial x}{\partial t} - \lambda \nabla^2 x + \mathbf{v} \cdot \nabla x + g x &= f && \text{in } \Omega \\ x(\mathbf{p}, 0) &= x_0(\mathbf{p}) && \forall \mathbf{p} \in \Omega \\ x &= \mu && \text{on } \partial\Omega \end{aligned} \quad (2.29)$$

where $\alpha > 0$, $f \in L^2(\Omega)$ and $\mu \neq 0$ is a function defined on $\partial\Omega$ that must satisfy some regularity conditions. Usually the inhomogeneous case is converted to the homogeneous one presented in Section 2.3.1 by a suitable change of variable. To this end, it is assumed there is a function $\widehat{\mu} \in H^1(\Omega)$ such that $\widehat{\mu} = \mu$ on $\partial\Omega$. It turns out that the correct space of test functions is still $H_0^1(\Omega)$. By defining $z = x - \widehat{\mu}$, the inhomogeneous Dirichlet boundary condition in (2.29) can be written as a homogeneous condition $z = 0$ on $\partial\Omega$. Thus the solution takes the form $x = z + \widehat{\mu}$, where $\widehat{\mu}$ is assumed to be known and $z \in H_0^1(\Omega)$ is unknown. The inhomogeneous Dirichlet IBVP (2.29) can now be given the following variational formulation:

Find $x = z + \widehat{\mu}$, $z \in H_0^1(\Omega)$ *such that* $\forall \psi \in H_0^1(\Omega)$

$$\begin{aligned} & \int_\Omega \frac{\partial z}{\partial t}\psi \, d\mathbf{p} + \int_\Omega \lambda \nabla z \cdot \nabla \psi \, d\mathbf{p} + \int_\Omega \mathbf{v} \cdot \nabla z \psi \, d\mathbf{p} + \int_\Omega g z \psi \, d\mathbf{p} \\ = \quad & \int_\Omega f \psi \, d\mathbf{p} - \int_\Omega \frac{\partial \widehat{\mu}}{\partial t}\psi \, d\mathbf{p} - \int_\Omega \lambda \nabla \widehat{\mu} \cdot \nabla \psi \, d\mathbf{p} \\ & - \int_\Omega \mathbf{v} \cdot \nabla \widehat{\mu} \psi \, d\mathbf{p} - \int_\Omega g \widehat{\mu} \psi \, d\mathbf{p} \end{aligned} \quad (2.30)$$

where $f \in L^2(\Omega)$, *and* $\widehat{\mu} \in H^1(\Omega)$ *is such that* $\widehat{\mu} = \mu$ *on* $\partial\Omega$.

In general, it may be complicated to find a suitable function $\widehat{\mu}$ satisfying the Dirichlet condition $\widehat{\mu} = \mu$ on $\partial\Omega$ (in an IBVP only μ is given, not $\widehat{\mu}$). However, in practice the weak form of a PDE is usually solved via spatial discretization, and hence it is easier to produce via numerical methods a function $\widehat{\mu}$ (approximately) satisfying the inhomogeneous Dirichlet condition. This will be clarified later, once we will consider how to approximately solve the weak form of PDEs. Notice also that (2.30) is the same expression as (2.28), except for the right-hand side term of the equation. Indeed, when dealing with inhomogeneous Dirichlet conditions, the forcing term in the weak form of the IBVP takes into account the boundary function μ through $\widehat{\mu}$ and its derivatives. Furthermore, observe that all the integrals appearing in (2.30) that involve $\widehat{\mu}$ and its derivatives can be computed, since we assumed $\widehat{\mu} \in H^1(\Omega)$.

2.4 Neumann problem

2.4.1 Homogeneous case

Next we derive the weak form of the homogeneous Neumann problem that takes the form

$$\begin{aligned} \frac{\partial x}{\partial t} - \lambda\nabla^2 x + \mathbf{v}\cdot\nabla x + gx &= f && \text{in } \Omega \\ x(\mathbf{p},0) &= x_0(\mathbf{p}) && \forall \mathbf{p} \in \Omega \\ \lambda\frac{\partial x}{\partial \mathbf{n}} &= 0 && \text{on } \partial\Omega \end{aligned} \tag{2.31}$$

where $\lambda > 0$ and $f \in L^2(\Omega)$. Here we deal with a natural boundary condition of type (2.7) that will not appear explicitly in the weak form. Indeed, the boundary integral $\int_{\partial\Omega} \lambda\frac{\partial x}{\partial \mathbf{n}}\psi \, d\mathbf{p}$ appearing on the left-hand side of (2.22) vanishes because of the homogeneous Neumann condition $\lambda\frac{\partial x}{\partial \mathbf{n}} = 0$ satisfied by the solution x on the boundary. The following homogeneous Neumann initial boundary-value problem is therefore obtained:

Find $x \in H^1(\Omega)$ *such that* $\forall \psi \in H^1(\Omega)$

$$\int_\Omega \frac{\partial x}{\partial t}\psi \, d\mathbf{p} + \int_\Omega \lambda\nabla x\cdot\nabla\psi \, d\mathbf{p} + \int_\Omega \mathbf{v}\cdot\nabla x\psi \, d\mathbf{p} + \int_\Omega gx\psi \, d\mathbf{p} = \int_\Omega f\psi \, d\mathbf{p} \tag{2.32}$$

where $f \in L^2(\Omega)$.

It is worth pointing out that the variational formulation (2.32) takes the same form of (2.28) characterizing a homogeneous Dirichlet IBVP. However, this does not mean that solving a homogeneous Neumann problem is equivalent to finding the weak solution of a homogeneous Dirichlet problem. Indeed, the same result is obtained for two different reasons. Specifically, whereas in the Dirichlet case it was the boundary condition on the test function ψ that caused the boundary integral to vanish, in the Neumann case this is due to the condition satisfied by the solution on the boundary. This, in turn, leads to a weak solution of the Neumann problem requiring $x \in H^1(\Omega)$, while the weak solution of the Dirichlet problem is such that $x \in H_0^1(\Omega)$.

2.4.2 Inhomogeneous case

Let us now consider the inhomogeneous Neumann problem

$$\begin{aligned} \frac{\partial x}{\partial t} - \lambda \nabla^2 x + \mathbf{v} \cdot \nabla x + gx &= f && \text{in } \Omega \\ x(\mathbf{p}, 0) &= x_0(\mathbf{p}) && \forall \mathbf{p} \in \Omega \\ \lambda \frac{\partial x}{\partial \mathbf{n}} &= \gamma && \text{on } \partial\Omega \end{aligned} \tag{2.33}$$

where $\lambda > 0$, $\gamma \neq 0$ and $f \in L^2(\Omega)$. The above Neumannn IBVP can be solved in an analogous way to the homogeneous one. However, in this case the inhomogeneous Neumann condition in (2.33) leads to a non-zero flux across the boundary, and hence the integral term in (2.22) depending on the normal derivative of the unknown does not vanish. The weak formulation of the inhomogeneous Neuman problem can be stated as follows.

Find $x \in H^1(\Omega)$ *such that* $\forall \psi \in H^1(\Omega)$

$$\begin{aligned} \int_\Omega \frac{\partial x}{\partial t} \psi \, d\mathbf{p} + \int_\Omega \lambda \nabla x \cdot \nabla \psi \, d\mathbf{p} \; + \int_\Omega \mathbf{v} \cdot \nabla x \psi \, d\mathbf{p} + \int_\Omega gx\psi \, d\mathbf{p} \\ = \int_\Omega f\psi \, d\mathbf{p} + \int_{\partial\Omega} \gamma\psi \, d\mathbf{p} \end{aligned} \tag{2.34}$$

where $f \in L^2(\Omega)$.

Notice that, as previously observed in the case of Dirichlet conditions, the weak form (2.34) is the same as the one obtained for homogeneous Neumann conditions, except for the forcing term on the right-hand side which

now accounts for the non-zero condition on the boundary of the domain of interest.

2.5 Robin problem

2.5.1 Homogeneous case

Consider the following parabolic PDE with Robin boundary conditions (2.8)

$$\begin{aligned} \frac{\partial x}{\partial t} - \lambda \nabla^2 x + \mathbf{v} \cdot \nabla x + gx &= f && \text{in } \Omega \\ x(\mathbf{p}, 0) &= x_0(\mathbf{p}) && \forall \mathbf{p} \in \Omega \\ \alpha \frac{\partial x}{\partial \mathbf{n}} + \beta x &= 0 && \text{on } \partial\Omega \end{aligned} \tag{2.35}$$

where $\alpha > 0$, $\beta > 0$ and $f \in L^2(\Omega)$. The Robin boundary condition is natural, therefore we can proceed as we did for the Neumann case by substituting in (2.22) the term proportional to the flux of the field x across the boundary given by the Robin condition. It is clear from (2.35) that the following expression holds on the boundary $\partial\Omega$:

$$-\alpha \frac{\partial x}{\partial \mathbf{n}} = \beta x.$$

Subsequently, the homogeneous Robin problem can be written in the weak form as follows.

Find $x \in H^1(\Omega)$ *such that* $\forall \psi \in H^1(\Omega)$

$$\begin{aligned} \int_\Omega \frac{\partial x}{\partial t} \psi \, d\mathbf{p} + \int_{\partial\Omega} \beta x \psi \, d\mathbf{p} + \int_\Omega \alpha \nabla x \cdot \nabla \psi \, d\mathbf{p} \\ + \int_\Omega \mathbf{v} \cdot \nabla x \psi \, d\mathbf{p} + \int_\Omega g x \psi \, d\mathbf{p} = \int_\Omega f \psi \, d\mathbf{p} \end{aligned} \tag{2.36}$$

where $f \in L^2(\Omega)$.

It can be easily noticed from (2.36) that the homogeneous Robin condition brings an additional term, proportional to the unknown, on the left-hand side of the equation. In this case, however, the forcing term on the right-hand side is not modified.

2.5.2 Inhomogeneous case

The Robin problem with inhomogeneous boundary conditions is described by

$$\begin{aligned} \frac{\partial x}{\partial t} - \lambda \nabla^2 x + \mathbf{v} \cdot \nabla x + gx &= f && \text{in } \Omega \\ x(\mathbf{p}, 0) &= x_0(\mathbf{p}) && \forall \mathbf{p} \in \Omega \\ \alpha \frac{\partial x}{\partial \mathbf{n}} + \beta x &= \gamma && \text{on } \partial\Omega \end{aligned} \tag{2.37}$$

where $\alpha > 0$, $\beta > 0$, $\gamma \neq 0$ and $f \in L^2(\Omega)$. As in the previous homogeneous case, the Robin boundary condition from (2.37) is replaced into (2.22), so as to obtain the following weak form

Find $x \in H^1(\Omega)$ *such that* $\forall \psi \in H^1(\Omega)$

$$\begin{aligned} \int_\Omega \frac{\partial x}{\partial t} \psi \, d\mathbf{p} + \int_{\partial\Omega} \beta x \psi \, d\mathbf{p} + \int_\Omega \alpha \nabla x \cdot \nabla \psi \, d\mathbf{p} + \int_\Omega \mathbf{v} \cdot \nabla x \psi \, d\mathbf{p} \\ + \int_\Omega g x \psi \, d\mathbf{p} = \int_\Omega f \psi \, d\mathbf{p} + \int_{\partial\Omega} \gamma \psi \, d\mathbf{p} \end{aligned} \tag{2.38}$$

where $f \in L^2(\Omega)$.

Note that in (2.38) there is an additional forcing term on the right-hand side, analogously to the case of inhomogeneous Neumann conditions. This is due to the fact that Neumann and Robin conditions are of the same type, i.e. natural conditions in the weak formulation. The Robin initial boundary-value problem can, in fact, be considered as a generalization of the Neumann problem in Section 2.4.

2.6 Mixed problem

Let us finally consider a partial differential equation characterized by mixed *natural/essential* boundary conditions. In this case, the overall boundary can be partitioned into two distinct paths $\partial\Omega_D$ and $\partial\Omega_R$ on which *essential* and, respectively, *natural* conditions are applied, and such that $\partial\Omega_D \cup \partial\Omega_R =$

$\partial\Omega$ with $\partial\Omega_D \cap \partial\Omega_R = \emptyset$. The mixed probem takes the following form

$$\begin{aligned} \frac{\partial x}{\partial t} - \lambda\nabla^2 x + \mathbf{v}\cdot\nabla x + gx &= f && \text{in } \Omega \\ x(\mathbf{p},0) &= x_0(\mathbf{p}) && \forall \mathbf{p}\in\Omega \\ x &= \mu && \text{on } \partial\Omega_D \\ \alpha\frac{\partial x}{\partial \mathbf{n}} + \beta x &= \gamma && \text{on } \partial\Omega_R \end{aligned} \tag{2.39}$$

where $\alpha > 0$, $\beta \geq 0$ and $f \in L^2(\Omega)$. A mixed problem is addressed by subdividing the intergal over the boundary into two parts: one defined over $\partial\Omega_D$ and the other over $\partial\Omega_R$. Then, we can proceed separately on each portion of $\partial\Omega$, by following the previously described weak formulation for Dirichlet and Robin problems. In the resulting weak form of a mixed problem all the terms originating in the separate cases of Dirichlet and Robin conditions will therefore appear.

Chapter 3

Finite-element approximation of spatially distributed systems

In this chapter we introduce the fundamental principles of the finite-element approximation of spatially distributed systems, governed by partial differential equations (discussed in the previous chapter). For the vast majority of problems, these PDEs cannot be solved with analytical methods. Instead, a finite-element approximation of the equations can be constructed. We start by presenting the Galerkin method which approximates the weak form of a PDE in a finite-dimensional subspace. Then, the implementation of the finite-element approach is presented following a standard step-by-step procedure. This includes the domain discretization into finite elements, the choice of the approximating subspace, and the derivation of the element equations. Once the individual element equations are derived, they must be assembled to characterize the overall behavior of the system.

3.1 Galerkin method

The Galerkin method produces the best approximation, from a given approximating subspace, to the true solution of a variational problem. In particular, this approximation is the solution of a finite-dimensional system of equations.

First of all, the infinite-dimensional problem is converted into its weak formulation, introduced in Section 2.2 for parabolic PDEs. It can be noticed that in each different type of parabolic initial-boundary value problem

described in Sections 2.3, 2.4, 2.5 and 2.6, the corresponding integral form always contains terms depending on both the unknown field and the test functions, as well as terms exclusively depending on the latter. More specifically, the weak form can be written as follows.

Find $x \in V$ such that

$$\int_\Omega \frac{\partial x}{\partial t}\psi \, d\mathbf{p} + a(x,\psi) = F(\psi), \qquad \forall \psi \in V \tag{3.1}$$

where V is the solution space (e.g. $H_0^1(\Omega)$ for the Dirichlet problem, $H^1(\Omega)$ in the Neumann and Robin cases), $a(\cdot,\cdot)$ is a bilinear form on $V \times V$ associated to operator $\mathcal{L}$, and F is a linear functional on V.

It can be proved that a sufficient condition for the existence and uniqueness of the solution to problem (3.1) is that the bilinear form $a(\cdot,\cdot)$ on a linear space V is *continuous* (or *bounded*), i.e. [35]

$$\exists\, C < \infty \text{ such that } |a(\psi,w)| \le C\|\psi\|_V \, \|w\|_V \qquad \forall \psi, w \in V \tag{3.2}$$

and *weakly coercive* [1], that is

$$\exists\, \epsilon \ge 0, \exists\, \rho > 0 : \quad a(\psi,\psi) + \epsilon\|\psi\|^2_{L^2(\Omega)} \ge \rho\|\psi\|^2_V \qquad \forall \psi \in V \tag{3.3}$$

which yields for $\epsilon = 0$ the standard definition of coercivity.

The Galerkin method is a *projection method* that aims at finding an approximate solution of problem (3.1) in V, by projecting the equation onto some suitable *finite-dimensional* subspace $V_h \subset V$. The higher the dimension of the subspace, the better the approximation of the unknown x, i.e. the dimension of V_h grows as $h \to 0$. We denote by $x_h \in V_h$ the approximate solution of $x \in V$, while V_h is a subspace depending on the discretization parameter h so that:

- $V_h \subset V$,
- $dim\ V_h = n < \infty \quad \forall h > 0, \quad \text{with} \quad n = n(h).$

The approximation of (3.1) is called *Galerkin problem*, defined as follows.

Find $x_h \in V_h$ such that

$$\int_\Omega \frac{\partial x_h}{\partial t}\psi_h \, d\mathbf{p} + a(x_h,\psi_h) = F(\psi_h), \qquad \forall \psi_h \in V_h \tag{3.4}$$

with initial condition $x_h(0) = x_0$.

Such problem is also called *semi-discretization* of (3.1), as the temporal variable has not yet been discretized. In order to carry out the projection, we introduce the (linearly independent) basis functions $\{\phi_j = \phi_j(\mathbf{p}), \quad j = 1, 2, ..., n\}$ for $V_h = \text{span}\{\phi_1, \ldots, \phi_n\}$ where $\mathbf{p} \in \Omega \subset \mathbb{R}^2$. Then, the Galerkin method seeks to find the approximate solution in the form of a linear combination of the basis. We observe that it suffices that (3.4) is verified for the basis functions in order to be satisfied by all the functions of the subspace $\psi_h \in V_h$. Indeed, each test function can be written as a linear combination of the basis. We therefore require the following n equations to be satisfied:

$$\int_\Omega \frac{\partial x_h}{\partial t}\phi_j \, d\mathbf{p} + a(x_h, \phi_j) = F(\phi_j), \qquad \forall j = 1, 2, ..., n \tag{3.5}$$

In this way, the infinite-dimensional problem (3.4) is transformed into problem (3.5) of finite dimension n. Expressing the approximate solution $x_h \in V_h$ in terms of the basis functions $\phi_j(\mathbf{p})$, we can write [35, 36]

$$x_h(\mathbf{p}, t) = \sum_{i=1}^{n} x_i(t)\, \phi_i(\mathbf{p}) = \sum_{i=1}^{n} x_i\, \phi_i, \tag{3.6}$$

where $\{x_i = x_i(t)\}_{i=1}^n$ are time-dependent unknown expansion coefficients to be determined. By substituting (3.6) into (3.5), we obtain

$$\begin{aligned}
\int_\Omega \sum_{i=1}^{n} \dot{x}_i \phi_i \phi_j \, d\mathbf{p} + a\left(\sum_{i=1}^{n} x_i \phi_i, \phi_j\right) &= F(\phi_j), \qquad \forall j = 1, 2, ..., n \\
\sum_{i=1}^{n} \dot{x}_i \int_\Omega \phi_i \phi_j \, d\mathbf{p} + \sum_{i=1}^{n} x_i a(\phi_i, \phi_j) &= F(\phi_j), \qquad \forall j = 1, 2, ..., n \\
\sum_{i=1}^{n} m_{ij}\dot{x}_i + \sum_{i=1}^{n} s_{ij} x_i &= u_j, \qquad \forall j = 1, 2, ..., n
\end{aligned} \tag{3.7}$$

where we denoted by $\dot{x}_i$ the time derivative of function $x_i(t)$, and we used the following definitions:

$$m_{ij} = \int_\Omega \phi_i \phi_j \, d\mathbf{p}, \qquad s_{ij} = a(\phi_i, \phi_j), \qquad u_j = F(\phi_j). \tag{3.8}$$

Thus, the *Galerkin problem* (3.4) can be rewritten as follows.

Find $\{x_1, \ldots, x_n\} \in \mathbb{R}^n$ *such that*

$$\sum_{i=1}^{n} m_{ij}\dot{x}_i + \sum_{i=1}^{n} s_{ij}x_i = u_j, \qquad \forall j = 1, 2, ..., n. \tag{3.9}$$

Note that (3.7) is a system of linear ordinary differential equations that can be rewritten in matrix form as

$$\mathbf{M}\dot{\mathbf{x}} + \mathbf{S}\mathbf{x} = \mathbf{u} \tag{3.10}$$

where we defined the following vectors and matrices:

$$\mathbf{x} = [x_1, x_2, ..., x_n]^T \in \mathbb{R}^n, \tag{3.11}$$

$$\mathbf{M} = \{m_{ij}\}_{i,j=1}^{n} \in \mathbb{R}^{n\times n}, \tag{3.12}$$

$$\mathbf{S} = \{s_{ij}\}_{i,j=1}^{n} \in \mathbb{R}^{n\times n}, \tag{3.13}$$

$$\mathbf{u} = [u_1, u_2, ..., u_n]^T \in \mathbb{R}^n. \tag{3.14}$$

The vector $\boldsymbol{\phi} = [\phi_1, \phi_2, ..., \phi_n]^T \in \mathbb{R}^n$ contains the so-called *shape functions* composing the basis of subspace V_h, whereas $\mathbf{x}$ is the vector of unknown coefficients $\{x_i\}_{i=1}^{n}$. Notice that, if $\boldsymbol{\phi}$ and $\mathbf{x}$ are known, then we can determine the approximation x_h of the unknown field x at each point of the domain $\mathbf{p} \in \Omega$ through (3.6). The forcing term $\mathbf{u}$ is usually called *load vector*, or *source term* of the equation. Matrices $\mathbf{M}$ and $\mathbf{S}$ are often referred to as the *mass* and, respectively, *stiffness* matrix, names coming from corresponding matrices in the context of structural problems. If the coefficients of the PDE under consideration are constant, then the matrices $\mathbf{M}$ and $\mathbf{S}$ are time-independent. These matrices depend on the choice of the shape functions, which is a key point of the Galerkin method. Indeed, based on the selection of the particular approximating subspace and of the shape functions, the Galerkin approximation leads to distinct methods, including the *Finite Element Method (FEM)*.

Next, we point out some properties of the mass and stiffness matrices that are independent of the basis chosen for V_h, but exclusively depend on the properties of the weak problem that is being approximated.

Theorem 1 (Positive definiteness of $\mathbf{M}$ and $\mathbf{S}$, [1])**.** *Given a coercive bilinear form* $a(\cdot,\cdot)$*, the matrices* $\mathbf{M}$ *and* $\mathbf{S}$ *arising from the discretization of a parabolic problem via the Galerkin method are positive definite.*

Proof: Let $\mathbf{w} \in \mathbb{R}^n$ denote a generic vector, $\mathbf{M} \in \mathbb{R}^{n\times n}$ the mass matrix, and $\mathbf{S} \in \mathbb{R}^{n\times n}$ the stiffness matrix with elements $s_{ij} = a(\phi_i, \phi_j)$ where $\{\phi_i\}_{i=1}^n$ is the basis of V_h. Then, it is possible to write an approximation of $\mathbf{w}$ using the function $w_{[h]} = \sum_{i=1}^n w_i\phi_i \in V_h$, and thanks to the coercivity of the bilinear form $a(\cdot,\cdot)$ one has

$$\mathbf{w}^T\mathbf{S}\mathbf{w} = \sum_{i=1}^n\sum_{j=1}^n w_i s_{ij} w_j = \sum_{i=1}^n\sum_{j=1}^n w_i a(\phi_i,\phi_j) w_j = \sum_{i=1}^n\sum_{j=1}^n a(w_i\phi_i, w_j\phi_j) =$$

$$= a\left(\sum_{i=1}^n w_i\phi_i, \sum_{j=1}^n w_j\phi_j\right) = a(w_{[h]}, w_{[h]}) \geq k\|w_{[h]}\|^2 \geq 0.$$

Moreover, if $\mathbf{w}^T\mathbf{S}\mathbf{w} = 0$, then $\|w_h\|^2 = 0$, that is $w_h = 0$ and hence $\mathbf{w} = 0$. In an analogous way, considering the mass matrix we can write

$$\mathbf{w}^T\mathbf{M}\mathbf{w} = \sum_{i=1}^n\sum_{j=1}^n w_i m_{ij} w_j = \sum_{i=1}^n\sum_{j=1}^n w_i \int_\Omega \phi_i\phi_j \, d\mathbf{p} w_j =$$

$$= \int_\Omega \sum_{i=1}^n w_i\phi_i \sum_{j=1}^n w_j\phi_j \, d\mathbf{p} = \int_\Omega w_{[h]}^2 \, d\mathbf{p} = \|w_{[h]}\|^2 \geq 0.$$

Finally, $\mathbf{w}^T\mathbf{M}\mathbf{w} = 0$ leads to $\|w_{[h]}\|^2 = 0$, that is $w_{[h]} = 0$ and thus $\mathbf{w} = 0$. ■

Furthermore, the following property can be proved [1]:

Theorem 2. *The mass matrix* $\mathbf{M}$ *is symmetric, while the stiffness matrix* $\mathbf{S}$ *is symmetric if and only if the bilinear form* $a(\cdot,\cdot)$ *is symmetric.*

□

For instance, in the Poisson problem with either Dirichlet or mixed boundary conditions, the stiffness matrix $\mathbf{S}$ is symmetric and definite positive. Conversely, other properties such as the condition number or the sparsity structure, depend on the specifically considered basis. For instance, bases formed by functions with small support are preferable, as all the elements $a(\phi_i, \phi_j)$ relative to basis functions having supports with empty intersections will be null. More in general, from a computational viewpoint, the most convenient choices of V_h will be the ones requiring a low computational effort for the computation of the matrix elements as well as the source term f.

To sum up, the Galerkin method approximates the original infinite-dimensional system (2.3) into a possibly large-scale finite-dimensional system described by the ordinary differential equation (ODE) (3.10). In particular, the latter is a *linear time-variant continuous-time dynamical system* of finite dimension n. Starting from the *semi-discrete* Galerkin approximation it is possible to obtain a *fully-discrete* (in space and time) system by discretizing in time (3.10) with standard integration methods (e.g., backward or forward Euler, zero-order-hold method, etc.)

3.2 Domain discretization

So far, the two theoretical foundations of the finite element approximation have been presented: the weak form of an initial-boundary value problem and the Galerkin method that provides an approximate solution to a variational equation from a given finite-dimensional subspace.

The key practical ingredient of the finite-element approximation is the domain discretization, which is carried out by means of a suitable *tessellation* (or *triangulation* for triangles) of the computational domain, i.e. by subdividing the domain of interest into a finite set of simple geometric entities called *elements* (e.g., triangles or quadrangles in 2D, tetrahedra, prisms or hexahedra in 3D), connected together at a set of points called *nodes*. The collection of all elements is the so-called *mesh* (or computational grid) $\mathcal{M}_h$.

For simplicity, we restrict our discussion primarily to the two-dimensional case. Moreover, we will deal only with the case of polygonal domains. For computational domains with curved boundaries, the interested reader can refer to [35], [37]. Note that the techniques presented for the 2D case can be extended to three-dimensional domains.

3.2.1 Mesh of a polygonal domain

Let us consider a bounded polygonal domain $\Omega \in \mathbb{R}^2$. We can partition Ω into polygons $\mathcal{E}_i, i = 1, \ldots, v$, with a mesh $\mathcal{M}_h = \{\mathcal{E}_i\}_{i=1}^{v}$ such that

1. the mesh covers the overall domain, i.e.

$$\overline{\Omega} = \bigcup_{i=1}^{v} \mathcal{E}_i \tag{3.15}$$

where $\overline{\Omega}$ is the closure of Ω;

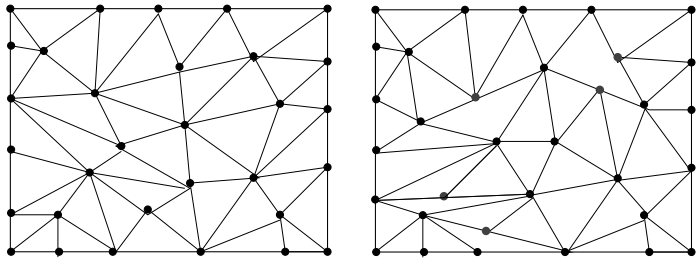


Figure 3.1: Example of conforming (left) and nonconforming (right) grid [1].

2. $\text{int}(\mathcal{E}_1) \cap \text{int}(\mathcal{E}_2) = \emptyset \quad \forall \mathcal{E}_1 \neq \mathcal{E}_2 \in \mathcal{M}_h$, where $\text{int}(\mathcal{E}) = \mathcal{E} \setminus \partial\mathcal{E}$ denotes the interior of $\mathcal{E}$;

3. if $\mathcal{E}_1 \cap \mathcal{E}_2 \neq \emptyset$ with $\mathcal{E}_1 \neq \mathcal{E}_2 \in \mathcal{M}_h$, then such intersection is either an edge or a vertex of the mesh;

4. the *size* of the mesh is defined as the length of the longest edge of the grid, i.e.
$$h = \max_{\mathcal{E} \in \mathcal{M}_h} h_{\mathcal{E}} \tag{3.16}$$
where $h_{\mathcal{E}}$ is the *diameter* of element $\mathcal{E}$ given by
$$h_{\mathcal{E}} = \max_{\mathbf{p}_1, \mathbf{p}_2 \in \mathcal{E}} \sqrt{(\mathbf{p}_1 - \mathbf{p}_2)^T (\mathbf{p}_1 - \mathbf{p}_2)}; \tag{3.17}$$

5. the following *regularity* condition holds
$$\frac{h_{\mathcal{E}}}{\rho_{\mathcal{E}}} \leq \delta \quad \forall \mathcal{E} \in \mathcal{M}_h \tag{3.18}$$
where $\rho_{\mathcal{E}}$ denotes the diameter of the circle inscribed in $\mathcal{E}$ (called *sphericity* of $\mathcal{E}$), and $\delta > 0$.

The second condition imposed on the mesh clearly requires that given two distinct elements, their interiors do not overlap. The third condition limits the admissible triangulations to the so-called *conforming* ones. Fig. 3.1 illustrates a conforming (left) and nonconforming (right) triangulation. In the remainder, conforming triangulations will be considered. However, there also exist finite element approximations which use nonconforming meshes and allow, for instance, the coupling of grids constructed from elements of

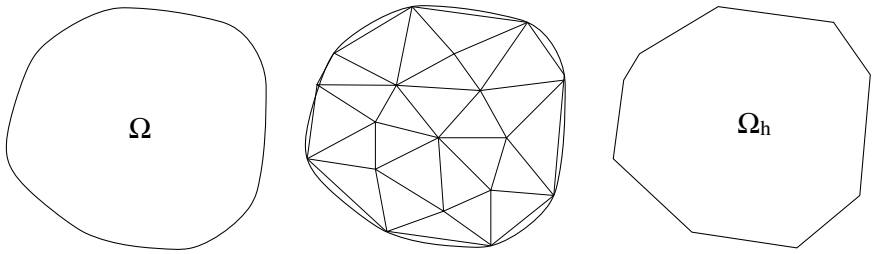


Figure 3.2: Example of triangulation of a non-polygonal domain Ω, which requires an approximation Ω_h [1].

different nature, e.g. triangles and quadrilaterals. The fourth condition links the parameter h to the maximum diameter of the elements of the mesh $\mathcal{M}_h$. Finally, the fifth condition excludes very deformed or stretched elements, and hence the option of using *anisotropic* meshes (usually adopted in the context of fluid dynamics in the presence of boundary layers). Notice that, in the case of polygonal domains, the discretized domain

$$\Omega_h = \mathrm{int}\left(\bigcup_{i=1}^{v} \mathcal{E}_i \right)$$

coincides with Ω. If Ω is not polygonal, it is necessary to approximate portions of $\partial\Omega$ by line segments or simple curves. Here we will not discuss the approximation of a non-polygonal domain with a finite element grid, shown in Fig. 3.2. Hence, from now on the symbol Ω will be adopted to denote without distinction both the domain of interest and its approximation.

3.2.2 Lagrangian finite elements

In general, a *finite element* can be formally defined by the triple $(\mathcal{E}, \mathcal{P}, \Sigma)$ with the following properties:

- $\mathcal{E}$ is a polyhedron in $\mathbb{R}^d$. In the one-dimensional case it is an interval, in the two-dimensional case it is generally a triangle but it can also be a quadrilateral; in the three-dimensional case it can be a tetrahedron, a prism or a hexahedron;
- $\mathcal{P}$ is a space of n_e polynomials defined on $\mathcal{E}$. Functions in $\mathcal{P}$ are called *shape functions* if they form a basis of $\mathcal{P}$;

- $\Sigma = \{\zeta_i : \mathcal{P} \to \mathbb{R}\}_{i=1}^{n_e}$ is a set of linearly independent functionals on $\mathcal{P}$, satisfying $\zeta_i(\phi_j) = \delta_{ij}$, δ_{ij} being the Kronecker delta. Every polynomial $f(\mathbf{p}) \in \mathcal{P}$ is uniquely defined by the values of the n_e functionals in Σ, since they allow a unique identification of the coefficients $\{x_j\}_{j=1}^{n_e}$ of the expansion of $f(\mathbf{p})$ with respect to the chosen basis, i.e. $f(\mathbf{p}) = \sum_{j=1}^{n_e} x_j \phi_j(\mathbf{p})$. As a matter of fact, we have $x_i = \zeta_i(f)$, $i = 1, \ldots, n_e$. These coefficients are called *degrees of freedom*, or *nodal variables* of the finite element, since they are the values that must be assigned to uniquely define the field within the element.

In the case of *Lagrangian finite elements*, the chosen basis is provided by the Lagrange polynomials and the degree of freedom x_i is equal to the value taken by the polynomial f at a point $\mathbf{p}_i$ of $\mathcal{E}$ (*node* of the element), that is we have $x_i = f(\mathbf{p}_i)$, $i = 1, \ldots, n_e$. In the remainder, we will exclusively refer to the case of Lagrange finite elements. In the construction of a Lagrange finite element, the choice of nodes is not arbitrary. Indeed, the problem of interpolation on a given set $\mathcal{E}$ may be ill-posed. For this reason the following definition [1] turns out to be useful:

Definition 1. *A set Σ is said to be unisolvent on $\mathcal{P}$ if, given n_e arbitrary scalars x_j, $j = 1, \ldots, n_e$, there exists a unique function $f \in \mathcal{P}$ such that*

$$f(\mathbf{p}_j) = x_j, \quad j = 1, \ldots, n_e. \tag{3.19}$$

In such case the triple $(\mathcal{E}, \mathcal{P}, \Sigma)$ is called *Lagrangian finite element.*

3.3 Selection of the approximating subspace

Originally, the Galerkin method was meant to produce accurate approximate solutions from conveniently chosen subspaces of low dimension. However, the key idea behind the finite-element approximation is to make the computations efficient, even in the case of a high-dimensional approximate subspace. As it will be explained in the following section, this can be achieved by selecting a special type of approximating subspace. The finite element method is indeed the Galerkin method with a subspace of *piecewise polynomial* functions (see Fig. 3.3). An approximating subspace consisting of piecewise polynomial functions is convenient due to the ease of integration and differentiation of the polynomials (main requirements for computing the mass and stiffness matrices), and the fact that functions of this type lead naturally to sparse matrices, allowing the problem to be efficiently solved.

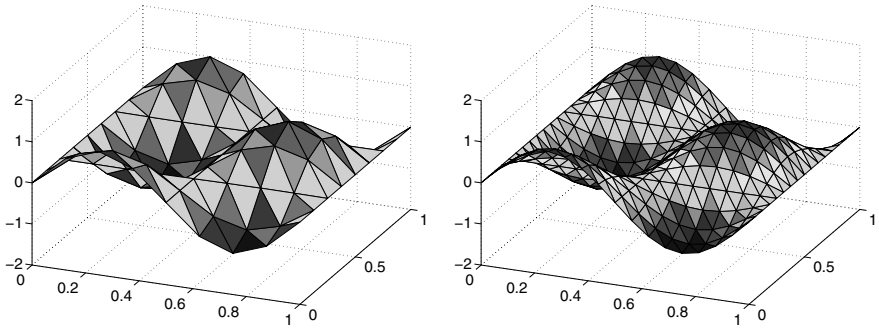


Figure 3.3: Finite-element approximation resulting from using piecewise linear (left) and piecewise quadratic (right) elements [1].

3.3.1 Piecewise linear functions on a triangular mesh

A piecewise polynomial is a function defined by a polynomial on each element of the mesh $\mathcal{M}_h$ defined over the domain Ω. In particular, let us consider a two-dimensional problem where $\mathbf{p} = [\xi, \eta]^T \in \mathbb{R}^2$ is the position vector in a Cartesian coordinate system with $\xi-$ and $\eta-$axes. We denote the space of polynomials with degree lower than or equal to $r \in \mathbb{N}$ as

$$\mathbb{P}_r = \left\{ f(\mathbf{p}) = \sum_{i+j \leq r} a_{ij} \xi^i \eta^j, \quad a_{ij} \in \mathbb{R} \right\}. \tag{3.20}$$

Thus, the spaces of piecewise polynomials of degree 1 or 2 (linear or quadratic) take the following form:

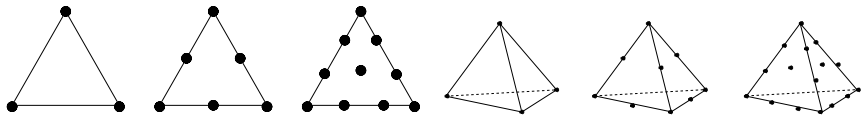
$$\begin{aligned} \mathbb{P}_1 &= \{ f(\mathbf{p}) = a + b\xi + c\eta, \quad a, b, c \in \mathbb{R} \} \\ \mathbb{P}_2 &= \{ f(\mathbf{p}) = a + b\xi + c\eta + d\xi\eta + e\xi^2 + g\eta^2, \quad a, b, c, d, e, g \in \mathbb{R} \}. \end{aligned} \tag{3.21}$$

It is possible to write the dimension of such spaces (in 2D) as

$$\dim \mathbb{P}_r = \frac{1}{2}(r+1)(r+2). \tag{3.22}$$

Let $\mathcal{M}_h$ be a suitable triangulation of the domain $\Omega \in \mathbb{R}^2$, then the following family of spaces can be constructed:

$$\mathcal{P}_h^r = \{ w \in C(\Omega) : w_{\mathcal{E}} \in \mathbb{P}_r, \ \forall \mathcal{E} \in \mathcal{M}_h \}, \qquad r \in \mathbb{N} \tag{3.23}$$



(a) (b)

Figure 3.4: Nodes for linear (left), quadratic (center) and cubic (right) polynomials on a triangular (a) and on a tetrahedral (b) element [1].

which will be referred to as the *space of finite elements*, i.e. the space of globally continuous functions that are polynomials $w_{\mathcal{E}}$ of degree r on the single element $\mathcal{E}$ of the mesh $\mathcal{M}_h$. Moreover, we introduce

$$\mathcal{P}^r_{h0} = \{w \in \mathcal{P}^r_h : \ w = 0 \quad \text{on } \partial\Omega\}, \qquad r \in \mathbb{N}. \tag{3.24}$$

Note that spaces $\mathcal{P}^r_h$ and $\mathcal{P}^r_{h0}$ are all subspaces of $H^1(\Omega)$ and, respectively, $H^1_0(\Omega)$. Hence, they represent possible choices for the space V_h in the Galerkin approximation (3.4), provided that the boundary conditions are properly incorporated. This results from the following property (we refer the reader to [38] for the proof).

Theorem 3. *A sufficient condition for a function w to belong to $H^1(\Omega)$ is that $w \in C(\overline{\Omega})$ and $w \in H^1(\mathcal{E})$, $\forall \mathcal{E} \in \mathcal{M}_h$.*

□

The fact that the functions of $\mathcal{P}^r_h$ are locally (element-wise) polynomials will make the computation of the load vector, of the mass and of the stiffness matrices substantially easier. Once the approximating subspace of finite elements has been selected, a complete basis $\{\phi_i\}_{i=1}^{n_e}$ for the space $\mathcal{P}^r_h$ has to be chosen, where $n_e = \dim \mathbb{P}_r$. It is convenient, based on the comment in Section 3.1, that the support of the generic basis function ϕ_i has non-empty intersection only with the support of a negligible number of other functions of the basis. In such way, many elements of the mass and stiffness matrices will be null. Furthermore, it is also convenient to consider a *Lagrangian* basis, i.e.

$$\forall i = 1, 2, ..., n \qquad \phi_i \in \mathcal{P}^r_h : \quad \phi_i(\mathbf{q}_j) = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{3.25}$$

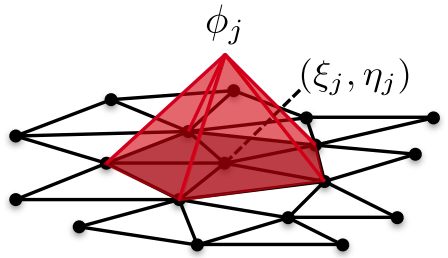


Figure 3.5: Basis function ϕ_j of $\mathcal{P}_h^1$ and its support.

where $\mathbf{q}_j$ are the *nodes* which in general form a superset of the vertices of the mesh $\mathcal{M}_h$. In other words, the coefficients of the expansion of a generic function $w \in \mathcal{P}_h^r$ in the basis will be the values taken by w at nodes $\mathbf{q}_j$. To clarify this, let us consider the simplest space $\mathcal{P}_h^1$ of continuous piecewise polynomials, which consists of continuous piecewise linear functions defined on a domain discretization (triangulation) $\mathcal{M}_h$ of a polygonal $\Omega \in \mathbb{R}^2$. A piecewise linear function w reduces to a first-degree polynomial $w_{\mathcal{E}} = a + b\xi + c\eta$ (3.21) on each triangular element $\mathcal{E} \in \mathcal{M}_h$. From (3.22) we have $\dim \mathbb{P}_1 = 3$, so that in each element the associated basis function is completely defined once we assign its values at the three *nodes* $\mathbf{q}_j$, $j = 1, 2, 3$, corresponding to the $n_e = 3$ vertices (*nodal values* of the piecewise linear function) of the triangle (see Fig. 3.4). Thus, if $\mathcal{M}_h$ has n vertices, then the space $\mathcal{P}_h^1$ is a finite-dimensional vector space with dimension n.

The approximation accuracy can be increased by introducing more nodes on each element and using polynomial shape functions of higher order. $\mathcal{P}_h^2$ is the space of piecewise quadratic polynomials with $\dim \mathbb{P}_2 = 6$. Such polynomials are determined once the values they take at six distinct points of each element are fixed. In this case, the degrees of freedom of $\mathcal{P}_h^2$ are the values taken at the vertices and at the midpoints of each edge (see Fig. 3.4).

In practice, we usually consider linear finite elements characterized by *shape functions* forming a *Lagrangian basis* (3.25) such that

$$\forall i, j = 1, 2, ..., n \quad \phi_i \in \mathcal{P}_h^1 : \quad \phi_i(\mathbf{p}_i) = 1, \quad \phi_i(\mathbf{p}_j) = 0, \quad j \neq i \qquad (3.26)$$

where $\{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}$ are the so-called *nodes* of the mesh. In this case the above basis is also referred to as *nodal basis*.

In a two-dimensional field estimation example, the unknown field $x(\xi, \eta, t)$ will be therefore approximated inside each Lagrangian linear element by a

first-degree polynomial $w_{\mathcal{E}} = a + b\,\xi + c\,\eta$. The coefficients a, b, c can be easily expressed in terms of the *nodal values* $x_1(t), x_2(t), x_3(t)$, so that the unknown field can be rewritten as an expansion of the form (3.6)

$$x(\xi, \eta, t) = \sum_{j=1}^{3} x_j(t)\, \phi_j(\xi, \eta)$$

where the functions ϕ_j are the so-called *basis* or *shape functions*, defined by (3.26). It is evident that the support of the generic shape function ϕ_j consists of only the triangles sharing node $\mathbf{p}_j$. This leads to sparse mass and stiffness matrices, whose non-zero elements correspond to the nodes of the mesh belonging to the same triangle. In the case of linear finite elements, once a node $\mathbf{p}_j$ has been fixed, then the basis function ϕ_j is characterized by a typical *hat* shape, as shown in Fig. 3.5. When we use, instead, continuous piecewise quadratic functions defined on a triangular mesh, the shape functions take the form illustrated in Fig. .

3.3.2 Piecewise linear functions in Galerkin method

Let us consider second-order partial differential equations of the form (2.16)

$$\frac{\partial x}{\partial t} - \lambda \nabla^2 x + \mathbf{v} \cdot \nabla x + g x = f \quad \text{in } \Omega \tag{3.27}$$

The aim is to show how an approximating subspace of piecewise linear functions can be constructed and used in the Galerkin method so as to provide a finite element approximation with linear Lagrange triangles of the original weak problem. As previously presented in Chapter 2, the weak form and the solution subspace change with the specific boundary conditions under consideration. In particular, in this section problems involving *natural* and *essential* boundary conditions will be separately discussed.

First of all, we consider two-dimensional problems with Neumann or Robin (i.e. *natural*) boundary conditions. It is possible to write this general *natural* condition as

$$\alpha \frac{\partial x}{\partial \mathbf{n}} + \beta x = \gamma \tag{3.28}$$

where $\alpha > 0$, $\beta \geq 0$ and $\gamma \in \mathbb{R}$. In this case, the weak problem takes the form (2.38). In order to apply the Galerkin method, a subspace V_h of V is needed. As described in Section 3.3.1, $\mathcal{P}_h^1$ can be chosen as the approximating subspace. By choosing the test functions in (2.38) as Lagrangian basis

functions and by using the expansion (3.6), the following Galerkin problem is obtained

$$\sum_{i=1}^{n} \dot{x}_i \int_\Omega \phi_i \phi_j \, d\mathbf{p} + \sum_{i=1}^{n} x_i \int_\Omega \alpha \nabla\phi_i \nabla\phi_j \, d\mathbf{p} + \sum_{i=1}^{n} x_i \int_{\partial\Omega} \beta \phi_i \phi_j \, d\mathbf{p} +$$
$$+ \sum_{i=1}^{n} x_i \int_\Omega \mathbf{v} \cdot \nabla\phi_i \phi_j \, d\mathbf{p} + \sum_{i=1}^{n} x_i \int_\Omega g \phi_i \phi_j \, d\mathbf{p} = \int_\Omega f \phi_j \, d\mathbf{p} + \int_{\partial\Omega} \gamma \phi_j \, d\mathbf{p}$$

$\forall j = 1, 2, ..., n$, which can be rewritten as

$$\sum_{i=1}^{n} m_{ij} \dot{x}_i + \sum_{i=1}^{n} (s_{ij}^{\alpha} + s_{ij}^{\beta} + s_{ij}^{v} + s_{ij}^{g}) x_i = u_j^f + u_j^\gamma, \quad \forall j = 1, 2, ..., n \tag{3.29}$$

where the following definitions have been used:

$$\begin{aligned} m_{ij} = \int_\Omega \phi_i \phi_j \, d\mathbf{p} \quad s_{ij}^{\alpha} &= \int_\Omega \alpha \nabla\phi_i \nabla\phi_j \, d\mathbf{p} \quad s_{ij}^{\beta} = \int_{\partial\Omega} \beta \phi_i \phi_j \, d\mathbf{p} \\ s_{ij}^{v} &= \int_\Omega \mathbf{v} \cdot \nabla\phi_i \phi_j \, d\mathbf{p} \quad s_{ij}^{g} = \int_\Omega g \phi_i \phi_j \, d\mathbf{p} \\ u_j^f &= \int_\Omega f \phi_j \, d\mathbf{p} \qquad\quad u_j^\gamma = \int_{\partial\Omega} \gamma \phi_j \, d\mathbf{p} \end{aligned} \tag{3.30}$$

Note that (3.29) can be written in matrix form, by definition of the following vectors and matrices:

$$\begin{aligned} \mathbf{x} &= [x_1, x_2, ..., x_n]^T \in \mathbb{R}^n, \\ \mathbf{M} &= \{m_{ij}\}_{i,j=1}^n \in \mathbb{R}^{n \times n}, \\ \mathbf{S}_\alpha &= \{s_{ij}^{\alpha}\}_{i,j=1}^n \in \mathbb{R}^{n \times n}, \\ \mathbf{S}_\beta &= \{s_{ij}^{\beta}\}_{i,j=1}^n \in \mathbb{R}^{n \times n}, \\ \mathbf{S}_v &= \{s_{ij}^{v}\}_{i,j=1}^n \in \mathbb{R}^{n \times n}, \\ \mathbf{S}_g &= \{s_{ij}^{g}\}_{i,j=1}^n \in \mathbb{R}^{n \times n}, \\ \mathbf{u}_f &= [u_1^f, u_2^f, ..., u_n^f]^T \in \mathbb{R}^n, \\ \mathbf{u}_\gamma &= [u_1^\gamma, u_2^\gamma, ..., u_n^\gamma]^T \in \mathbb{R}^n. \end{aligned}$$

The above leads to a linear system of ordinary differential equations of the form

$$\mathbf{M}\dot{\mathbf{x}} + (\mathbf{S}_\alpha + \mathbf{S}_\beta + \mathbf{S}_v + \mathbf{S}_g)\mathbf{x} = \mathbf{u}_f + \mathbf{u}_\gamma. \tag{3.31}$$

It is easy to see that defining $\mathbf{S} = \mathbf{S}_\alpha + \mathbf{S}_\beta + \mathbf{S}_v + \mathbf{S}_g$ and $\mathbf{u} = \mathbf{u}_f + \mathbf{u}_\gamma$, then the system (3.31) reduces to (3.10). Here the stiffness matrix $\mathbf{S}$ and the load vector $\mathbf{u}$ include additional terms arising from the natural boundary conditions. More specifically:

- the inhomogeneous Robin condition ($\beta > 0, \gamma \neq 0$) generates two additional terms $\mathbf{S}_\beta \neq \mathbf{0}$ and $\mathbf{u}_\gamma \neq \mathbf{0}$;
- the homogeneous Robin condition ($\beta > 0, \gamma = 0$) gives rise to an additional term $\mathbf{S}_\beta \neq \mathbf{0}$ in the stiffness matrix;
- the inhomogeneous Neumann condition ($\beta = 0, \gamma \neq 0$) produces $\mathbf{u}_\gamma \neq \mathbf{0}$ in the load vector;
- the homogeneous Neumann condition ($\beta = \gamma = 0$) does not generate any additional term in $\mathbf{S}$ and $\mathbf{u}$.

Next, we consider *essential* boundary conditions. In the case, for instance, of homogeneous Dirichlet conditions the weak form is described by (2.6) and $x \in H_0^1(\Omega)$. The linear finite element approximation can be applied by subdividing the overall domain into a mesh of elements on which the approximating subspace $\mathcal{P}_{h0}^1$ is defined. Differently from the case of natural conditions, in Dirichlet problems the field on the boundary is known, as it is imposed by the essential boundary condition. The Galerkin problem can, therefore, be solved only for the *internal* nodes of the domain. To this end, the set of nodes is divided into two subsets $\{1, 2, ..., N\}$ and $\{N+1, N+2, ..., n\}$ of *internal* and, respectively, *boundary* nodes. Then, the Galerkin approximation takes the form

$$\sum_{i=1}^{N} \dot{x}_i \int_\Omega \phi_i \phi_j \, d\mathbf{p} + \sum_{i=1}^{N} x_i \int_\Omega \alpha \nabla \phi_i \nabla \phi_j \, d\mathbf{p} + \sum_{i=1}^{N} x_i \int_\Omega \mathbf{v} \cdot \nabla \phi_i \phi_j \, d\mathbf{p} + \\ + \sum_{i=1}^{N} x_i \int_\Omega g \phi_i \phi_j \, d\mathbf{p} = \int_\Omega f \phi_j \, d\mathbf{p}, \qquad \forall j = 1, ..., N$$

that can be rewritten as

$$\sum_{i=1}^{N} m_{ij} \dot{x}_i + \sum_{i=1}^{N} (s_{ij}^\alpha + s_{ij}^v + s_{ij}^g) x_i = u_j^f, \qquad \forall j = 1, ..., N \tag{3.32}$$

where definitions (3.30) have been used. Next, we define

$$\begin{aligned}
\mathbf{x} &= [x_1, x_2, ..., x_N]^T \in \mathbb{R}^N, \\
\mathbf{M} &= \{m_{ij}\}_{i,j=1}^N \in \mathbb{R}^{N\times N}, \\
\mathbf{S}_\alpha &= \{s_{ij}^\alpha\}_{i,j=1}^N \in \mathbb{R}^{N\times N}, \\
\mathbf{S}_v &= \{s_{ij}^v\}_{i,j=1}^N \in \mathbb{R}^{N\times N}, \\
\mathbf{S}_g &= \{s_{ij}^g\}_{i,j=1}^N \in \mathbb{R}^{N\times N}, \\
\mathbf{u} &= [u_1, u_2, ..., u_N]^T \in \mathbb{R}^N.
\end{aligned}$$

and thus we can rewrite (3.32) in matrix form as follows

$$\mathbf{M}\dot{\mathbf{x}} + (\mathbf{S}_\alpha + \mathbf{S}_v + \mathbf{S}_g)\,\mathbf{x} = \mathbf{u}. \tag{3.33}$$

If the stiffness matrix is defined as $\mathbf{S} = \mathbf{S}_\alpha + \mathbf{S}_v + \mathbf{S}_g$, then (3.33) reduces to (3.10). However, differently from the problems with natural boundary conditions, the above system has lower dimension $N < n$, that is the number of *internal* nodes of the mesh.

Next, we are interested in extending the linear finite element approximation to the case of inhomogeneous Dirichlet initial-boundary value problems characterized by $x = \mu \neq 0$ on $\partial\Omega$. As introduced in Section 2.3.2, the associated weak form (2.30) of the problem is an integral form in the new unknown $z \in H_0^1(\Omega)$, $z = x - \widehat{\mu}$, where $\widehat{\mu} \in H^1(\Omega)$ is a function such that $\widehat{\mu} = \mu$ on $\partial\Omega$. The approximating subspace is the space of linear finite elements $\mathcal{P}_{h0}^1$. The approximate field can be written as the sum of the terms relative to *internal* and, respectively, *boundary* nodes as follows

$$x_h = \sum_{i=1}^{N} x_i\phi_i + \sum_{i=N+1}^{n} \mu_i\phi_i, \tag{3.34}$$

where $\mu_i = \mu_i(t) = \mu(\mathbf{p}_i, t)$ for $i = N+1, ..., n$, are the fixed nodal values on the boundary, specified by the inhomogeneous Dirichlet condition. Thus, the Galerkin approximation leads to

$$\begin{aligned}
&\sum_{i=1}^{N} \dot{x}_i \int_\Omega \phi_i\phi_j\, d\mathbf{p} + \sum_{i=1}^{N} x_i \int_\Omega \alpha\nabla\phi_i\nabla\phi_j\, d\mathbf{p} + \sum_{i=1}^{N} x_i \int_\Omega \mathbf{v}\cdot\nabla\phi_i\phi_j\, d\mathbf{p} + \\
&\quad + \sum_{i=1}^{N} x_i \int_\Omega g\phi_i\phi_j\, d\mathbf{p} = \int_\Omega f\phi_j\, d\mathbf{p} - \sum_{i=N+1}^{n} \dot{\mu}_i \int_\Omega \phi_i\phi_j\, d\mathbf{p} - \\
&\quad - \sum_{i=N+1}^{n} \mu_i \left(\int_\Omega \alpha\nabla\phi_i\nabla\phi_j\, d\mathbf{p} + \int_\Omega \mathbf{v}\cdot\nabla\phi_i\phi_j\, d\mathbf{p} + \int_\Omega g\phi_i\phi_j\, d\mathbf{p}\right)
\end{aligned}$$

which can be rewritten in the following compact form

$$\sum_{i=1}^{N}[m_{ij}\dot{x}_i+(s_{ij}^{\alpha}+s_{ij}^{v}+s_{ij}^{g})x_i] = u_j^f - \sum_{i=N+1}^{n}[m_{ij}\dot{\mu}_i+(s_{ij}^{\alpha}+s_{ij}^{v}+s_{ij}^{g})\mu_i]. \quad (3.35)$$

It can be observed that on the right-hand side of (3.35) now there are terms depending on the values of the field at boundary nodes $\mathbf{p}_i$, $i \in \{N+1, N+2, ..., n\}$. Compared to the homogeneous case, here the following additional quantities are introduced:

$$\begin{aligned}
\boldsymbol{\mu} &= [\mu_{N+1}, ..., \mu_n]^T \in \mathbb{R}^{n-N}, \\
\widehat{\mathbf{M}} &= \{m_{ij}\}_{i,j} \in \mathbb{R}^{N\times(n-N)}, \\
\widehat{\mathbf{S}}_{\alpha} &= \{s_{ij}^{\alpha}\}_{i,j} \in \mathbb{R}^{N\times(n-N)}, \\
\widehat{\mathbf{S}}_{v} &= \{s_{ij}^{v}\}_{i,j} \in \mathbb{R}^{N\times(n-N)}, \\
\widehat{\mathbf{S}}_{g} &= \{s_{ij}^{g}\}_{i,j} \in \mathbb{R}^{N\times(n-N)}, \\
\mathbf{u}_f &= [u_1^f, u_2^f, ..., u_N^f]^T \in \mathbb{R}^N.
\end{aligned}$$

Hence, it is possible to rewrite (3.35) in the following matrix form

$$\mathbf{M}\dot{\mathbf{x}} + (\mathbf{S}_{\alpha} + \mathbf{S}_v + \mathbf{S}_g)\mathbf{x} = \mathbf{u}_f - [\ \widehat{\mathbf{M}}\ \dot{\boldsymbol{\mu}} + (\widehat{\mathbf{S}}_{\alpha} + \widehat{\mathbf{S}}_v + \widehat{\mathbf{S}}_g)\ \boldsymbol{\mu}\ ] \quad (3.36)$$

and by defining $\mathbf{S} = \mathbf{S}_{\alpha} + \mathbf{S}_v + \mathbf{S}_g$ and $\widehat{\mathbf{S}} = \widehat{\mathbf{S}}_{\alpha} + \widehat{\mathbf{S}}_v + \widehat{\mathbf{S}}_g$, we obtain

$$\mathbf{M}\dot{\mathbf{x}} + \mathbf{S}\mathbf{x} = \mathbf{u}_f - (\ \widehat{\mathbf{M}}\ \dot{\boldsymbol{\mu}} + \widehat{\mathbf{S}}\ \boldsymbol{\mu}). \quad (3.37)$$

Hence, the inhomogeneous Dirichlet boundary conditions produce an additional term $\mathbf{u}_{\mu} = \widehat{\mathbf{M}}\ \dot{\boldsymbol{\mu}} + \widehat{\mathbf{S}}\ \boldsymbol{\mu}$ on the right-hand side of (3.37). It it worth pointing out that if we define $\mathbf{u} = \mathbf{u}_f - \mathbf{u}_{\mu}$, then, as in the homogeneous case, (3.37) leads to a lower-dimensional linear system of equations with dimension equal to the number of *internal* nodes N.

Finally, let us derive the approximate semi-discrete equations for initial-boundary value problems with mixed boundary conditions, i.e. problems with *natural* conditions on a portion $\partial\Omega_R$ of the overall domain Ω, and *essential* conditions on the remaining boundary $\partial\Omega_D$. In Section 2.6 we described the weak formulation of mixed IBVPs which essentially consists of dividing the integral form over Ω into two integral terms over $\partial\Omega_R$ and $\partial\Omega_D$. For the finite element approximation we choose a basis in the space $\mathcal{P}_{h0}^1$ so that the following semi-discrete system is obtained

$$\mathbf{M}\dot{\mathbf{x}} + (\mathbf{S}_{\alpha} + \mathbf{S}_{\beta} + \mathbf{S}_v + \mathbf{S}_g)\mathbf{x} = \mathbf{u}_f + \mathbf{u}_{\gamma} - \mathbf{u}_{\mu}. \quad (3.38)$$

As expected, the resulting linear differential equation (3.38) includes all the terms accounting for the different types of mixed natural/essential boundary conditions. In particular:

- the terms $\mathbf{S}_\beta$ and $\mathbf{u}_\gamma$ originate from the integral over $\partial\Omega_R$ to which a *natural* condition is applied;
- the vector $\mathbf{u}_\mu$ originates from the *essential* inhomogeneous condition imposed on $\partial\Omega_D$.

Note that system (3.38) has dimension given by the sum of the number of nodes on $\partial\Omega_R$ and the number of *internal* nodes.

3.4 Finite-element programming

In this section we present the derivation of the element matrices and vectors for a given approximating subspace. This is a key step for the practical implementation of the finite element approximation of spatially distributed systems governed by PDEs, since it will be shown how to locally (i.e. for each element) compute the mass and the stiffness matrices as well as the load vector. Indeed, these are the necessary components for the actual programming of the finite element approximation.

As seen in the previous section, the elements of such matrices are integral terms either computed over the domain Ω or over the boundary $\partial\Omega$. Due to the fact that the triangulation of Ω entirely covers the polygonal domain, and exploiting the additivity of the integration operation, it turns out that all the integrals involved in the global semi-discrete system can be decomposed into several integrals defined over each element $\mathcal{E}$ of the resulting mesh $\mathcal{M}_h$. To clarify this, let us consider for instance the mass matrix $\mathbf{M}$, which can be written as

$$\mathbf{M} = \int_\Omega \boldsymbol{\phi}\boldsymbol{\phi}^T \, d\mathbf{p} = \sum_{\mathcal{E}\in\mathcal{M}_h} \int_{\mathcal{E}} \boldsymbol{\phi}\boldsymbol{\phi}^T \, d\mathbf{p} = \sum_{\mathcal{E}\in\mathcal{M}_h} \mathbf{M}_{\mathcal{E}}$$

where we defined $\mathbf{M}_{\mathcal{E}} = \int_{\mathcal{E}} \boldsymbol{\phi}\boldsymbol{\phi}^T \, d\mathbf{p}$. Thus, the idea is to compute all the *local* matrices $\mathbf{M}_{\mathcal{E}}$ for each element $\mathcal{E}$ instead of the *global* matrix $\mathbf{M}$. In practice, the computational complexity can be significantly reduced by using Lagrangian finite elements which make the elements of $\boldsymbol{\phi}\boldsymbol{\phi}^T$ in $\mathbf{M}_{\mathcal{E}}$ nonnull only for those nodes belonging to the same triangle $\mathcal{E}$. The computation will,

therefore, be carried out element-by-element and only for those nodes in the same triangle. To this end, we proceed according to the following steps:

- after the discretization of Ω via the triangulation $\mathcal{M}_h$, a *global* numbering scheme is established on each node $\{1, 2, ..., n\}$ of the mesh;
- in each element $\mathcal{E} \in \mathcal{M}_h$ a *local* (e.g., counterclockwise) numbering $\{1, 2, 3\}$ is assigned to each node of element $\mathcal{E}$;
- a map is created to associate *local* and *global* numbering.

Thus, we start by computing the local vectors and matrices for each triangle. These are subsequently inserted into the global ones as elements whose specific location is given by the corresponding global numbering. Such procedure is the so-called *assembly* step of the finite element method.

For a mixed problem of type (3.38), the following quantitites are locally defined for each triangular element:

$$\begin{aligned} \boldsymbol{\phi}^e &= [\phi_1^e \ , \ \phi_2^e \ , \ \phi_3^e]^T \in \mathbb{R}^3, \\ \mathbf{S}^e &= \mathbf{S}_\alpha^e + \mathbf{S}_\beta^e + \mathbf{S}_v^e + \mathbf{S}_g^e \in \mathbb{R}^{3\times 3}, \\ \mathbf{u}^e &= \mathbf{u}_f^e + \mathbf{u}_\gamma^e - \mathbf{u}_\mu^e \in \mathbb{R}^3, \end{aligned}$$

and $\mathbf{M}^e \in \mathbb{R}^{3\times 3}$. The vector $\boldsymbol{\phi}^e$ is formed by the shape functions of the three nodes of the triangle, matrices $\mathbf{M}^e$ and $\mathbf{S}^e$ are the local mass and, respectively, stiffness matrix, while $\mathbf{u}^e$ represents the local load vector. Note that, the terms $\mathbf{S}_\beta^e$, $\mathbf{u}_\gamma^e$, $\mathbf{u}_\mu^e$ originating from the boundary conditions appear in the local definitions of the stiffness matrix and the load vector. Clearly, those terms are calculated only for those elements with at least one edge on the boundary $\partial\Omega$. A typical finite element program consists of the following steps:

1. *Pre-processing*: this step consists of setting up the problem and coding its computational domain, which, as seen in Section 3.2.1, requires the construction of the mesh. In general, the generation of an adequate mesh is a numerical problem of considerable interest for which ad hoc techniques have been developed. It is usually performed by dedicated programs or modules of FEM solvers.

2. *Local processing*: the core processing is the local computation of the mass matrix, the stiffness matrix and the load vector for each element of the mesh.

3. *Assembly*: the local quantitites are grouped together for the element-by-element construction of the global mass, stiffness, and forcing terms.

3.4.1 Shape functions

Let $\mathcal{E} \in \mathcal{M}_h$ be a triangular element of the mesh with vertices

$$\mathbf{p}_1 = [\xi_1, \eta_1]^T, \qquad \mathbf{p}_2 = [\xi_2, \eta_2]^T, \qquad \mathbf{p}_3 = [\xi_3, \eta_3]^T. \tag{3.39}$$

We assume first-order finite elements, such that the shape functions[1] $\{\phi_i\}_{i=1}^3$ take the form

$$\phi_i = a_i + b_i\ \xi + c_i\ \eta, \quad \forall i = 1, 2, 3, \quad \forall \mathbf{p} \in \mathcal{E}, \quad a_i, b_i, c_i \in \mathbb{R}. \tag{3.40}$$

We also assume a Lagrangian basis, so that the following conditions hold:

$$\begin{array}{rcllll} \phi_1(\mathbf{p}_1) & = & 1, & \phi_2(\mathbf{p}_1) = 0, & \phi_3(\mathbf{p}_1) = 0, \\ \phi_1(\mathbf{p}_2) & = & 0, & \phi_2(\mathbf{p}_2) = 1, & \phi_3(\mathbf{p}_2) = 0, \\ \phi_1(\mathbf{p}_3) & = & 0, & \phi_2(\mathbf{p}_3) = 0, & \phi_3(\mathbf{p}_3) = 1. \end{array} \tag{3.41}$$

Thanks to the above conditions, we can easily determine the coefficients a_i, b_i, c_i which allow the computation of the shape functions. For instance, the function ϕ_1 is obtained by solving the following linear system of equations:

$$\left\{ \begin{array}{l} \phi_1(\mathbf{p}_1) = 1 \\ \phi_1(\mathbf{p}_2) = 0 \\ \phi_1(\mathbf{p}_3) = 0 \end{array} \right. \qquad \left\{ \begin{array}{l} a_1 + b_1\ \xi_1 + c_1\ \eta_1 = 1 \\ a_1 + b_1\ \xi_2 + c_1\ \eta_2 = 0 \\ a_1 + b_1\ \xi_3 + c_1\ \eta_3 = 0 \end{array} \right. \qquad \left\{ \begin{array}{l} a_1 = \frac{\xi_2\eta_3 - \xi_3\eta_2}{2\mathrm{A}} \\ b_1 = \frac{\eta_2 - \eta_3}{2\mathrm{A}} \\ c_1 = \frac{\xi_3 - \xi_2}{2\mathrm{A}} \end{array} \right. ,$$

where A denotes the area of element $\mathcal{E}$ (positive for a counter-clockwise numbering of vertices), obtained as follows

$$\mathrm{A} = \frac{1}{2} \det \begin{bmatrix} 1 & \xi_1 & \eta_1 \\ 1 & \xi_2 & \eta_2 \\ 1 & \xi_3 & \eta_3 \end{bmatrix} = \frac{1}{2}\ (\xi_2\eta_3 - \xi_3\eta_2 + \xi_1\eta_2 - \xi_1\eta_3 + \xi_3\eta_1 - \xi_2\eta_1).$$

In an analogous way, we can calculate the coefficients of the other shape functions. The general expressions are

$$a_i = \frac{\xi_j\eta_k - \xi_k\eta_j}{2\mathrm{A}} \qquad b_i = \frac{\eta_j - \eta_k}{2\mathrm{A}} \qquad c_i = \frac{\xi_k - \xi_j}{2\mathrm{A}}, \tag{3.42}$$

[1] Since we carry out an element-by-element computation of the shape functions, to simplify the notation, from now on the local shape functions will be denoted by ϕ_1, ϕ_2, ϕ_3, i.e. the superscript "e" will be omitted. Thus, a local numbering is assumed.

where the subscripts $i, j, k = 1, 2, 3$ must follow a suitable permutation, that is $(i, j, k) \in \{(1, 2, 3), (2, 3, 1), (3, 1, 2)\}$.

3.4.2 Local mass matrix

Once defined the local shape functions $\boldsymbol{\phi} = [\phi_1 \ \phi_2 \ \phi_3]^T$ obtained from (3.42), the mass matrix can be written as

$$\mathbf{M}^e = \int_{\mathcal{E}} \boldsymbol{\phi}\boldsymbol{\phi}^T \, d\mathbf{p} = \int_{\mathcal{E}} \begin{bmatrix} \phi_1^2 & \phi_1\phi_2 & \phi_1\phi_3 \\ \phi_2\phi_1 & \phi_2^2 & \phi_2\phi_3 \\ \phi_3\phi_1 & \phi_3\phi_2 & \phi_3^2 \end{bmatrix} d\mathbf{p} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}.$$

Notice that, as anticipated in Theorem 2, the mass matrix is symmetric. In order to compute $\mathbf{M}^e$ in 2D, we need to introduce the following exact integration formula (see, e.g., [39] for the general expression):

Theorem 4 (**Eisenberg**, **Malvern**). *Let $\mathcal{E}$ be a triangular element of the mesh $\mathcal{M}_h$ generated over the domain Ω, and let ϕ_i and ϕ_j, $i, j = 1, 2, 3$, be two local shape functions forming a Lagrangian basis in the space of linear finite elements. Then:*

$$\int_{\mathcal{E}} \phi_i^l \ \phi_j^m \, d\boldsymbol{p} = 2\mathrm{A}\frac{l! \ m!}{(l + m + 2)!} \qquad \forall i, j = 1, 2, 3, \qquad \forall l, m \geq 0, \qquad (3.43)$$

where A *denotes the area of element $\mathcal{E}$.*

□

The above expression can be used to obtain $\forall i, j = 1, 2, 3$

$$\begin{aligned} m_{ii} &= \int_{\mathcal{E}} \phi_i^2 \, d\mathbf{p} = 2\mathrm{A}\frac{2! \ 0!}{(2 + 0 + 2)!} = \frac{\mathrm{A}}{6} \\ m_{ij} &= \int_{\mathcal{E}} \phi_i \ \phi_j \, d\mathbf{p} = 2\mathrm{A}\frac{1! \ 1!}{(1 + 1 + 2)!} = \frac{\mathrm{A}}{12} \qquad i \neq j \end{aligned}$$

so that the mass matrix takes the following form

$$\mathbf{M}^e = \frac{\mathrm{A}}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}. \qquad (3.44)$$

It can be observed that, as anticipated in Theorem 1, $\mathbf{M}^e$ turns out to be definite positive.

3.4.3 Local stiffness matrix

Suppose α, g and $\mathbf{v}$ uniform, i.e. constant in space. The diffusive term of the PDE (3.27) affects the stiffness matrix through $\mathbf{S}_\alpha^e$, which is given by

$$\mathbf{S}_\alpha^e = \int_{\mathcal{E}} \alpha \nabla \boldsymbol{\phi}^T \nabla \boldsymbol{\phi} \, d\mathbf{p} = \alpha \int_{\mathcal{E}} \nabla \boldsymbol{\phi}^T \nabla \boldsymbol{\phi} \, d\mathbf{p},$$

where the matrix $\nabla \boldsymbol{\phi} \in \mathbb{R}^{2\times 3}$ has columns equal to the gradients of the three local shape functions. Assuming first-order finite elements, then $\nabla \boldsymbol{\phi}$ turns out to be constant. Indeed, recalling that the coefficients of the local shape functions are given by (3.42), we obtain

$$\nabla \boldsymbol{\phi} = [\nabla \phi_1, \ \nabla \phi_2, \ \nabla \phi_3] = \begin{bmatrix} \frac{\partial \phi_1}{\partial \xi} & \frac{\partial \phi_2}{\partial \xi} & \frac{\partial \phi_3}{\partial \xi} \\ \frac{\partial \phi_1}{\partial \eta} & \frac{\partial \phi_2}{\partial \eta} & \frac{\partial \phi_3}{\partial \eta} \end{bmatrix} = \begin{bmatrix} b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix}.$$

and thus $\nabla \boldsymbol{\phi}^T \nabla \boldsymbol{\phi}$ can be taken out of the integral. This means that

$$\mathbf{S}_\alpha^e = \alpha \ \nabla \boldsymbol{\phi}^T \nabla \boldsymbol{\phi} \int_{\mathcal{E}} d\mathbf{p} = \alpha \ \nabla \boldsymbol{\phi}^T \nabla \boldsymbol{\phi} \ \mathrm{A}$$

and finally

$$\mathbf{S}_\alpha^e = \alpha \mathrm{A} \begin{bmatrix} b_1^2 + c_1^2 & b_1 b_2 + c_1 c_2 & b_1 b_3 + c_1 c_3 \\ b_1 b_2 + c_1 c_2 & b_2^2 + c_2^2 & b_2 b_3 + c_2 c_3 \\ b_1 b_3 + c_1 c_3 & b_2 b_3 + c_2 c_3 & b_3^2 + c_3^2 \end{bmatrix}. \tag{3.45}$$

It can be noticed that (3.45) is a symmetric and positive definite matrix. The reaction term in the PDE (3.27) affects the stiffness matrix through $\mathbf{S}_g^e$, that can be written as

$$\mathbf{S}_g^e = \int_{\mathcal{E}} g \ \boldsymbol{\phi} \boldsymbol{\phi}^T \, d\mathbf{p} = g \ \int_{\mathcal{E}} \boldsymbol{\phi} \boldsymbol{\phi}^T \, d\mathbf{p} = g \ \mathbf{M}^e.$$

and thus takes the form

$$\mathbf{S}_g^e = \frac{g\mathrm{A}}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}. \tag{3.46}$$

Note that (3.46) is also symmetric and positive definite.

Next, the aim is to compute $\mathbf{S}_v^e$, originating from the advection term in (2.4). We assume a uniform transport vector $\mathbf{v} = [v_\xi, v_\eta]^T$, whose components represent a transport process along the axes of the Cartesian coordinate

system. This matrix takes the following expression

$$\mathbf{S}_v^e = \int_{\mathcal{E}} \boldsymbol{\phi}\ \mathbf{v}^T \nabla \boldsymbol{\phi}\, d\mathbf{p} = \int_{\mathcal{E}} \boldsymbol{\phi}\, d\mathbf{p}\ \mathbf{v}^T \nabla \boldsymbol{\phi}.$$

The integrals can be computed by substituting $l = 1$ and $m = 0$ in (3.43) so that

$$\int_K \phi_i\, d\mathbf{p} = 2\mathrm{A}\frac{1!\ 0!}{(1+0+2)!} = \frac{\mathrm{A}}{3}, \qquad \forall i = 1,2,3$$

and thus

$$\mathbf{S}_v^e = \frac{\mathcal{A}}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}\begin{bmatrix} v_\xi & v_\eta \end{bmatrix}\begin{bmatrix} b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix}.$$

The local stiffness matrix accounts for the advection term of a parabolic PDE of type (3.27) through the following matrix:

$$\mathbf{S}_v^e = \frac{\mathcal{A}}{3}\begin{bmatrix} v_\xi b_1 + v_\eta c_1 & v_\xi b_2 + v_\eta c_2 & v_\xi b_3 + v_\eta c_3 \\ v_\xi b_1 + v_\eta c_1 & v_\xi b_2 + v_\eta c_2 & v_\xi b_3 + v_\eta c_3 \\ v_\xi b_1 + v_\eta c_1 & v_\xi b_2 + v_\eta c_2 & v_\xi b_3 + v_\eta c_3 \end{bmatrix}. \tag{3.47}$$

It is worth mentioning that, as discussed in Theorem 2, the stiffness matrix, in general, is not symmetric. This is due to the fact that (3.47) is not symmetric, and consequently if the advection term is present, then also the total stiffness matrix will not be symmetric. However, it will be positive definite, as proved in Theorem 1.

In the presence of Robin boundary conditions, $\mathbf{S}_\beta^e$ is given by

$$\mathbf{S}_\beta^e = \int_{\partial\mathcal{E}} \beta\boldsymbol{\phi}\boldsymbol{\phi}^T\, d\mathbf{p} = \beta\int_{\partial\mathcal{E}} \boldsymbol{\phi}\boldsymbol{\phi}^T\, d\mathbf{p} = \beta\int_{\partial\mathcal{E}} \begin{bmatrix} \phi_1^2 & \phi_1\phi_2 & \phi_1\phi_3 \\ \phi_2\phi_1 & \phi_2^2 & \phi_2\phi_3 \\ \phi_3\phi_1 & \phi_3\phi_2 & \phi_3^2 \end{bmatrix} d\mathbf{p}$$

where $\partial\mathcal{E}$ denotes the edge of the triangle to which the natural condition is applied. For instance, let us assume that a Robin condition is imposed on the edge $\partial\mathcal{E}_{12}$, i.e. the edge between nodes 1 and 2 of the element. In order to determine $\mathbf{S}_\beta^e$, we need to compute the above line integrals. To this end, we introduce the following parametrization of the segment corresponding to $\partial\mathcal{E}_{12}$:

$$\mathbf{p} = \mathbf{p}_1 + \sigma(\mathbf{p}_2 - \mathbf{p}_1) \implies \begin{cases} \xi = \xi_1 + \sigma(\xi_2 - \xi_1) \\ \eta = \eta_1 + \sigma(\eta_2 - \eta_1) \end{cases}, \qquad \sigma \in [0,1]$$

$$\mathbf{p}'(\sigma) = \frac{d\mathbf{p}}{d\sigma} = \mathbf{p}_2 - \mathbf{p}_1 \implies \|\mathbf{p}'(\sigma)\| = \sqrt{(\mathbf{p}_2 - \mathbf{p}_1)^T(\mathbf{p}_2 - \mathbf{p}_1)} \triangleq \ell \tag{3.48}$$

where we denoted by ℓ the length of $\partial\mathcal{E}_{12}$. Since only Lagrangian finite elements are considered, the shape function ϕ_3 is such that $\phi_3 = 0$ along $\partial\mathcal{E}_{12}$, and thus $\mathbf{S}^e_\beta$ can be determined by simply computing the integrals involving ϕ_1 and ϕ_2. In particular, by exploiting the definition of line integral and by using (3.41), the elements of $\mathbf{S}^e_\beta$ can be computed as

$$\begin{aligned}
&\beta \int_{\partial\mathcal{E}_{12}} \phi_1^2 \, d\mathbf{p} \\
= \;& \beta \int_{\partial\mathcal{E}_{12}} (a_1 + b_1\xi + c_1\eta)^2 \, d\mathbf{p} \\
= \;& \beta \int_0^1 [a_1 + b_1\xi_1 + \sigma(b_1\xi_2 - b_1\xi_1) + c_1\eta_1 + \sigma(c_1\eta_2 - c_1\eta_1)]^2 \; \ell \, d\sigma \\
= \;& \beta\ell \int_0^1 [(a_1 + b_1\xi_1 + c_1\eta_1) - \sigma(b_1\xi_1 + c_1\eta_1 - b_1\xi_2 - c_1\eta_2)]^2 \, d\sigma \\
= \;& \beta\ell \int_0^1 [\phi_1(\mathbf{p}_1) - \sigma(\phi_1(\mathbf{p}_1) - \phi_1(\mathbf{p}_2))]^2 \, d\sigma = \beta\ell \int_0^1 (1-\sigma)^2 \, d\sigma = \frac{\beta\ell}{3};
\end{aligned}$$

$$\begin{aligned}
&\beta \int_{\partial\mathcal{E}_{12}} \phi_1\phi_2 \, d\mathbf{p} \\
= \;& \beta \int_{\partial\mathcal{E}_{12}} (a_1 + b_1\xi + c_1\eta)(a_2 + b_2\xi + c_2\eta) \, d\mathbf{p} \\
= \;& \beta \int_0^1 [a_1 + b_1\xi_1 + \sigma(b_1\xi_2 - b_1\xi_1) + c_1\eta_1 + \sigma(c_1\eta_2 - c_1\eta_1)] \cdot \\
& \cdot [a_2 + b_2\xi_1 + \sigma(b_2\xi_2 - b_2\xi_1) + c_2\eta_1 + \sigma(c_2\eta_2 - c_2\eta_1)] \; \ell \, d\sigma \\
= \;& \beta\ell \int_0^1 [\phi_1(\mathbf{p}_1) - \sigma(\phi_1(\mathbf{p}_1) - \phi_1(\mathbf{p}_2))][\phi_2(\mathbf{p}_1) + \sigma(\phi_2(\mathbf{p}_2) - \phi_2(\mathbf{p}_1))] \, d\sigma \\
= \;& \beta\ell \int_0^1 (1-\sigma)\sigma \, d\sigma = \beta\ell \int_0^1 (\sigma - \sigma^2) \, d\sigma = \frac{\beta\ell}{6}.
\end{aligned}$$

The remaining integrals can be computed in a similar way. To sum up, the term in the local stiffness matrix accounting for a Robin boundary condition takes the form

$$\mathbf{S}^e_\beta = \frac{\beta\ell}{6} \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.49}$$

whenever the Robin condition is applied on the edge $\partial\mathcal{E}_{12}$. In the case of a

Robin condition on $\partial\mathcal{E}_{23}$, one has

$$\mathbf{S}^e_\beta = \frac{\beta\ell}{6}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}; \tag{3.50}$$

finally, if the Robin condition is applied on the edge $\partial\mathcal{E}_{13}$ we obtain

$$\mathbf{S}^e_\beta = \frac{\beta\ell}{6}\begin{bmatrix} 2 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 2 \end{bmatrix}. \tag{3.51}$$

Notice that, no matter what case (3.49)-(3.51) is considered, $\mathbf{S}^e_\beta$ turns out to be symmetric and positive definite. In conclusion, the local stiffness matrix is obtained as the sum of the terms (3.45), (3.46), (3.47), and one or two (depending on how many edges are affected by the boundary condition) terms among (3.49)-(3.51), i.e.

$$\mathbf{S}^e = \mathbf{S}^e_\alpha + \mathbf{S}^e_\beta + \mathbf{S}^e_g + \mathbf{S}^e_v.$$

3.4.4 Local load vector

The aim is to determine the local load vector $\mathbf{u}^e$, arising from a second-order PDE of the form (3.27). The following two cases will be considered: i) f is a uniform forcing term, constant over the whole domain; ii) f is a point function concentrated in a single point of the domain. An exogeneous source f acts on the local load vector through $\mathbf{u}^e_f$. Let us start by considering the uniform case i), which leads to

$$\mathbf{u}^e_f = \int_{\mathcal{E}} f\boldsymbol{\phi}\, d\mathbf{p} = f\int_{\mathcal{E}} \boldsymbol{\phi}\, d\mathbf{p} = f\int_{\mathcal{E}} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} d\mathbf{p} \tag{3.52}$$

By using the Eisenberg–Malvern formula (3.43), (3.52) becomes

$$\mathbf{u}^e_f = \frac{f\mathrm{A}}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{3.53}$$

so that the forcing term f is equally shared by the three nodes of the element.

Next, consider the case ii) with a forcing term assumed to be concentrated in node 1 of element $\mathcal{E}$. Such source can be modeled as a spatial *Dirac delta*,

i.e. $f = f^\star\ \delta(\mathbf{p} - \mathbf{p}_1)$, where $f^\star \in \mathbb{R}$ is the *intensity* of the source, assumed uniform and constant. In this case, the load vector takes the form

$$\begin{aligned} \mathbf{u}_f^e &= \int_{\mathcal{E}} f^\star \delta(\mathbf{p} - \mathbf{p}_1) \boldsymbol{\phi}\, d\mathbf{p} = f^\star \int_{\mathcal{E}} \delta(\mathbf{p} - \mathbf{p}_1) \boldsymbol{\phi}\, d\mathbf{p} \\ &= f^\star \boldsymbol{\phi}(\mathbf{p}_1) = f^\star \begin{bmatrix} \phi_1(\mathbf{p}_1) \\ \phi_2(\mathbf{p}_1) \\ \phi_3(\mathbf{p}_1) \end{bmatrix}. \end{aligned}$$

Due to the fact that we use Lagrangian linear elements, conditions (3.41) hold, and thus the local load vector for a point source located at node 1 is given by

$$\mathbf{u}_f^e = f^\star \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \tag{3.54}$$

In an analogous way, $\mathbf{u}_f^e$ can be obtained for the other two cases of a point source located at nodes 2 or 3 of the element.

In the case of a point source located at a generic point $\mathbf{p}_0$ inside element $\mathcal{E}$, the value of the intensity of the source is shared among the three nodes of the element, based on the values that the local shape functions take at the source location $\mathbf{p}_0$. This results in

$$\mathbf{u}_f^e = f^\star \begin{bmatrix} \phi_1(\mathbf{p}_0) \\ \phi_2(\mathbf{p}_0) \\ \phi_3(\mathbf{p}_0) \end{bmatrix}. \tag{3.55}$$

Moreover, an inhomogeneous Robin or Neumann boundary condition originates an additional term of the form

$$\mathbf{u}_\gamma^e = \int_{\partial\mathcal{E}} \gamma \boldsymbol{\phi}\, d\mathbf{p} = \gamma \int_{\partial\mathcal{E}} \boldsymbol{\phi}\, d\mathbf{p} = \gamma \int_{\partial\mathcal{E}} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} d\mathbf{p} \tag{3.56}$$

where γ is assumed uniform. We can exploit the parametrization (3.48) in order to compute the line integrals. Following the same rationale used for

$\mathbf{S}^e_\beta$, we obtain

$$\begin{aligned}
&\gamma \int_{\partial\mathcal{E}_{12}} \phi_1 \, d\mathbf{p} \\
= \;& \gamma \int_{\partial\mathcal{E}_{12}} (a_1 + b_1\xi + c_1\eta) \, d\mathbf{p} \\
= \;& \gamma \int_0^1 [a_1 + b_1\xi_1 + \sigma(b_1\xi_2 - b_1\xi_1) + c_1\eta_1 + \sigma(c_1\eta_2 - c_1\eta_1)]\ell \, d\sigma \\
= \;& \gamma\ell \int_0^1 [(a_1 + b_1\xi_1 + c_1\eta_1) - \sigma(b_1\xi_1 + c_1\eta_1 - b_1\xi_2 - c_1\eta_2)] \, d\sigma \\
= \;& \gamma\ell \int_0^1 [\phi_1(\mathbf{p}_1) - \sigma(\phi_1(\mathbf{p}_1) - \phi_1(\mathbf{p}_2))] \, d\sigma = \gamma\ell \int_0^1 (1-\sigma) \, d\sigma = \frac{\gamma\ell}{2}.
\end{aligned}$$

$$\begin{aligned}
&\gamma \int_{\partial\mathcal{E}_{12}} \phi_2 \, d\mathbf{p} \\
= \;& \gamma \int_{\partial\mathcal{E}_{12}} (a_2 + b_2\xi + c_2\eta) \, d\mathbf{p} \\
= \;& \gamma \int_0^1 [a_2 + b_2\xi_1 + \sigma(b_2\xi_2 - b_2\xi_1) + c_2\eta_1 + \sigma(c_2\eta_2 - c_2\eta_1)]\ell \, d\sigma \\
= \;& \gamma\ell \int_0^1 [(a_2 + b_2\xi_1 + c_2\eta_1) + \sigma(b_2\xi_2 + c_2\eta_2 - b_2\xi_1 - c_2\eta_1)] \, d\sigma \\
= \;& \gamma\ell \int_0^1 [\phi_2(\mathbf{p}_1) + \sigma(\phi_2(\mathbf{p}_2) - \phi_2(\mathbf{p}_1))] \, d\sigma = \gamma\ell \int_0^1 \sigma \, d\sigma = \frac{\gamma\ell}{2}.
\end{aligned}$$

Thus, if the *natural* boundary condition is applied on $\partial\mathcal{E}_{12}$, one has

$$\mathbf{u}^e_\gamma = \frac{\gamma\ell}{2} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}. \tag{3.57}$$

Analogously, for the case of a *natural* boundary condition applied on $\partial\mathcal{E}_{23}$

$$\mathbf{u}^e_\gamma = \frac{\gamma\ell}{2} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \tag{3.58}$$

while, if is on $\partial\mathcal{E}_{13}$, we obtain

$$\mathbf{u}^e_\gamma = \frac{\gamma\ell}{2} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}. \tag{3.59}$$

Finally, as discussed in Section 3.3.2, the presence of inhomogeneous Dirichlet conditions generates an additional term in the load vector on the right-hand side of (3.37) of the form

$$\mathbf{u}_\mu^e = \widehat{\mathbf{M}}^e \dot{\boldsymbol{\mu}}^e + \widehat{\mathbf{S}}^e \boldsymbol{\mu}^e \tag{3.60}$$

where $\boldsymbol{\mu}^e$ represents the vector of Dirichlet values imposed on the boundary nodes. Matrices $\widehat{\mathbf{M}}^e$ and $\widehat{\mathbf{S}}^e$ take into account the interaction between nodes under *essential* boundary conditions and the remaining (unassigned) nodes. It is useful to recall that in the case of Dirichlet conditions, the semi-discrete system (3.37) has dimension given by the number N of nodes on which the essential condition is not applied. Based on N, $\boldsymbol{\mu}^e$ takes two possible forms: i) if the *essential* condition is applied on an edge of a triangle that coincides with, e.g., $\partial\mathcal{E}_{23}$, then we have $N = 1$ and thus $u_\mu^e \in \mathbb{R}$, with $\boldsymbol{\mu}^e = [\mu_2, \mu_3]^T = [\mu(\mathbf{p}_2, t), \mu(\mathbf{p}_3, t)]^T \in \mathbb{R}^2$. In this case $\widehat{\mathbf{M}}^e, \widehat{\mathbf{S}}^e \in \mathbb{R}^{1\times 2}$ and

$$u_\mu^e = \begin{bmatrix} m_{12} & m_{13} \end{bmatrix} \dot{\boldsymbol{\mu}}^e + \begin{bmatrix} s_{12} & s_{13} \end{bmatrix} \boldsymbol{\mu}^e = m_{12}\dot{\mu}_2 + m_{13}\dot{\mu}_3 + s_{12}\mu_2 + s_{13}\mu_3.$$

In addition, if the Dirichlet condition $x = \mu(\mathbf{p}, t)$ is assumed uniform and constant along the whole boundary $\partial\mathcal{E}_{23}$, then $\mu_2 = \mu_3 = \mu$, and hence $\dot{\mu}_2 = \dot{\mu}_3 = 0$. Finally, we can rewrite

$$u_\mu^e = \alpha\, \mu\, \mathrm{A}\, (b_1 b_2 + b_1 b_3 + c_1 c_2 + c_1 c_3);$$

ii) If, instead, there is only a single node of element $\mathcal{E}$ on $\partial\mathcal{E}_{23}$, for instance node 3, then $\mu_3 \in \mathbb{R}$, $\mathbf{u}_\mu^e \in \mathbb{R}^2$, and $\widehat{\mathbf{M}}^e, \widehat{\mathbf{S}}^e \in \mathbb{R}^{2\times 1}$. In this case one has

$$\mathbf{u}_\mu^e = \begin{bmatrix} m_{13} \\ m_{23} \end{bmatrix} \dot{\mu}_3 + \begin{bmatrix} s_{13} \\ s_{23} \end{bmatrix} \mu_3 = \begin{bmatrix} m_{13}\dot{\mu}_3 + s_{13}\mu_3 \\ m_{23}\dot{\mu}_3 + s_{23}\mu_3 \end{bmatrix}.$$

Assuming a uniform and constant Dirichlet condition $x = \mu(\mathbf{p}, t)$ along the whole boundary, then we can write

$$\mathbf{u}_\mu^e = \alpha\, \mu\, \mathrm{A} \begin{bmatrix} b_1 b_3 + c_1 c_3 \\ b_2 b_3 + c_2 c_3 \end{bmatrix}.$$

To conclude, the local load vector $\mathbf{u}^e$ is given by the superposition of $\mathbf{u}_f^e$ chosen among (3.53)-(3.55) and $\mathbf{u}_\gamma^e$ (3.57)-(3.59). Then the effect of (3.60) must be subtracted, so as to obtain

$$\mathbf{u}^e = \mathbf{u}_f^e + \mathbf{u}_\gamma^e - \mathbf{u}_\mu^e.$$

3.4.5 Assembly of the global matrices

Assembling global matrices starting from the local ones is straightforward for first-order elements. All that is required is a look-up table connecting local and global numbering schemes [36]. For higher order elements that procedure remains unchanged apart from the increased number of nodes of each element. Besides local matrices, boundary conditions also contribute to the assembly of the global matrices. As seen in Section 2.2, Dirichlet-type boundary conditions must be explicitly enforced. Neumann boundary conditions, on the other hand, constitute natural boundary conditions and need not be explicitly enforced.

As already noted, global matrices are sparse, hence it is advisable to seek storage methods that are able to minimize memory requirements. Two main methods are suitable to achieve memory savings: sparse matrix storage and banded matrix storage methods. The former approach can be used regardless of the global node numbering adopted and it is particularly convenient when the percentage of nonzero elements is really low. The latter requires the usage of a smart global numbering to effectively reduce the global matrix bandwidth.

It is important to note that the number of nonzero entries in a row i of a global FE matrix is equal to the number of nodes directly connected to node i, that is, the number of the nodes belonging to the elements which share node i. Hence, the global FE matrix has a sparse structure reflecting the fact that a local approximation is used for the exact unknown function. This is one of the most attractive features of FE approximation since it allows a significant reduction of memory storage requirements as well as CPU time reduction.

3.5 Time discretization

The above finite-element approximation leads to a finite-dimensional continuous-time linear system (3.10). In order to obtain a discrete-time system, several time discretization schemes are available. For instance, the standard $\theta-$method can be adopted by discretizing the time derivative with a simple difference quotient and replacing the other terms with a linear combination of the values at time k and time $k+1$ depending on the real

parameter θ, $0 \leq \theta \leq 1$, i.e.

$$\mathbf{M}\frac{\mathbf{x}_{k+1}-\mathbf{x}_k}{\Delta}+\mathbf{S}(\theta\mathbf{x}_{k+1}+(1-\theta)\mathbf{x}_k)=\theta\mathbf{u}_{k+1}+(1-\theta)\mathbf{u}_k, \tag{3.61}$$

where $\Delta = t_{k+1} - t_k, k = 0, 1, \ldots$ denotes the discretization step, assumed to be constant. Then, the following methods can be obtained [1]:

- $\theta = 0$ leads to the *forward (explicit) Euler* method

$$\mathbf{M}\frac{\mathbf{x}_{k+1}-\mathbf{x}_k}{\Delta}+\mathbf{S}\mathbf{x}_k=\mathbf{u}_k \tag{3.62}$$

  Using the above approximation it is possible to obtain the following discrete-time system

$$\mathbf{x}_{k+1}=\mathbf{A}\mathbf{x}_k+\mathbf{B}\mathbf{u}_k \tag{3.63}$$

  with $\mathbf{A} = \mathbf{I} + \Delta\mathbf{M}^{-1}\mathbf{S}$, $\mathbf{B} = \mathbf{M}^{-1}\Delta$, $\mathbf{u}_k \triangleq \mathbf{u}(k\Delta)$, $\mathbf{x}_k \triangleq \mathbf{x}(k\Delta) = col\{x_j(k\Delta)\}_{j=1}^n$.

- $\theta = 1$ leads to the *backward (implicit) Euler* method

$$\mathbf{M}\frac{\mathbf{x}_{k+1}-\mathbf{x}_k}{\Delta}+\mathbf{S}\mathbf{x}_{k+1}=\mathbf{u}_{k+1} \tag{3.64}$$

  Using (3.64) we obtain the discrete-time system (3.63) with $\mathbf{A} = \left(\mathbf{I}+\Delta\mathbf{M}^{-1}\mathbf{S}\right)^{-1}$, $\mathbf{B} = \mathbf{A}\mathbf{M}^{-1}\Delta$, $\mathbf{u}_k \triangleq \mathbf{u}((k+1)\Delta)$, $\mathbf{x}_k \triangleq \mathbf{x}(k\Delta) = col\{x_j(k\Delta)\}_{j=1}^n$. Notice that $\mathbf{A}$ is well defined for any $\Delta > 0$ since both $\mathbf{M}$ and $\mathbf{S}$ are positive definite (Theorem 1).

- $\theta = 1/2$ leads to the *Crank-Nicolson (trapezoidal)* method

$$\mathbf{M}\frac{\mathbf{x}_{k+1}-\mathbf{x}_k}{\Delta}+\frac{1}{2}\mathbf{S}(\mathbf{x}_{k+1}+\mathbf{x}_k)=\frac{1}{2}(\mathbf{u}_{k+1}+\mathbf{u}_k)\,. \tag{3.65}$$

  which is accurate of second order with respect to Δ, while for $\theta = 0$ and $\theta = 1$ we obtain first-order methods.

In conclusion, by performing one of the above time-discretization methods on the semi-discrete descriptor system (3.10), a discrete-time explicit system of the form (3.63) can be obtained. The latter state-space model will be used in the next chapters for the design of suitable field estimators.

Chapter 4

Centralized and distributed design of field estimators

4.1 Introduction

The recent breakthrough of wireless sensor network technology has made possible to cost-effectively monitor spatially distributed systems via deployment of multiple sensors over the area of interest. This clearly paves the way for several important practical monitoring applications concerning, e.g., weather forecasting [2], water flow regulation [11], fire detection, diffusion of pollutants [3], smart grids [9], vehicular traffic [10]. The problem of fusing data from different sensors can be accomplished either in a *centralized* way, i.e. when there is a single fusion center collecting data from all sensors and taking care of the overall spatial domain of interest, or in *distributed* (decentralized) fashion with multiple intercommunicating fusion centers (nodes) each of which can only access part of the sensor data and take care of a sub-region of the overall domain. The decentralized approach is preferable in terms of scalability of computation with the problem size and will be, therefore, undertaken.

Since spatially distributed processes are usually modeled as infinite–dimensional systems, governed by *partial differential equations* (see Chapter 2), distributed state estimation for such systems turns out to be a key issue to be addressed. While a lot of work has dealt with distributed filters for finite-dimensional, both linear [40–43] and nonlinear [44], systems as well as for multitarget tracking [45], considerably less attention has been devoted to

the more difficult case of distributed-parameter systems.

Recent work [14–19] has addressed the design of distributed state estimators/observers for large-scale systems formed by the sparse interconnection of many subsystems (compartments). Such systems are possibly (but not necessarily) originated from spatial discretization of PDEs. In particular, [14] presents a fully scalable distributed Kalman filter based on a suitable spatial decomposition of a complex large-scale system as well as on appropriate observation fusion techniques among the local Kalman filters. In [15], non-scalable consensus-based multi-agent estimators are proposed wherein each agent aims to estimate the state of the whole large-scale system. In [16], a moving-horizon partition-based approach is followed in order to estimate the state of a large-scale interconnected system and decentralization is achieved via suitable approximations of covariances. Further, [17] deals with dynamic field estimation by wireless sensor networks with special emphasis on sensor scheduling for trading off communication/energy efficiency versus estimation performance. In [18], a single-time scale distributed estimator of dynamic random fields is proposed, where the sensing time scale coincide with the consensus time scale. Finally, in [19] the design of distributed continuous-time observers for partitioned linear systems is addressed.

As for the specific case of distributed-parameter systems, interesting contributions have been provided in [27, 28] which present consensus filters wherein each node of the network aims to estimate the system state on the whole spatial domain of interest.

As compared to [27,28], here a different strategy will be adopted in which each node is only responsible for estimating the state over a sub-domain of the overall domain. This setup allows for a solution which is *scalable* with respect to the spatial domain (i.e., the computational complexity in each node does not depend on the size of the whole spatial domain but only of its region of competence). In this context, the contributions of this chapter are summarized as follows:

- We develop *scalable* distributed filters for distributed-parameter systems by suitably adapting the so-called Schwarz decomposition methods [46–51], which allow to split the overall domain into smaller subdomains and assign each of them to different interconnected processing nodes.

- We exploit the *finite element* (FE) method [35, 36, 52] in order to approximate the original infinite-dimensional filtering problem into a,

possibly large-scale, finite-dimensional one. Combining these two ingredients, we propose a novel distributed *finite element Kalman filter* which generalizes to the more challenging distributed case previous work on FE Kalman filtering [53, 54].

- We show that the parallel FE-based implementation of the Schwarz method on the overall system is equivalent to performing a novel time-discretization scheme on the interconnected subsystems. Furthermore, we verify the well-posedness of the proposed discretization method in terms of numerical stability (i.e., in terms of boundedness and convergence of the time-discretization errors).

- We provide results on the stability of the proposed distributed FE Kalman filter. Last but not least, a practical procedure, which requires the tuning of only one (or few) scalar parameters, is provided to check and guarantee the stability property.

Preliminary ideas on the topic can be found in [55]. Further, all the results of this chapter are reported in [56].

The rest of the chapter is structured as follows. Section 4.2 introduces the basic notation and problem formulation. Then Section 4.3 presents the centralized FE Kalman filter for distributed-parameter systems. Section 4.4 shows how to extend such a filter to the distributed setting by means of the parallel Schwarz method and analyzes the numerical stability in terms of boundedness and convergence of the discretization errors. Then, Section 4.5 provides results on the exponential stability of the proposed distributed FE Kalman filter while Section 4.6 demonstrates its effectiveness via numerical examples related to the estimation of a bi-dimensional temperature field. Finally, Section 4.7 ends the chapter with concluding remarks and perspectives for future work.

4.2 Problem formulation

This chapter addresses the estimation of a scalar, time-and-space-dependent, field from given discrete, in both time and space, measurements related to such a field provided by multiple sensors placed within the domain of interest. Let Ω be a bounded domain of a d-dimensional Euclidean space $\mathbb{R}^d$ with boundary $\partial\Omega$, where $d \in \{1, 2, 3\}$. The spatial coordinate vector is denoted

by $\mathbf{p} \in \Omega$. The scalar field to be estimated $x(\mathbf{p}, t)$ is defined over the space-time domain $\Omega \times \mathbb{R}_+$, as the solution of a *partial differential equation (PDE)* of the form (2.3)

$$\frac{\partial x}{\partial t} + \mathcal{L}(x) \;=\; f \tag{4.1}$$

with (possibly unknown) initial condition $x(\mathbf{p}, 0) = x_0(\mathbf{p})$, $\mathbf{p} \in \Omega$, and homogeneous boundary conditions (see Section 2.1.3)

$$\mathcal{B}(x) \;=\; 0 \text{ on } \partial\Omega\,. \tag{4.2}$$

The dynamic field is observed by a network of sensors $i \in \mathcal{S} \triangleq \{1, \ldots, S\}$ placed at the spatial locations $\mathbf{s}_i \in \Omega$, which provide the measurements

$$y_{q,i} \;=\; h_i\left(x\left(\mathbf{s}_i, t_q\right)\right) + v_{q,i} \tag{4.3}$$

collected at discrete sampling instants t_q, $q \in \mathcal{Z}_+ = \{1, 2, \ldots\}$, such that $0 < t_1 < t_2 < \cdots$. In (4.1)-(4.3): $f(\mathbf{p}, t)$ is a forcing term possibly affected by process noise; $h_i(\cdot)$ is the measurement function of sensor i; $v_{q,1}, \ldots, v_{q,S}$ are mutually independent white measurement noise sequences, also independent from the initial state $x_0(\mathbf{p}) = x(\mathbf{p}, 0)$ for any $\mathbf{p} \in \Omega$.

The aim is to design a decentralized Kalman filter for spatially distributed systems, i.e. to solve in a fully distributed fashion the infinite-dimensional filtering problem of estimating the state $x(\mathbf{p}, t)$ of system (4.1)-(4.2) given the locally gathered measurements (4.3). The proposed solution relies on (i) the FE method [35]- [36] for the approximation of the above problem into a finite-dimensional one, and (ii) a domain decomposition method for the subdivision of the system into interconnected subsystems with possibly overlapping states. The idea is to decompose the original problem on the whole domain of interest into estimation subproblems concerning smaller subdomains, and then assign such subproblems to different *nodes* which can locally process and exchange data in order to estimate their own state. This ensures scalability of the distributed filter for monitoring the target field. To this end, let us consider the set of nodes $\mathcal{N} = \{1, \ldots, N\}$, subdivide the domain Ω into possibly overlapping subdomains Ω_m, $m \in \mathcal{N}$, such that $\Omega = \bigcup_{m \in \mathcal{N}} \Omega_m$. Further, let $\mathbf{y}_q^m \triangleq col\left\{y_{q,i} : \mathbf{s}_i \in \Omega_m\right\}$ denote the vector of local measurements available to node m at time t_q. Then, the task of each node m is to estimate the field x over the corresponding subdomain Ω_m exploiting only the local measurements $\mathbf{y}_q^m$ and the information coming from the nodes associated to the neighboring subdomains.

Throughout the chapter, we make the following assumptions.

A1. $\mathcal{L}(\cdot)$ and $\mathcal{B}(\cdot)$ are linear operators over a suitable Hilbert space V, with $\mathcal{L}(\cdot)$ self-adjoint.

A2. Under the boundary conditions (4.2), the quadratic form $\int_\Omega \mathcal{L}(x)\, x\, d\mathbf{p}$ is bounded and coercive (i.e., positive definite).

A third and last assumption A3 on the properties of the local measurement function and local observability will be introduced in Section 4.5 (to which we refer for a formal definition of local observability and for a discussion of its implications).

An example of the above general problem is the estimation of the temperature field x over the spatial domain of interest given point measurements of temperature sensors. In this case, V is usually taken as the Sobolev space $H^1(\Omega)$, the measurement function is simply $h(x) = x$, while the PDE (4.1) reduces to the well known *heat equation* (2.9) introduced in Section 2.1.4, with $\mathcal{L}(x) = -\nabla \cdot (\lambda \nabla(x))$, $\mathcal{B}(x) = \alpha\, \partial x/\partial \mathbf{n} + \beta x$ and $\alpha(\mathbf{p})\beta(\mathbf{p}) \geq 0$, $\alpha(\mathbf{p}) + \beta(\mathbf{p}) > 0$, $\forall \mathbf{p} \in \partial\Omega$. Here $\lambda(\mathbf{p})$ is the *thermal diffusivity*, $\cdot$ stands for scalar product, $\nabla \triangleq \partial/\partial \mathbf{p}$ denotes the gradient operator, $\mathbf{n}$ is the outward pointing unit normal vector of the boundary $\partial\Omega$, and $\partial x/\partial \mathbf{n} = \nabla x \cdot \mathbf{n}$. Clearly, when the thermal diffusivity is space-independent, one has $\mathcal{L}(x) = -\lambda \nabla^2(x)$, where $\nabla^2 = \nabla \cdot \nabla$ is the Laplacian operator.

Notice that considering homogeneous boundary conditions as in (4.2) is not restrictive, since the inhomogeneous case $\mathcal{B}(x) = g$ on $\partial\Omega$ can be subsumed into the homogeneous one by means of the change of variables $z = x - \widehat{\mu}$, where $\widehat{\mu}$ is any function belonging to V and satisfying the inhomogeneous boundary conditions (see Section 2.3.2 for a detailed description of the inhomogeneous case).

4.3 Centralized finite-element Kalman filter

In this section, it is shown how to approximate the continuous-time infinite-dimensional system (4.1) into a discrete-time finite-dimensional linear dynamical system within the FE framework, and how, thanks to this space-

time discretization, a centralized filter for field estimation can be directly designed.

By subdividing the domain Ω into a suitable set of non overlapping regions, or elements, and by defining a suitable set of basis functions $\phi_j(\mathbf{p}) \in V\ (j = 1, \ldots, n)$ on them, it is possible to write the approximation (3.6) of the unknown function $x(\mathbf{p}, t)$ as

$$x(\mathbf{p}, t) \approx \sum_{j=1}^{n} \phi_j(\mathbf{p})\, x_j(t) \ = \ \boldsymbol{\phi}^T(\mathbf{p})\, \mathbf{x}(t) \tag{4.4}$$

where: $x_j(t)$ is the unknown expansion coefficient of function $x(\mathbf{p}, t)$ relative to time t and basis function $\phi_j(\mathbf{p})$; $\boldsymbol{\phi}(\mathbf{p}) \stackrel{\triangle}{=} col\{\phi_j(\mathbf{p})\}_{j=1}^{n}$ and $\mathbf{x}(t) \stackrel{\triangle}{=} col\{x_j(t)\}_{j=1}^{n}$.

As introduced in Chapter 3, the choices of the basis functions ϕ_j and of the elements are key points of the FE method. Typically, the elements define a FE mesh with vertices $\mathbf{p}_j \in \Omega, j = 1, \ldots, n$. Then each basis function ϕ_j is a piece-wise polynomial which vanishes outside the FEs around $\mathbf{p}_j$ and such that $\phi_j(\mathbf{p}_i) = \delta_{ij}$, δ_{ij} denoting the Kronecker delta.

In order to apply the Galerkin method presented in Section 3.1, let the PDE (4.1) be recast in the following weak form

$$\int_\Omega \frac{\partial x}{\partial t}\, \psi\, d\mathbf{p} + \int_\Omega \mathcal{L}(x)\, \psi\, d\mathbf{p} = \int_\Omega f\, \psi d\mathbf{p} \tag{4.5}$$

where $\psi(\mathbf{p})$ is a generic space-dependent weight function. Then, by choosing the test function $\psi(\mathbf{p})$ equal to the selected basis functions ϕ_j and exploiting the approximation (4.4) in (4.5), we get

$$\int_\Omega \frac{\partial x}{\partial t}\, \phi_j\, d\mathbf{p} + \int_\Omega \mathcal{L}(x)\, \phi_j\, d\mathbf{p} = \int_\Omega f\, \phi_j d\mathbf{p} \quad j = 1, \ldots, n.$$

Stacking (one on top of the other) the above scalar equations into a single vector equation, yields

$$\int_\Omega \boldsymbol{\phi}\, \frac{\partial}{\partial t}\left(\boldsymbol{\phi}^T \mathbf{x}\right)\, d\mathbf{p} + \int_\Omega \boldsymbol{\phi}\, \mathcal{L}\left(\boldsymbol{\phi}^T \mathbf{x}\right)\, d\mathbf{p} = \int_\Omega \boldsymbol{\phi} f\, d\mathbf{p}$$

from which, defining $\mathcal{L}(\boldsymbol{\phi}) \stackrel{\triangle}{=} col\left\{\mathcal{L}(\phi_j)\right\}_{j=1}^{n}$ and thanks to the linearity of

operator $\mathcal{L}(\cdot)$, the usual FE weak form (3.10) is obtained

$$\begin{aligned}&\underbrace{\left[\int_\Omega \boldsymbol{\phi}(\mathbf{p})\boldsymbol{\phi}^T(\mathbf{p})d\mathbf{p}\right]}_{\mathbf{M}}\dot{\mathbf{x}}(t)+\underbrace{\left[\int_\Omega \boldsymbol{\phi}(\mathbf{p})\left[\mathcal{L}\left(\boldsymbol{\phi}(\mathbf{p})\right)\right]^T d\mathbf{p}\right]}_{\mathbf{S}}\mathbf{x}(t)\\&=\underbrace{\int_\Omega \boldsymbol{\phi}(\mathbf{p})f(\mathbf{p},t)d\mathbf{p}}_{\mathbf{u}(t)}.\end{aligned}\tag{4.6}$$

It is evident how in (4.6) the *mass* matrix $\mathbf{M}$ and the stiffness matrix $\mathbf{S}$, defined in Section 3.1, depend only on basis functions and can be computed *a priori*. The third integral depends on the forcing term f, which is assumed to be known, and can hence be computed *a priori*, leading to a time dependent *load vector* $\mathbf{u}(t)$.

It is worth pointing out that, in the FE weak form (4.6), the boundary conditions (4.2) can be *essential* or *natural* (see Section 2.2). In both cases, the resulting linear differential equation takes the matrix form (3.10)

$$\mathbf{M}\,\dot{\mathbf{x}}+\mathbf{S}\,\mathbf{x}=\mathbf{u}+\boldsymbol{\epsilon}\tag{4.7}$$

where $\boldsymbol{\epsilon}$ arises from the approximation error[1] in the finite-dimensional representation (4.4) of x in terms of basis functions. Notice that from Theorem 1 $\mathbf{M}$ turns out to be positive definite by linear independence of the basis functions $\phi_j(\cdot)$, while $\mathbf{S}$ is positive definite as well thanks to the coercivity of the quadratic form in the left-hand side of (4.5) of assumption A2. Hence the system (4.7) turns out to be asymptotically stable, the state transition matrix $-\mathbf{M}^{-1}\mathbf{S}$ being well defined and strictly Hurwitz thanks to the positive definitess of $\mathbf{M}$ and $\mathbf{S}$. System (4.7) can be discretized in time by different methods, as discussed in Section 3.5, to provide the discrete-time state-space model

$$\mathbf{x}_{k+1}=\mathbf{A}\mathbf{x}_k+\mathbf{B}\mathbf{u}_k+\mathbf{w}_k\tag{4.8}$$

where the process noise $\mathbf{w}_k$ has been introduced to account for the various uncertainties and/or imprecisions (e.g. FE approximation, time discretization, and imprecise knowledge of boundary conditions). Specifically, the backward Euler method described in Section 3.5 (here adopted for stability

[1] If x is sufficiently smooth, then the FE approximation error is point-wise bounded and converges to zero as the size of the FE mesh tends to zero.

issues) leads to a marching in time FE implementation [52] which yields (4.8) with

$$\begin{aligned} \mathbf{A} &= \left(\mathbf{I} + \Delta \mathbf{M}^{-1}\mathbf{S}\right)^{-1} \\ \mathbf{B} &= \mathbf{A}\mathbf{M}^{-1}\Delta \\ \mathbf{u}_k &\stackrel{\triangle}{=} \mathbf{u}((k+1)\Delta) \\ \mathbf{x}_k &\stackrel{\triangle}{=} \mathbf{x}(k\Delta) = col\{x_j(k\Delta)\}_{j=1}^{n} \end{aligned}$$

where Δ denotes the time integration interval. As previously noticed in Section 3.5, $\mathbf{A}$ is well defined for any $\Delta > 0$ since $\mathbf{M}$ is positive definite. It is worth pointing out that, in the presented formulation, the *descriptor* (or *implicit*) system (4.7) has been transformed into the standard state-space system (4.8) characterized by full matrices $\mathbf{A}$ and $\mathbf{B}$ defined above. This means that the inherent property of sparsity, distinguishing matrices $\mathbf{M}$ and $\mathbf{S}$ in the original descriptor system, is unavoidably lost. In order to preserve sparsity and exploit the associated advantages in terms of computational efficiency, suitable linear state estimators for descriptor systems could be designed starting from model (4.7).

In the following, for the sake of notational simplicity, it will be assumed that each sampling instant is a multiple of Δ, i.e., $t_q = T_q\Delta$ with $T_q \in \mathbb{Z}_+$, and we let $\mathcal{T} = \{T_1, T_2, \ldots\}$; irregular sampling could, however, be easily dealt with. This amounts to assuming that the numerical integration rate of the PDE (4.1) in the filter can be higher than the measurement collection rate, which can be useful in order to reduce numerical errors. In a centralized setting where all sensor measurements are available to the filter, the measurement equation (4.3) takes the discrete-time form

$$\mathbf{y}_k = \mathbf{h}\left(\mathbf{x}_k\right) + \mathbf{v}_k \tag{4.9}$$

for any $k = T_q \in \mathcal{T}$, where

$$\begin{aligned} \mathbf{y}_k &\stackrel{\triangle}{=} col\left\{y_{q,i}\right\}_{i\in\mathcal{S}} \\ \mathbf{h}\left(\mathbf{x}\right) &\stackrel{\triangle}{=} col\left\{h_i\left(\boldsymbol{\phi}^T(\mathbf{s}_i)\mathbf{x}\right)\right\}_{i\in\mathcal{S}} \\ \mathbf{v}_k &\stackrel{\triangle}{=} col\left\{v_{q,i}\right\}_{i\in\mathcal{S}} \end{aligned}$$

In particular, in the case wherein all sensors directly measure the target field x, i.e. $h_i(x) = x$ for all $i \in \mathcal{S}$, the measurement equation (4.9) turns out to

be linear with $\mathbf{h}(\mathbf{x}) = \mathbf{C}\mathbf{x}$, where

$$\mathbf{C} = col\left\{\boldsymbol{\phi}^T(\mathbf{s}_i)\right\}_{i\in\mathcal{S}} \tag{4.10}$$

Summarizing, the original infinite-dimensional continuous-time problem has been reduced to a much simpler finite-dimensional (possibly large-scale) discrete time filtering problem (a linear one provided that all sensor measurement functions are linear) to which the *Kalman filter*, or *extended Kalman filter* when sensor nonlinearities are considered, can be readily applied. The resulting centralized filter recursion becomes:

$$\begin{aligned}
\hat{\mathbf{x}}_{k|k} &= \begin{cases} \hat{\mathbf{x}}_{k|k-1} + \mathbf{L}_k\left(\mathbf{y}_k - \mathbf{h}\left(\hat{\mathbf{x}}_{k|k-1}\right)\right) & \text{if } k\in\mathcal{T} \\ \hat{\mathbf{x}}_{k|k-1} & \text{otherwise} \end{cases} \\
\mathbf{P}_{k|k} &= \begin{cases} \mathbf{P}_{k|k-1} - \mathbf{L}_k\mathbf{C}_k^T\mathbf{P}_{k|k-1} & \text{if } k\in\mathcal{T} \\ \mathbf{P}_{k|k-1} & \text{otherwise} \end{cases} \\
\hat{\mathbf{x}}_{k+1|k} &= \mathbf{A}\hat{\mathbf{x}}_{k|k} + \mathbf{B}\mathbf{u}_k \\
\mathbf{P}_{k+1|k} &= \mathbf{A}\mathbf{P}_{k|k}\mathbf{A}^T + \mathbf{Q}_k
\end{aligned} \tag{4.11}$$

where

$$\begin{aligned}
\mathbf{C}_k &= \frac{\partial\mathbf{h}}{\partial\mathbf{x}}\left(\hat{\mathbf{x}}_{k|k-1}\right) \\
\mathbf{L}_k &= \mathbf{P}_{k|k-1}\mathbf{C}_k\left(\mathbf{R}_k + \mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}_k^T\right)^{-1}
\end{aligned} \tag{4.12}$$

for $k\in\mathcal{T}$. The recursion is initialized from suitable $\hat{\mathbf{x}}_{1|0}$ and $\mathbf{P}_{1|0} = \mathbf{P}_{1|0}^T > \mathbf{0}$. In (4.11), $\mathbf{Q}_k$ and $\mathbf{R}_k$ denote the covariance matrices of the process noise $\mathbf{w}_k$ and, respectively, measurement noise $\mathbf{v}_k$.

The following two remarks concerning optimality of the Kalman filter and, respectively, handling of sensor nonlinearities are in order.

Remark 1. *Notice that the process noise $\mathbf{w}_k$ in (4.8) arises from the superposition of several uncertainties and/or perturbations (including, e.g., the FE approximation of the continuous field) so that its whiteness and uncorrelation with the initial state, usually assumed in a stochastic framework, do not hold true in practice. As a result, the Kalman filter algorithm (4.11)-(4.12), even in the linear case $\mathbf{h}(\mathbf{x}) = \mathbf{C}\mathbf{x}$, looses its Bayes optimality but still preserves deterministic least-squares optimality as the minimizer of the*

following cost function

$$\begin{aligned} J &= \left(\mathbf{x}_1 - \hat{\mathbf{x}}_{1|0}\right)^T \mathbf{P}_{1|0}^{-1} \left(\mathbf{x}_1 - \hat{\mathbf{x}}_{1|0}\right) + \\ &\sum_{i=1}^{k-1} \left(\mathbf{x}_{i+1} - \mathbf{A}\mathbf{x}_i\right)^T \mathbf{Q}_i^{-1} \left(\mathbf{x}_{i+1} - \mathbf{A}\mathbf{x}_i\right) + \\ &\sum_{i=1}^{k} \left(\mathbf{y}_i - \mathbf{C}\mathbf{x}_i\right)^T \mathbf{R}_i^{-1} \left(\mathbf{y}_i - \mathbf{C}\mathbf{x}_i\right) \end{aligned}$$

Remark 2. *Sensor nonlinearities, provided that the measurement functions $h_i(\cdot)$ in (4.3) are invertible, can be handled by applying the inverse measurement functions $h_i^{-1}(\cdot)$ to the sensor outputs, i.e. by defining transformed sensor outputs $y'_{q,i} = h_i^{-1}\left(y_{q,i}\right)$ and considering the transformed linear measurement equations*

$$y'_{q,i} = x\left(\mathbf{s}_i, t_q\right) + v'_{q,i} \tag{4.13}$$

in place of (4.3). This approach has the advantage of eliminating any need for a nonlinear filter. However, while (4.13) is exact in the ideal, noiseless, case i.e. when $v_{q,i} = v'_{q,i} = 0$, it becomes only an approximation in presence of measurement noise. In particular, even if $v_{q,i}$ in (4.3) can be reasonably assumed to be zero-mean, white and uncorrelated with the state $x\left(\mathbf{s}_i, t_q\right)$, non-negligible biases and/or correlations can be induced by the nonlinear transformation $h_i^{-1}(\cdot)$ in the noise term $v'_{q,i}$ appearing in (4.13). For this reason, depending on the particular measurement function under consideration, the use of truly nonlinear filters can be useful also when the sensor nonlinearity is invertible. For non-invertible sensor nonlinearities, nonlinear filters such as, for instance, the extended Kalman filter for sufficiently smooth $h_i(\cdot)$ or the unscented Kalman filter for arbitrary $h_i(\cdot)$, must be used.

4.4 Distributed finite-element Kalman filter

In order to develop a scalable distributed filter for monitoring the target field, the idea is to run in each node $m \in \mathcal{N}$ a field estimator for the region Ω_m exploiting local measurements $\mathbf{y}_q^m$, information from the nodes assigned to neighboring subdomains, as well as the PDE model (4.1) properly discretized in time and space. The proposed approach takes inspiration from the parallel Schwarz method [46], [50], originally conceived for an iterative solution of boundary value problems. Subsequently, the Schwarz method has received renewed interest [47, 48] in connection with the parallelization

of PDE solvers. In loose terms, the idea of the *parallel Schwarz method* is to decompose the original PDE problem on the overall domain of interest into subproblems concerning smaller subdomains, and then to solve in parallel such subproblems via iterations in which previous solutions concerning neighboring subdomains are used as boundary conditions. As shown below, such an idea turns out to be especially useful for the distributed filtering problem considered in this work.

Let us define, for any $m \in \mathcal{N}$, a partition $\{\Gamma_{mj}\}_{j\in\mathcal{N}_m}$ of $\partial\Omega_m$ (the boundary of Ω_m) such that

$$\begin{aligned} &\Gamma_{mm} = \partial\Omega \cap \partial\Omega_m \\ &\partial\Omega_m = \bigcup_{j\in\mathcal{N}_m} \Gamma_{mj} \\ &\Gamma_{mj} \subset \Omega_j, \quad \forall j \neq m \\ &\Gamma_{mj} \cap \Gamma_{mh} = \emptyset, \quad \forall j \neq h \end{aligned} \tag{4.14}$$

In this way, each piece Γ_{mj} of $\partial\Omega_m$ for any $j \in \mathcal{N}_m\backslash\{m\}$ is uniquely assigned to node j. Notice that in the above definitions, for each node m, $\mathcal{N}_m$ indicates the in-neighborhood of node m, where j is called an in-neighbor of node m whenever $\Gamma_{mj} \neq \emptyset$ (by definition, $\mathcal{N}_m$ includes the node m). This clearly originates a directed network (graph) $\mathcal{G} = (\mathcal{N}, \mathfrak{L})$ with node set $\mathcal{N}$ and link set $\mathfrak{L} \triangleq \{(j,m) \in \mathcal{N}\times\mathcal{N} : \Gamma_{mj} \neq \emptyset\}$.

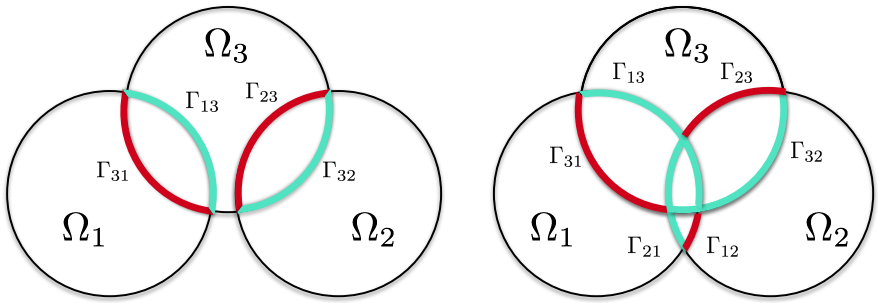


Figure 4.1: Definition of interfaces Γ_{mj} in two different configurations with three overlapping subdomains.

In order to describe the filtering cycle to be implemented in node m within the sampling interval $[t_q, t_{q+1})$, let us assume that at time t_q^-, before the acquisition of $\mathbf{y}_q^m$, such a node is provided with a prior estimate $\hat{x}_{q|q-1}^m$

as the result of the previous filtering cycles. Let δ be the time interval necessary for performing a *distributed prediction* step consisting of an information exchange between neighbors and a local field prediction over a subdomain. Then, $L_q \triangleq (t_{q+1} - t_q)/\delta$ represents the number of distributed prediction steps (equal to the number of allowed data exchanges) in the q-th sampling interval. Note that, for the sake of notational simplicity, hereafter it is supposed that $t_{q+1} - t_q$ is an integer multiple of δ, i.e., $L_q \in \mathbb{Z}_+$. Anyway, the method could easily encompass the general case. Then, the above mentioned filtering cycle for the proposed distributed estimation algorithm essentially consists of:

1. **Correction**, i.e. incorporation (assimilation) of the last measurement $\mathbf{y}_q^m$ into the current estimate;
2. **Distributed prediction**, i.e. alternate exchanges of estimates with the neighborhood $\mathcal{N}_m$ and predictions over the time sub-intervals $[t_q + (\ell - 1)\delta, t_q + \ell\delta]$ for $\ell = 1, \ldots, L_q$, i.e. L_q times.

The proposed *Parallel Schwarz* filter is detailed in Table 4.1.

Some remarks concerning the algorithm reported in Table 4.1 are in order. As it can be seen from step 5, the information received by neighboring nodes is taken into account by explicitly imposing the inhomogeneous Dirichlet interface conditions (4.16) on $\Gamma_{mj}, j \in \mathcal{N}_m \setminus \{m\}$. Clearly, a delay is introduced in those terms concerning neighboring nodes which makes the algorithm well-suited for distributed computation. With this respect, it is worth pointing out that the proposed algorithm is based on the parallel Schwarz method for evolution problems, which, as well known, enjoys nice convergence properties to the centralized solution as the time discretization step δ tends to zero [47]- [48]. Hence, it seems a sensible and promising approach to spread the information through the network.

4.4.1 Finite-element implementation

In practice, the algorithm, and in particular the solution of the boundary value problem (4.15)-(4.17), has to be implemented via a finite dimensional approximation. In particular, we follow the same approach described in Section 4.3 for the centralized case by constructing a FE mesh for the global domain Ω, and then decomposing such a grid into N possibly overlapping

Table 4.1: Algorithm 1: Parallel Schwarz filter

1: Given $\mathbf{y}_q^m$, update the prior estimate $\hat{x}_{q|q-1}^m$ into $\hat{x}_{q|q}^m$.
2: Initialize the prediction with $\hat{x}_{q,0}^m = \hat{x}_{q|q}^m$ and $\hat{x}_{q,-1}^m = \hat{x}_{q|q}^m$.
3:
4: **for** $\ell = 1, \ldots, L_q$ **do**
5: Exchange data with the neighborhood; specifically send to neighbor j the data $\hat{x}_{q,\ell-1}^m$ concerning the sub-boundary $\Gamma_{jm} \subset \partial\Omega_j$, and get from neighbor j the data $\hat{x}_{q,\ell-1}^j$ concerning the sub-boundary $\Gamma_{mj} \subset \partial\Omega_m$.
6: Solve the problem

$$\frac{\hat{x}_{q,\ell}^m - \hat{x}_{q,\ell-1}^m}{\delta} + \mathcal{L}\left(\hat{x}_{q,\ell}^m\right) \;=\; f_{q,\ell} \quad \text{in } \Omega_m \tag{4.15}$$

subject to the Dirichlet boundary conditions

$$\hat{x}_{q,\ell}^m = \hat{x}_{q,\ell-1}^j \quad \text{on } \Gamma_{mj} \qquad \forall j \in \mathcal{N}_m \backslash \{m\} \tag{4.16}$$

and the linear boundary conditions

$$\mathcal{B}(\hat{x}_{q,\ell}^m) = 0 \text{ on } \Gamma_{mm}\,. \tag{4.17}$$

where $f_{q,\ell}(\mathbf{p}) \stackrel{\triangle}{=} f\left(\mathbf{p}, t_q + \ell\delta\right)$.
7: **end for**
8:
9: Set $\hat{x}_{q+1|q}^m = \hat{x}_{q,L_q}^m$ for the next cycle.

sub-meshes, according to the domain decomposition. For the sequel, it is important to distinguish vertices lying on the boundary between neighbors (*interface*) from the other vertices of the subdomain. To this end, let $\text{int}(S)$ denote the interior of a generic set S. Then, we introduce the sets of indices $\mathfrak{I}_m \stackrel{\triangle}{=} \{i : \mathbf{p}_i \in \text{int}(\Omega_m) \cup \Gamma_{mm}\}$ and $\mathfrak{I}_{mj} \stackrel{\triangle}{=} \{i : \mathbf{p}_i \in \Gamma_{mj}\}$ of the basis functions corresponding to *internal* and, respectively, *interface* vertices of subdomain Ω_m. In particular, let $\mathbf{x}^m \stackrel{\triangle}{=} col\{x_i : i \in \mathfrak{I}_m\}$, $m = 1, \ldots, N$, denote the vector of field values in vertices belonging to $\text{int}(\Omega_m) \cup \Gamma_{mm}$, i.e. the *internal* state of subsystem m. Then, it is possible to extract from (4.7)

the rows relative to states $\mathbf{x}^m$ so that

$$\begin{aligned}\mathbf{M}^{mm} \quad \dot{\mathbf{x}}^m + \sum_{j\in\mathcal{N}_m\setminus\{m\}} \mathbf{M}^{mj}\dot{\mathbf{x}}^j + \mathbf{S}^{mm}\mathbf{x}^m \\ + \sum_{j\in\mathcal{N}_m\setminus\{m\}} \mathbf{S}^{mj}\mathbf{x}^j = \mathbf{u}^m + \boldsymbol{\epsilon}^m\end{aligned} \tag{4.18}$$

where the matrices $\mathbf{M}^{mj}$ and $\mathbf{S}^{mj}$ take into account the contribution of state variables in vertices $\mathbf{p}_j \in \Gamma_{mj}$, and $\boldsymbol{\epsilon}^m$ accounts for the approximation error in the finite-dimensional representation (4.4) of x in terms of basis functions. Notice that both $\mathbf{M}^{mm}$ and $\mathbf{S}^{mm}$ are positive definite due to positive definiteness of $\mathbf{M}$ and $\mathbf{S}$. As a result, the ODE (4.7) can be written as the interconnection of N subsystems of the form (4.18).

Each of the subsystems (4.18) can be discretized in time in the interval $[t_q, t_{q+1}]$ using a modified backward Euler technique wherein a delay is introduced in the terms concerning neighboring nodes, so that at time $t_q + \ell\delta$ we obtain the following discrete-time linear descriptor system

$$\begin{aligned}&\mathbf{M}^{mm}\left(\frac{\mathbf{x}^m_{q,\ell+1} - \mathbf{x}^m_{q,\ell}}{\delta}\right) + \mathbf{S}^{mm}\mathbf{x}^m_{q,\ell+1} \\ &\qquad + \sum_{j\in\mathcal{N}_m\setminus\{m\}}\left[\mathbf{M}^{mj}\left(\frac{\mathbf{x}^j_{q,\ell} - \mathbf{x}^j_{q,\ell-1}}{\delta}\right) + \mathbf{S}^{mj}\mathbf{x}^j_{q,\ell}\right] \\ &= \mathbf{u}^m_{q,\ell+1} + \boldsymbol{\epsilon}^m_{q,\ell+1} + \boldsymbol{\tau}^m_{q,\ell}\end{aligned} \tag{4.19}$$

where $\mathbf{x}^m_{q,\ell} \triangleq \mathbf{x}^m(t_q + \ell\delta)$, for $\ell = 1 \ldots, L_q$, and $\boldsymbol{\tau}^m_{q,\ell}$ denotes the time discretization error at time $t_q + \ell\delta$. The recursion (4.19) is initialized at time t_q by setting

$$\begin{aligned}&\mathbf{x}^m_{q,0} = \mathbf{x}^m(t_q), \\ &\mathbf{x}^j_{q,0} = \mathbf{x}^j(t_q), \quad \mathbf{x}^j_{q,-1} = \mathbf{x}^j(t_q), \quad j \in \mathcal{N}_m \setminus \{m\}\end{aligned} \tag{4.20}$$

The well-posedness of the discretization scheme resulting from (4.19)-(4.20) will be analyzed in Section 4.4.2.

It can be readily seen that such a hybrid Euler time-discretization implements the Parallel Schwarz method described earlier. In fact, it is equivalent to approximate x in Ω_m at time $t_q + \ell\delta$ as

$$x(\mathbf{p}, t_q + \ell\delta) \approx \sum_{i\in\mathfrak{I}_m} \phi^m_i(\mathbf{p})\, x^{m,i}_{q,\ell} + \sum_{j\in\mathcal{N}_m\setminus\{m\}}\sum_{i\in\mathfrak{I}_{mj}} \phi^j_i(\mathbf{p})\, x^{j,i}_{q,\ell-1} \tag{4.21}$$

which, in turn, corresponds to explicitly imposing inhomogeneous Dirichlet interface conditions on $\Gamma_{mj}, j \in \mathcal{N}_m \setminus \{m\}$ taken from neighboring nodes (like in (4.16)).

Thanks to the positive definiteness of $\mathbf{M}^{mm}$ and $\mathbf{S}^{mm}$, each discretized model (4.19) can be easily transformed into a state-space model of the form

$$\begin{aligned} \mathbf{x}_{q,\ell}^m &= \mathbf{A}^m \mathbf{x}_{q,\ell-1}^m + \sum_{j \in \mathcal{N}_m \setminus \{m\}} \mathbf{A}^{mj} \hat{\mathbf{x}}_{q,\ell-1}^j \\ &\quad + \sum_{j \in \mathcal{N}_m \setminus \{m\}} \bar{\mathbf{A}}^{mj} \mathbf{x}_{q,\ell-2}^j + \mathbf{B}^m \mathbf{u}_{q,\ell}^m + \mathbf{w}_{q,\ell}^m \end{aligned} \tag{4.22}$$

where

$$\begin{aligned} \mathbf{A}^m &= \left(\mathbf{M}^{mm} + \delta \mathbf{S}^{mm}\right)^{-1} \mathbf{M}^{mm} \\ \mathbf{A}^{mj} &= \left(\mathbf{M}^{mm} + \delta \mathbf{S}^{mm}\right)^{-1} \left(-\delta \mathbf{S}^{mj} - \mathbf{M}^{mj}\right) \\ \bar{\mathbf{A}}^{mj} &= \left(\mathbf{M}^{mm} + \delta \mathbf{S}^{mm}\right)^{-1} \mathbf{M}^{mj} \\ \mathbf{B}^m &= \left(\mathbf{M}^{mm} + \delta \mathbf{S}^{mm}\right)^{-1} \delta \end{aligned}$$

and $\mathbf{w}_{q,\ell}^m = \left(\mathbf{M}^{mm} + \delta\, \mathbf{S}^{mm}\right)^{-1} \delta \left(\tilde{\boldsymbol{\epsilon}}_{q,\ell+1}^m + \boldsymbol{\tau}_{q,\ell}^m\right)$ is the error combining the effects of both spatial and temporal discretizations.

Such interconnected models can be exploited so as to derive a FE approximation of the distributed-state estimation algorithm with Parallel Schwarz method (Algorithm 1 in Table 4.1). In particular, the numerical solution of (4.15)-(4.17) takes the form of the local one-step-ahead predictor for model (4.22) at time $t_q + (\ell - 1)\delta$, whereas the correction step of the local filtering cycle is the usual (extended) Kalman filter update step for the local subsystem. The resulting distributed finite-element (extended) Kalman filter is reported in Table 4.2.

As previously shown, the additional terms $\sum_{j \in \mathcal{N}_m \setminus \{m\}} \mathbf{A}^{mj} \hat{\mathbf{x}}_{q,\ell-1}^j$ and $\sum_{j \in \mathcal{N}_m \setminus \{m\}} \bar{\mathbf{A}}^{mj} \hat{\mathbf{x}}_{q,\ell-2}^j$ in equation (4.22) arise from the non-homogeneous Dirichlet boundary conditions (4.16). In this respect, it is worth noting that the matrices $\mathbf{A}^{mj}$ and $\bar{\mathbf{A}}^{mj}$ are sparse since only the components of the neighbor estimates $\hat{\mathbf{x}}_{q,\ell-1}^j$ and $\hat{\mathbf{x}}_{q,\ell-2}^j$ concerning the sub-boundary Γ_{mj} are involved. The positive real $\gamma > 1$ is a covariance boosting factor whose role, as will be discussed in the stability analysis of the distributed FE-KF, is that of guaranteeing convergence of the estimates. The covariance boosting factor is also necessary in order to compensate for the additional uncertainty associated with the boundary conditions at the interfaces, i.e., for

Table 4.2: Algorithm 2: Distributed finite-element Kalman filter

1: Given $\mathbf{y}_q^m$, update the prior estimate $\hat{\mathbf{x}}_{q|q-1}^m$ and covariance $\mathbf{P}_{q|q-1}^m$ into $\hat{\mathbf{x}}_{q|q}^m$ and $\mathbf{P}_{q|q}^m$ as follows

$$\begin{aligned}
\hat{\mathbf{x}}_{q|q}^m &= \hat{\mathbf{x}}_{q|q-1}^m + \mathbf{L}_q^m \left(\mathbf{y}_q^m - \mathbf{h}^m \left(\hat{\mathbf{x}}_{q|q-1}^m \right) \right) \\
\mathbf{P}_{q|q}^m &= \mathbf{P}_{q|q-1}^m - \mathbf{L}_q^m (\mathbf{C}_q^m)^T \mathbf{P}_{q|q-1}^m \\
\mathbf{C}_q^m &= \frac{\partial \mathbf{h}^m}{\partial \mathbf{x}} \left(\hat{\mathbf{x}}_{q|q-1}^m \right) \\
\mathbf{L}_q^m &= \mathbf{P}_{q|q-1}^m \mathbf{C}_q^m \left(\mathbf{R}_q^m + \mathbf{C}_q^m \mathbf{P}_{q|q-1}^m (\mathbf{C}_q^m)^T \right)^{-1}
\end{aligned}$$

where $\mathbf{h}^m \stackrel{\triangle}{=} col\left\{h_i : \mathbf{s}_i \in \Omega_m\right\}$ denote the local measurement function at node m.

2: Initialize the distributed prediction with $\hat{\mathbf{x}}_{q,0}^m = \hat{\mathbf{x}}_{q|q}^m$, $\mathbf{P}_{q,0}^m = \mathbf{P}_{q|q}^m$ and $\hat{\mathbf{x}}_{q,-1}^m = \hat{\mathbf{x}}_{q|q}^m$, $\mathbf{P}_{q,-1}^m = \mathbf{P}_{q|q}^m$.

3:

4: **for** $\ell = 1, \ldots, L_q$ **do**

5: Exchange data with the neighborhood; specifically send to neighbor j the estimates $\hat{\mathbf{x}}_{q,\ell-1}^m$concerning the sub-boundary $\Gamma_{jm} \subset \partial\Omega_j$, and get from neighbor j the estimates $\hat{\mathbf{x}}_{q,\ell-1}^j$concerning the sub-boundary $\Gamma_{mj} \subset \partial\Omega_m$.

6: set

$$\begin{aligned}
\hat{\mathbf{x}}_{q,\ell}^m &= \mathbf{A}^m \hat{\mathbf{x}}_{q,\ell-1}^m + \sum_{j \in \mathcal{N}_m \setminus \{m\}} \mathbf{A}^{mj} \hat{\mathbf{x}}_{q,\ell-1}^j \\
&\quad + \sum_{j \in \mathcal{N}_m \setminus \{m\}} \bar{\mathbf{A}}^{mj} \hat{\mathbf{x}}_{q,\ell-2}^j + \mathbf{B}^m \mathbf{u}_{q,\ell}^m \qquad (4.23) \\
\mathbf{P}_{q,\ell}^m &= \gamma^2 \mathbf{A}^m \mathbf{P}_{q,\ell-1}^m \left(\mathbf{A}^m \right)^T + \mathbf{Q}^m \qquad (4.24)
\end{aligned}$$

with $\gamma > 1$.

7: **end for**

8:

9: Set $\hat{\mathbf{x}}_{q+1|q}^m = \hat{\mathbf{x}}_{q,L_q}^m$ and $\mathbf{P}_{q+1|q}^m = \mathbf{P}_{q,L_q}^m$ for the next cycle.

the uncertainty associated with the estimates $\sum_{j\in\mathcal{N}_m\setminus\{m\}} \mathbf{A}^{mj}\hat{\mathbf{x}}^j_{q,\ell-1}$ and $\sum_{j\in\mathcal{N}_m\setminus\{m\}} \bar{\mathbf{A}}^{mj}\hat{\mathbf{x}}^j_{q,\ell-2}$. In fact, such an uncertainty is not explicitly accounted for in (4.24) due to the fact that the correlation between the estimates of neighboring nodes is not precisely known. The interested reader is referred to [16] for additional insights on this issue in the context of distributed estimation of large-scale interconnected systems. As in the centralized context, the positive definite matrix $\mathbf{Q}^m$ accounts for the various uncertainties and imprecisions (i.e., discretization errors, imprecise knowledge of the exogenous input f and of the boundary conditions (4.17)).

4.4.2 Numerical stability

As previously shown, in the FE-based implementation the Parallel Schwarz step amounts to performing a hybrid Euler discretization on the interconnection of the N subsystems (4.18). Hence, as a preliminary analysis step, it is important to verify the well-posedness of such a modified discretization method in terms of numerical stability (i.e., in terms of boundedness and convergence of the time-discretization errors). To this end, it is convenient to consider the global dynamics of the interconnection.

Let us consider the augmented global state $\tilde{\mathbf{x}} \triangleq col\{\mathbf{x}^m,\, m=1,\ldots,N\}$, which clearly may contain repeated components of the state due to the possibly overlapping nature of the decomposition. Let the vectors $\tilde{\mathbf{u}}$ and $\tilde{\boldsymbol{\epsilon}}$ be defined in a similar way. In terms of $\tilde{\mathbf{x}}$, the interconnection of the N subsystems of the form (4.18) gives rise to a global augmented system which obeys the following continuous-time linear dynamics

$$\tilde{\mathbf{M}}\,\dot{\tilde{\mathbf{x}}} + \tilde{\mathbf{S}}\,\tilde{\mathbf{x}} = \tilde{\mathbf{u}} + \tilde{\boldsymbol{\epsilon}} \tag{4.25}$$

Note that the only difference between (4.7) and (4.25) is the presence of duplicated states in the latter linear ODE. Nevertheless, the two systems originate an identical state evolution. According to the divide-and-conquer strategy, matrices $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{S}}$ can be decomposed as

$$\tilde{\mathbf{M}} = \tilde{\mathbf{M}}_D + \tilde{\mathbf{M}}_F \tag{4.26}$$

$$\tilde{\mathbf{S}} = \tilde{\mathbf{S}}_D + \tilde{\mathbf{S}}_F \tag{4.27}$$

with $\tilde{\mathbf{M}}_D = \text{block-diag}(\mathbf{M}^{11},\ldots,\mathbf{M}^{NN})$, $\tilde{\mathbf{S}}_D = \text{block-diag}(\mathbf{S}^{11},\ldots,\mathbf{S}^{NN})$, whereas $\tilde{\mathbf{M}}_F$ and $\tilde{\mathbf{S}}_F$ take into account the FE interconnection structure

among neighboring subsystems. By substituting (4.26)-(4.27) into (4.25), one obtains

$$\tilde{\mathbf{M}}_D\,\dot{\tilde{\mathbf{x}}} + \tilde{\mathbf{S}}_D\,\tilde{\mathbf{x}} + \tilde{\mathbf{M}}_F\,\dot{\tilde{\mathbf{x}}} + \tilde{\mathbf{S}}_F\,\tilde{\mathbf{x}} = \tilde{\mathbf{u}} + \tilde{\boldsymbol{\epsilon}}\,. \tag{4.28}$$

Then, by applying the hybrid Euler time discretization (4.19), the time-discretized augmented system takes the form

$$\begin{aligned}\tilde{\mathbf{M}}_D\left(\frac{\tilde{\mathbf{x}}_{q,\ell+1}-\tilde{\mathbf{x}}_{q,\ell}}{\delta}\right) + \tilde{\mathbf{S}}_D\tilde{\mathbf{x}}_{q,\ell+1} + \tilde{\mathbf{M}}_F\left(\frac{\tilde{\mathbf{x}}_{q,\ell}-\tilde{\mathbf{x}}_{q,\ell-1}}{\delta}\right)\\ + \tilde{\mathbf{S}}_F\tilde{\mathbf{x}}_{q,\ell} = \tilde{\mathbf{u}}_{q,\ell+1} + \tilde{\boldsymbol{\epsilon}}_{q,\ell+1} + \boldsymbol{\tau}_{q,\ell}\end{aligned} \tag{4.29}$$

for $\ell = 0,\ldots,L_q-1$, where $\tilde{\mathbf{x}}_{q,\ell} \stackrel{\triangle}{=} \tilde{\mathbf{x}}(t_q+\ell\delta)$ and, as previously, $\boldsymbol{\tau}_{q,\ell}$ denotes the time-discretization error at time $t_q+\ell\delta$. Further, the initialization (4.20) can be simply rewritten as

$$\tilde{\mathbf{x}}_{q,0} = \tilde{\mathbf{x}}_{q,-1} = \tilde{\mathbf{x}}(t_q) \tag{4.30}$$

The following result can now be stated which summarizes the numerical stability properties[2] of (4.29)-(4.30).

Theorem 5. *The hybrid Euler time-discretization scheme (4.29)-(4.30) is* consistent *with local truncation error of order* 1. *Further, it is* zero-stable *provided that the following condition holds*

$$\rho(\tilde{\mathbf{M}}_D^{-1}\,\tilde{\mathbf{M}}_F) < 1 \tag{4.31}$$

where $\rho(\cdot)$ *denotes the spectral radius.*

Proof: Let $\mathcal{D}$ denote the differential operator in the left-hand side of (4.28), i.e.,

$$\mathcal{D}(\boldsymbol{\xi},t) = \tilde{\mathbf{M}}_D\,\dot{\boldsymbol{\xi}}(t) + \tilde{\mathbf{S}}_D\,\boldsymbol{\xi}(t) + \tilde{\mathbf{M}}_F\,\dot{\boldsymbol{\xi}}(t) + \tilde{\mathbf{S}}_F\,\boldsymbol{\xi}(t)$$

for any smooth time-function $\boldsymbol{\xi}$. Further, let $\mathcal{D}_\delta$ denote the discrete-time operator in the left-hand side of (4.29), i.e.,

$$\begin{aligned}\mathcal{D}_\delta(\boldsymbol{\xi},t) \;=\; & \tilde{\mathbf{M}}_D\left(\frac{\boldsymbol{\xi}(t+\delta)-\boldsymbol{\xi}(t)}{\delta}\right) + \tilde{\mathbf{S}}_D\,\boldsymbol{\xi}(t+\delta)\\ & + \tilde{\mathbf{M}}_F\left(\frac{\boldsymbol{\xi}(t)-\boldsymbol{\xi}(t-\delta)}{\delta}\right) + \tilde{\mathbf{S}}_F\,\boldsymbol{\xi}(t)\,.\end{aligned}$$

[2]The interested reader is referred to chapter 12 of [57] for an introduction on the concepts of consistency, zero-stability, and convergence of time-discretization methods.

As well known, the time-discretization scheme (4.29) is consistent when, for any smooth time-function $\boldsymbol{\xi}$ and for any time t, $\mathcal{D}_\delta(\boldsymbol{\xi},t)$ converges to $\mathcal{D}(\boldsymbol{\xi},t)$ as δ goes to 0. By taking the Taylor expansion of $\boldsymbol{\xi}$ in t, we can write $\boldsymbol{\xi}(t+\delta)=\boldsymbol{\xi}(t)+\delta\,\dot{\boldsymbol{\xi}}(t)+\delta^2\,\ddot{\boldsymbol{\xi}}(t)+O(\delta^3)$ and $\boldsymbol{\xi}(t-\delta)=\boldsymbol{\xi}(t)-\delta\,\dot{\boldsymbol{\xi}}(t)+\delta^2\,\ddot{\boldsymbol{\xi}}(t)+O(\delta^3)$. Hence, after some algebra, we have

$$\begin{aligned}\mathcal{D}_\delta(\boldsymbol{\xi},t) &= \mathcal{D}(\boldsymbol{\xi},t)+\tilde{\mathbf{M}}_D\,\delta\,\ddot{\boldsymbol{\xi}}(t)+\tilde{\mathbf{S}}_D\,\delta\,\dot{\boldsymbol{\xi}}(t)-\tilde{\mathbf{M}}_F\,\delta\,\ddot{\boldsymbol{\xi}}(t)\\ &\quad+O(\delta^2)\end{aligned}$$

which shows that the scheme is consistent and the local truncation error has order 1.

In order to study zero-stability, we start by considering the limit for δ going to zero of the time-difference equation (4.29), which is given by

$$\tilde{\mathbf{M}}_D\,(\tilde{\mathbf{x}}_{q,\ell+1}-\tilde{\mathbf{x}}_{q,\ell})+\tilde{\mathbf{M}}_F\,(\tilde{\mathbf{x}}_{q,\ell}-\tilde{\mathbf{x}}_{q,\ell-1})=0\,. \tag{4.32}$$

In fact, zero-stability of the time-discretization scheme (4.29) corresponds to the neutral stability of the discrete-time system (4.32) (recall that system (4.32) is neutrally stable when its trajectories remain bounded as ℓ goes to infinity for any initial condition). Then the proof can be concluded by noting that, by defining $\boldsymbol{\zeta}_{q,\ell+1}=\tilde{\mathbf{x}}_{q,\ell+1}-\tilde{\mathbf{x}}_{q,\ell}$, system (4.32) can be rewritten as

$$\begin{bmatrix}\tilde{\mathbf{x}}_{q,\ell+1}\\ \boldsymbol{\zeta}_{q,\ell+1}\end{bmatrix}=\begin{bmatrix}\mathbf{I} & -\tilde{\mathbf{M}}_D^{-1}\tilde{\mathbf{M}}_F\\ \mathbf{0} & -\tilde{\mathbf{M}}_D^{-1}\tilde{\mathbf{M}}_F\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{x}}_{q,\ell}\\ \boldsymbol{\zeta}_{q,\ell}\end{bmatrix}$$

which is neutrally stable if and only if condition (4.31) holds. ■

Recall that, in view of the *Dahlquist's Equivalence Theorem*, zero-stability is necessary and sufficient for convergence of a consistent time-discretization scheme [57]. Hence, under condition (4.31), the hybrid Euler time-discretization scheme (4.19) turns out to be convergent. For instance, this means that in each interval $[t_q,t_{q+1}]$ the predicted estimates obtained via the Parallel Schwarz step (4.23) converge to the solution of a centralized prediction equation of the form

$$\tilde{\mathbf{M}}\,\dot{\hat{\mathbf{x}}}+\tilde{\mathbf{S}}\,\hat{\mathbf{x}}=\tilde{\mathbf{u}}$$

as the time-discretization step δ goes to 0, or equivalently as the number L_q of distributed prediction steps goes to infinity.

Remark 3. *It is worth noting that (4.31) translates into a block diagonal dominance condition for the global system, which requires that the effect of*

the isolated subsystems on the state evolution prevails over the effect originated from the interconnections among subsystems. Taking into account the particular structure of the FE mass matrix $\mathbf{M}$*, which is reflected in the sparse structure of* $\tilde{\mathbf{M}}$*, the numerical stability condition (4.31) is usually satisfied in practice (see, for instance, the simulation example of Section VI). In addition, in the unlikely case in which condition (4.31) does not hold, it is possible to modify the hybrid Euler time-discretization scheme (4.29) (and hence the implementation of the Parallel Schwarz step) so as to retrieve zero-stability. Specifically, by introducing a suitable scalar* $\omega \in (0, 1]$*, one can replace (4.29) with*

$$\begin{aligned}
&\tilde{\mathbf{M}}_D \left(\frac{\tilde{\mathbf{x}}_{q,\ell+1} - (2-\omega)\,\tilde{\mathbf{x}}_{q,\ell} + (1-\omega)\,\tilde{\mathbf{x}}_{q,\ell-1}}{\omega\,\delta} \right) \\
&+ \tilde{\mathbf{S}}_D \tilde{\mathbf{x}}_{q,\ell+1} + \tilde{\mathbf{M}}_F \left(\frac{\tilde{\mathbf{x}}_{q,\ell} - \tilde{\mathbf{x}}_{q,\ell-1}}{\delta} \right) + \tilde{\mathbf{S}}_F \tilde{\mathbf{x}}_{q,\ell} \\
&= \tilde{\mathbf{u}}_{q,\ell+1} + \tilde{\boldsymbol{\epsilon}}_{q,\ell+1} + \boldsymbol{\tau}_{q,\ell}
\end{aligned} \tag{4.33}$$

which is still well-suited for distributed implementation. Notice that such a modified scheme coincides with (4.29) for $\omega = 1$*. Further, along the lines of Theorem 1, it is possible to show that (4.33) is consistent for any value of* $\omega \in (0, 1]$*, and zero-stable provided that*

$$\rho(\omega\,\tilde{\mathbf{M}}_D^{-1}\,\tilde{\mathbf{M}}_F - (1-\omega)\,\mathbf{I}) < 1\,. \tag{4.34}$$

In turn, since

$$\rho(\omega\,\tilde{\mathbf{M}}_D^{-1}\,\tilde{\mathbf{M}}_F - (1-\omega)\,\mathbf{I}) \leq \max\{\omega\,\rho(\tilde{\mathbf{M}}_D^{-1}\,\tilde{\mathbf{M}}_F),\, 1-\omega\}$$

for any $\omega \in (0, 1]$*, condition (4.34) can be always satisfied for suitably small values of* ω *even when condition (4.31) does not hold. The price to be paid for the improved numerical stability is a slow-down of the information spread.*

4.5 Stability analysis

In this section, the stability of the estimation error dynamics resulting from application of the distributed finite-element Kalman filter of Algorithm 2 (Table 4.2) is analyzed by supposing the measurement equation in each domain to be linear (as it happens when the sensors directly measure the target field like in (4.10)). Further, in order to simplify the notation, the interval

$t_{q+1} - t_q$ between consecutive measurements is supposed to be constant, so that in each sampling interval $[t_q, t_{q+1})$ a fixed number L of distributed prediction steps is performed. In this respect, we make the following assumption.

A3. For each $m \in \mathcal{N}$, the local measurement function is linear, i.e., $\mathbf{h}^m(\mathbf{x}^m) = \mathbf{C}^m\mathbf{x}^m$. Further, local observability holds in the sense that the pair $((\mathbf{A}^m)^L, \mathbf{C}^m)$ is observable for any $m \in \mathcal{N}$.

A set of sensors ensuring local observability in each domain ensures also global observability (i.e., observability of the global state vector given all the measurements). However, the converse need not hold in that local observability requires a sufficient number of sensors to be present in each subdomain. Nevertheless, under global observability, the local observability condition can be satisfied by choosing each subdomain large enough so that a sufficient number of sensors is included inside. Recalling that the matrices $\mathbf{A}^m$ arise from space-time discretization of a PDE, some comments on how the local observability assumption A3 maps to the original continuous field are important. In this respect, while the relationship between observability of a continuous field and of its space-time discretization is far from trivial [58,59], the following considerations can be made: i) from the practical point of view, unless the domain Ω has a very specific form, the exact observability of the original PDE solution cannot be directly checked, and one invariably needs to resort to some numerical approximation scheme [58] like the one considered here; ii) on the other hand, it has been proved that, for a convergent discrete approximation scheme, the observability of the discrete numerical model is sufficient (and necessary) for the stability of the related field estimation process (see [58] for a formal statement of this property); iii) finally, it has been recently shown [59] that quantitative observability measures, defined in terms of suitable observability Gramians, carry over in a consistent way from the original PDE to its space-time discretization for any convergent numerical approximation scheme.

Let us first rewrite (4.29) into the state-space form

$$\tilde{\mathbf{x}}_{q,\ell+1} = \underbrace{\left(\tilde{\mathbf{M}}_D + \delta\tilde{\mathbf{S}}_D\right)^{-1}\tilde{\mathbf{M}}_D}_{\tilde{\mathbf{A}}_D}\tilde{\mathbf{x}}_{q,\ell} + \underbrace{\left(\tilde{\mathbf{M}}_D + \delta\tilde{\mathbf{S}}_D\right)^{-1}\left(-\delta\tilde{\mathbf{S}}_F - \tilde{\mathbf{M}}_F\right)}_{\tilde{\mathbf{A}}_F}\tilde{\mathbf{x}}_{q,\ell}$$
$$+ \underbrace{\left(\tilde{\mathbf{M}}_D + \delta\tilde{\mathbf{S}}_D\right)^{-1}\tilde{\mathbf{M}}_F}_{\bar{\mathbf{A}}_F}\tilde{\mathbf{x}}_{q,\ell-1} + \underbrace{\left(\tilde{\mathbf{M}}_D + \delta\tilde{\mathbf{S}}_D\right)^{-1}\delta}_{\tilde{\mathbf{B}}}\tilde{\mathbf{u}}_{q,\ell+1} + \tilde{\mathbf{w}}_{q,\ell} \qquad (4.35)$$

where, clearly, $\tilde{\mathbf{A}}_D = \text{block} - \text{diag}(\mathbf{A}^1, \ldots, \mathbf{A}^N)$ is the block diagonal matrix of state transition matrices, representing the N isolated subsystems.

Recalling that, in each interval $[t_q, t_{q+1})$, the recursion (4.35) is initialized with the initial conditions (4.30), it can be easily noticed that at the last distributed prediction step $\ell = L$ one obtains

$$\tilde{\mathbf{x}}_{q,L} = \tilde{\mathbf{A}}_D^L\,\tilde{\mathbf{x}}_{q,0} + \tilde{\mathbf{A}}_{F,L}\tilde{\mathbf{x}}_{q,0} + \tilde{\mathbf{B}}_L\tilde{\mathbf{U}}_q + \tilde{\mathbf{D}}_L\tilde{\mathbf{W}}_q \qquad (4.36)$$

where $\tilde{\mathbf{U}}_q \triangleq col\{\mathbf{u}_{q,\ell},\ \ell = 1, \ldots, L\}$, $\tilde{\mathbf{W}}_q \triangleq col\{\mathbf{w}_{q,\ell},\ \ell = 1, \ldots, L\}$ and $\tilde{\mathbf{B}}_L$, $\tilde{\mathbf{D}}_L$, $\tilde{\mathbf{A}}_{F,L}$ are suitable matrices with the latter defining the interconnection couplings between subsystems. Noting that, by definition, $\tilde{\mathbf{x}}_{q,L} = \tilde{\mathbf{x}}_{q+1,0} = \tilde{\mathbf{x}}(T_{q+1}\Delta)$, the latter equation can be rewritten as

$$\tilde{\mathbf{x}}_{q+1} = \tilde{\mathbf{A}}_D^L\,\tilde{\mathbf{x}}_q + \tilde{\mathbf{A}}_{F,L}\tilde{\mathbf{x}}_q + \tilde{\mathbf{B}}_L\tilde{\mathbf{U}}_q + \tilde{\mathbf{D}}_L\tilde{\mathbf{W}}_q \qquad (4.37)$$

where $\tilde{\mathbf{x}}_q \triangleq \tilde{\mathbf{x}}(T_q\Delta)$.

Similarly, application of step 3 of Algorithm 2 yields, at the last distributed prediction step $\ell = L$,

$$\hat{\mathbf{x}}_{q,L} = \tilde{\mathbf{A}}_D^L\,\hat{\mathbf{x}}_{q,0} + \tilde{\mathbf{A}}_{F,L}\hat{\mathbf{x}}_{q,0} + \tilde{\mathbf{B}}_L\tilde{\mathbf{U}}_q\,. \qquad (4.38)$$

where $\hat{\mathbf{x}}_{q,\ell} \triangleq col\{\hat{\mathbf{x}}_{q,\ell}^m,\ m \in \mathcal{N}\}$. Further, by defining $\hat{\mathbf{x}}_{q|q} \triangleq col\{\hat{\mathbf{x}}_{q|q}^m,\ m \in \mathcal{N}\}$ and $\hat{\mathbf{x}}_{q|q-1} \triangleq col\{\hat{\mathbf{x}}_{q|q-1}^m,\ m \in \mathcal{N}\}$, the global correction step of Algorithm 2 at time t_{q+1} can be written as

$$\hat{\mathbf{x}}_{q+1|q+1} = \hat{\mathbf{x}}_{q+1|q} + \tilde{\mathbf{L}}_{q+1}(\tilde{\mathbf{y}}_{q+1} - \tilde{\mathbf{C}}\,\hat{\mathbf{x}}_{q+1|q}) \qquad (4.39)$$

where $\tilde{\mathbf{y}}_{q+1} \triangleq col\{\mathbf{y}_{q+1}^m,\ m \in \mathcal{N}\}$, $\tilde{\mathbf{L}}_{q+1} = \text{block} - \text{diag}(\mathbf{L}_{q+1}^1, \ldots, \mathbf{L}_{q+1}^N)$, and $\tilde{\mathbf{C}} \triangleq col\{\mathbf{C}^m,\ m \in \mathcal{N}\}$.

Recalling that $\hat{\mathbf{x}}_{q,L} = \hat{\mathbf{x}}_{q+1|q}$ and $\hat{\mathbf{x}}_{q,0} = \hat{\mathbf{x}}_{q|q}$, equations (4.38) and (4.39) can be easily combined so as to write $\hat{\mathbf{x}}_{q+1|q+1}$ as a function of $\hat{\mathbf{x}}_{q|q}$ and thus

obtain a recursive expression for the global estimate. In addition, noting that the global output vector can be written as $\tilde{\mathbf{y}}_{q+1} = \tilde{\mathbf{C}}\tilde{\mathbf{x}}_{q+1} + \tilde{\mathbf{v}}_{q+1}$ with $\tilde{\mathbf{v}}_{q+1} \triangleq col\{\mathbf{v}_{q+1}^m,\, m \in \mathcal{N}\}$, we can also write a recursive expression for the dynamics of the global estimation error $\tilde{\mathbf{e}}_q \triangleq col\{\tilde{\mathbf{x}}_q - \hat{\mathbf{x}}_{q|q},\, m \in \mathcal{N}\}$. Specifically, standard calculations yield

$$\tilde{\mathbf{e}}_{q+1} = \left(\mathbf{I} - \tilde{\mathbf{L}}_{q+1}\tilde{\mathbf{C}}\right)\left(\tilde{\mathbf{A}}_D^L + \tilde{\mathbf{A}}_{F,L}\right)\tilde{\mathbf{e}}_q + \tilde{\boldsymbol{\nu}}_q \tag{4.40}$$

where the term $\tilde{\boldsymbol{\nu}}_q = (\mathbf{I} - \tilde{\mathbf{L}}_{q+1}\tilde{\mathbf{C}})\tilde{\mathbf{D}}_L\tilde{\mathbf{W}}_q + \tilde{\mathbf{v}}_{q+1}$ accounts for the time-space discretization errors, for the measurement noise, and for all the other possible uncertainties.

As for the time evolution of the global covariance matrix

$$\tilde{\mathbf{P}}_{q|q} \triangleq \text{block} - \text{diag}(\mathbf{P}_{q|q}^1, \ldots, \mathbf{P}_{q|q}^N),$$

with similar reasoning as above it is an easy matter to see that application of Algorithm 2 leads to the following recursion

$$\begin{aligned}\tilde{\mathbf{P}}_{q+1|q+1} \;=\; & \left(\mathbf{I} - \tilde{\mathbf{L}}_{q+1}\tilde{\mathbf{C}}^T\right)\tilde{\mathbf{P}}_{q+1|q} \\ & \left(\mathbf{I} - \tilde{\mathbf{L}}_{q+1}\tilde{\mathbf{C}}^T\right)\left[\gamma^{2L}\tilde{\mathbf{A}}_D^L\tilde{\mathbf{P}}_{q|q}(\tilde{\mathbf{A}}_D^L)^T + \tilde{\boldsymbol{\Phi}}\right]\end{aligned} \tag{4.41}$$

where $\tilde{\boldsymbol{\Phi}} \triangleq \sum_{i=0}^{L-1}\gamma^{2i}\tilde{\mathbf{A}}_D^i\tilde{\mathbf{Q}}(\tilde{\mathbf{A}}_D^i)^T$ and $\tilde{\mathbf{Q}} \triangleq \text{block} - \text{diag}(\mathbf{Q}^1, \ldots, \mathbf{Q}^N)$. The following stability result can now be stated.

Theorem 6. *Let assumptions A1-A3 hold and let the matrices* $\tilde{\mathbf{Q}}$ *and* $\tilde{\mathbf{R}} \triangleq \text{block} - \text{diag}(\mathbf{R}^1, \ldots, \mathbf{R}^N)$ *be positive definite. Then, the global covariance matrix asymptotically converges to the unique positive solution* $\tilde{\mathbf{P}}(\gamma)$ *of the algebraic Riccati equation*

$$[\tilde{\mathbf{P}}(\gamma)]^{-1} = \left[\gamma^{2L}\tilde{\mathbf{A}}_D^L\tilde{\mathbf{P}}(\gamma)(\tilde{\mathbf{A}}_D^L)^T + \tilde{\boldsymbol{\Phi}}\right]^{-1} + \tilde{\mathbf{C}}^T\,\tilde{\mathbf{R}}^{-1}\,\tilde{\mathbf{C}}\,,$$

and the global Kalman gain converges to the steady-state value

$$\begin{aligned}\tilde{\mathbf{L}}(\gamma) = & \left[\gamma^{2L}\tilde{\mathbf{A}}_D^L\tilde{\mathbf{P}}(\gamma)(\tilde{\mathbf{A}}_D^L)^T + \tilde{\boldsymbol{\Phi}}\right]\tilde{\mathbf{C}}^T \\ & \times\left\{\tilde{\mathbf{C}}\left[\gamma^{2L}\tilde{\mathbf{A}}_D^L\tilde{\mathbf{P}}(\gamma)(\tilde{\mathbf{A}}_D^L)^T + \tilde{\boldsymbol{\Phi}}\right]\tilde{\mathbf{C}}^T + \tilde{\mathbf{R}}\right\}^{-1}.\end{aligned} \tag{4.42}$$

Then, the dynamics (4.40) of the estimation error is exponentially stable if and only if

$$\rho\left\{\left[\mathbf{I}-\tilde{\mathbf{L}}(\gamma)\tilde{\mathbf{C}}\right]\left(\tilde{\mathbf{A}}_D^L+\tilde{\mathbf{A}}_{F,L}\right)\right\}<1\,. \tag{4.43}$$

Proof: Notice first that assumption A2 implies observability of the pair $(\tilde{\mathbf{A}}_D^L, \tilde{\mathbf{C}})$ which, as it can be easily verified through the PBH test, also implies observability of $(\gamma^L\tilde{\mathbf{A}}_D^L, \tilde{\mathbf{C}})$ for any real $\gamma > 0$. Then, the convergence of $\tilde{\mathbf{P}}_{q|q}$ to $\tilde{\mathbf{P}}(\gamma) > 0$ follows from well known results on discrete-time Kalman filtering, since (4.41) is the standard Kalman filter covariance recursion for a linear system with state matrix $\gamma^L\tilde{\mathbf{A}}_D^L$ and output matrix $\tilde{\mathbf{C}}$.

Further, it is immediate to see that the gain $\tilde{\mathbf{L}}(\gamma)$ defined in (4.42) is the steady-state global Kalman gain associated with the steady-state covariance $\tilde{\mathbf{P}}(\gamma)$. Notice finally that the matrix $\left(\mathbf{I}-\tilde{\mathbf{L}}_{q+1}\tilde{\mathbf{C}}\right)\left(\tilde{\mathbf{A}}_D^L+\tilde{\mathbf{A}}_{F,L}\right)$, which determines the dynamics of the estimation error, exponentially converges to $\left[\mathbf{I}-\tilde{\mathbf{L}}(\gamma)\tilde{\mathbf{C}}\right]\left(\tilde{\mathbf{A}}_D^L+\tilde{\mathbf{A}}_{F,L}\right)$, so that the estimation error dynamics is exponentially stable if and only if $\left[\mathbf{I}-\tilde{\mathbf{L}}(\gamma)\tilde{\mathbf{C}}\right]\left(\tilde{\mathbf{A}}_D^L+\tilde{\mathbf{A}}_{F,L}\right)$ is Schur stable, i.e., if and only if condition (4.43) is satisfied. ■

In practice, the design of the proposed distributed finite-element Kalman filter requires the tuning of the scalar parameter γ. Specifically, for any given value of γ the stability of the filter can be readily checked by means of condition (4.43). Then, the tuning of γ can be performed numerically by finding, among the values of γ satisfying the stability condition (4.43), the one yielding the best estimation accuracy (see Fig. 4.9 in Section 4.6 for an illustration of these ideas in a specific case study).
In order to understand the role played by the scalar γ in the satisfiability of condition (4.43) the following result is helpful.

Proposition 1. *A sufficient condition for (4.43) to hold is that the scalar γ satisfies the relationship*

$$\gamma^L > \left\|\mathbf{I}+\left(\tilde{\mathbf{A}}_D^L\right)^{-1}\tilde{\mathbf{A}}_{F,L}\right\|_{\tilde{\mathbf{P}}(\gamma)}\,, \tag{4.44}$$

where $\|\cdot\|_{\mathbf{M}}$ denotes the matrix norm induced by the vector norm $\|\mathbf{x}\|_{\mathbf{M}} \triangleq \sqrt{\mathbf{x}^T\mathbf{M}\mathbf{x}}$.

Proof: With standard manipulations, it can be seen that $\tilde{\mathbf{L}}(\gamma)$ and $\tilde{\mathbf{P}}(\gamma)$ satisfy the relationship

$$\begin{aligned}\tilde{\mathbf{P}}(\gamma) &= [\mathbf{I}-\tilde{\mathbf{L}}(\gamma)\tilde{\mathbf{C}}^T]\left[\gamma^{2L}\tilde{\mathbf{A}}_D^L\tilde{\mathbf{P}}(\gamma)(\tilde{\mathbf{A}}_D^L)^T+\tilde{\mathbf{\Phi}}\right]\\ &\quad\times[\mathbf{I}-\tilde{\mathbf{L}}(\gamma)\tilde{\mathbf{C}}^T]^T+\tilde{\mathbf{L}}(\gamma)\tilde{\mathbf{R}}[\tilde{\mathbf{L}}(\gamma)]^T\end{aligned}$$

so that

$$[\mathbf{I}-\tilde{\mathbf{L}}(\gamma)\tilde{\mathbf{C}}^T]\left[\gamma^{2L}\tilde{\mathbf{A}}_D^L\tilde{\mathbf{P}}(\gamma)(\tilde{\mathbf{A}}_D^L)^T\right][\mathbf{I}-\tilde{\mathbf{L}}(\gamma)\tilde{\mathbf{C}}^T]^T\le\tilde{\mathbf{P}}(\gamma)$$

and, hence,

$$\left\|[\mathbf{I}-\tilde{\mathbf{L}}(\gamma)\tilde{\mathbf{C}}^T]\tilde{\mathbf{A}}_D^L\right\|_{\tilde{\mathbf{P}}(\gamma)}\le 1/\gamma^L\,. \tag{4.45}$$

Hence, in order to complete the proof, it is sufficient to observe that

$$\begin{aligned}&\left\|\left[\mathbf{I}-\tilde{\mathbf{L}}(\gamma)\tilde{\mathbf{C}}\right]\left(\tilde{\mathbf{A}}_D^L+\tilde{\mathbf{A}}_{F,L}\right)\right\|_{\tilde{\mathbf{P}}(\gamma)}\\ &\quad\le\left\|\left[\mathbf{I}-\tilde{\mathbf{L}}(\gamma)\tilde{\mathbf{C}}\right]\tilde{\mathbf{A}}_D^L\right\|_{\tilde{\mathbf{P}}(\gamma)}\left\|\mathbf{I}+\left(\tilde{\mathbf{A}}_D^L\right)^{-1}\tilde{\mathbf{A}}_{F,L}\right\|_{\tilde{\mathbf{P}}(\gamma)}\\ &\quad\le\left\|\mathbf{I}+\left(\tilde{\mathbf{A}}_D^L\right)^{-1}\tilde{\mathbf{A}}_{F,L}\right\|_{\tilde{\mathbf{P}}(\gamma)}/\gamma^L\end{aligned}$$

where the latter inequality follows from (4.45). In fact, this implies

$$\left\|\left[\mathbf{I}-\tilde{\mathbf{L}}(\gamma)\tilde{\mathbf{C}}\right]\left(\tilde{\mathbf{A}}_D^L+\tilde{\mathbf{A}}_{F,L}\right)\right\|_{\tilde{\mathbf{P}}(\gamma)}<1$$

and hence (4.43) whenever (4.44) holds. ■

It can be seen from (4.44) that the smaller is $\tilde{\mathbf{A}}_{F,L}$ (the part of the dynamics due to interaction between subdomains) as compared to $\tilde{\mathbf{A}}_D^L$ (the local dynamics in the subdomains), the easier it becomes to achieve stability. In fact, in the limit case of no interaction ($\tilde{\mathbf{A}}_{F,L}=0$) the condition is satisfied for any $\gamma>1$. In this respect, it is worth pointing out that the quantity $\left(\tilde{\mathbf{A}}_D^L\right)^{-1}\tilde{\mathbf{A}}_{F,L}$ is usually small because of the structure of the FE matrices and the fact that the interactions are limited to the interfaces. For instance, in the case study of Section 4.6 stability of the filter is guaranteed for a wide range of values of γ. Nevertheless, in general it is not possible to guarantee that a value of γ satisfying (4.44), or (4.43), always exists. This state of

affairs can be understood by noting that in (4.44) both the left-hand and the right-hand side increase with γ. In case a suitable γ cannot be found, the stability of the filter can be guaranteed by resorting to a slight modification of the proposed approach which is summarized in the following procedure:

a) select the time interval δ so that the dynamics of (4.37) is asymptotically stable;

b) pick any $\gamma > 1$ (for example, by minimizing, the left-hand side of (4.43));

c) find a scalar $\kappa > 0$ such that

$$\rho\left\{\left[\mathbf{I} - \kappa\,\tilde{\mathbf{L}}(\gamma)\tilde{\mathbf{C}}\right]\left(\tilde{\mathbf{A}}_D^L + \tilde{\mathbf{A}}_{F,L}\right)\right\} < 1\,; \tag{4.46}$$

d) modify the correction step (4.39) as follows

$$\hat{\mathbf{x}}_{q+1|q+1} = \hat{\mathbf{x}}_{q+1|q} + \kappa\,\tilde{\mathbf{L}}_{q+1}(\tilde{\mathbf{y}}_{q+1} - \tilde{\mathbf{C}}\,\hat{\mathbf{x}}_{q+1|q})\,. \tag{4.47}$$

Notice that the stability of (4.37), obtained from time-discretization of the asymptotically stable system (4.7), can be preserved by making δ suitably small when the time-discretization scheme is zero-stable (a property which either holds when (4.31) is satisfied or can be enforced by means of the arrangements of Remark 3). Further, under stability of (4.37), condition (4.46) can be always satisfied as well for suitably small values of κ. The idea is that the gain of the local Kalman filters should not be too large so that stability is preserved. Hence, in the considered setting, the above-reported procedure is guaranteed to succeed. Finally, it is an easy matter to verify that, under condition (4.46), the distributed finite-element Kalman filter with the modified correction step (4.47) leads to an asymptotically stable estimation error dynamics (the proof is analogous to the one of Theorem 2).

As a final remark, we point out that, once the original filtering problem has been recast in the form (4.38)-(4.39), the problem of designing the filter gains falls within the wider framework of partition-based distributed Kalman filtering (see [60] and the reference therein for an insight on this problem). The proposed solution has the advantage of requiring the tuning of one (or few) scalar quantities and hence is well-suited to keeping the computational load manageable even when the state vector has a large dimension (as it usually happens in the context of field estimation). Further, the proposed approach requires that only the estimates pertaining to the interfaces are exchanged between neighboring nodes, thus keeping the communication requirements as low as possible.

4.6 Numerical examples

This section provides numerical examples and relative results illustrating the effectiveness of the proposed distributed *finite element Kalman filter* presented in Section 4.4. Consider the transient *heat conduction* problem, introduced in Section 4.2 as a particular example of (4.1), in a thin polygonal metal plate with constant, homogeneous, and isotropic properties. Assuming that the thickness of the slab is considerably smaller than the planar dimensions, then the temperature can be assumed to be constant along the width direction, and the problem is reduced to two dimensions. Hence, the diffusion process in a thin plate is modelled by the 2D parabolic PDE $\partial x/\partial t - \lambda\left(\partial^2 x/\partial \xi^2 + \partial^2 x/\partial \eta^2\right) = 0$ with boundary condition $\mathcal{B}(x) = \alpha(\xi,\eta)\,\partial x/\partial \mathbf{n} + \beta(\xi,\eta)x$ such that $\alpha(\xi,\eta)\,\beta(\xi,\eta) \geq 0$, $\alpha(\xi,\eta) + \beta(\xi,\eta) > 0$, $\forall(\xi,\eta) \in \partial\Omega$. Notice that, $x(\xi,\eta,t)$ denotes the temperature as a function of time t and spatial variables $(\xi,\eta) \in \Omega$, $f = 0$ stands for no inner heat-generation, whereas $\lambda = 1.11 \times 10^{-4}\left[m^2/s\right]$ is the *thermal diffusivity* of copper at $25\,[^\circ C]$ (Table 12, Appendix 2 in [61]), assumed to be constant in time and space.

A network of $S = 23$ sensors (Fig. 4.2) located in the known positions $\mathbf{s}_i = [\xi_i, \eta_i]^T$ is assumed to collect point temperature measurements at regularly time-spaced instants $t_q = q\,T_s$, with $T_s = 100\,[s]$ and standard deviation of measurement noise $\sigma_v = 0.1\,[K]$. The considered sensor network has been chosen to guarantee local observability (assumption A2).

The Matlab PDE Toolbox is used to generate the triangular mesh (252 vertices, 436 elements) shown in Fig. 4.2 of size $b = 0.2[m]$ (defined as the length of the longest edge of the element), over the global 2D domain Ω. Next, as can be seen from Fig. 4.2, the domain under consideration is decomposed into $N = 8$ overlapping subdomains Ω_m, i.e. $\mathcal{N} = \{1,\ldots,8\}$, each being assigned to a *node* with local processing and communication capabilities. It is worth pointing out that domain decomposition comes with an appropriate partitioning of the original global mesh so that the resulting local grids actually match on the regions of overlap between subdomains.

Domain triangulation allows for a simple construction of basis functions $\{\phi_j(\xi,\eta)\}_{j=1}^n$, which are continuous piecewise polynomial functions, such that their value is unity in vertex j and vanishes at the remaining vertices, i.e.

$$\phi_j(\xi_i,\eta_i) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \qquad i,j = 1,2,...,n$$

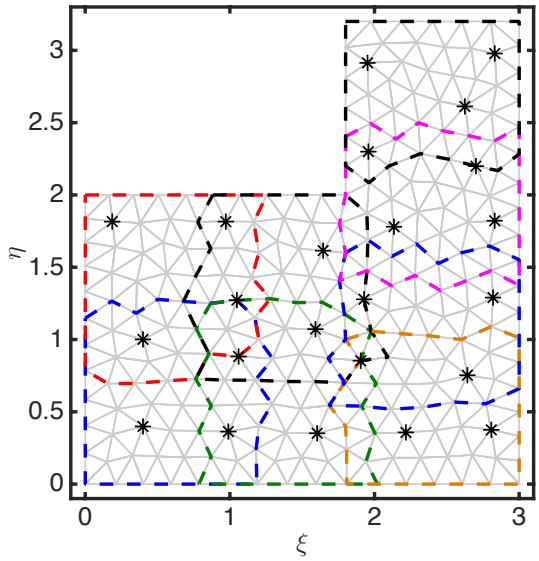
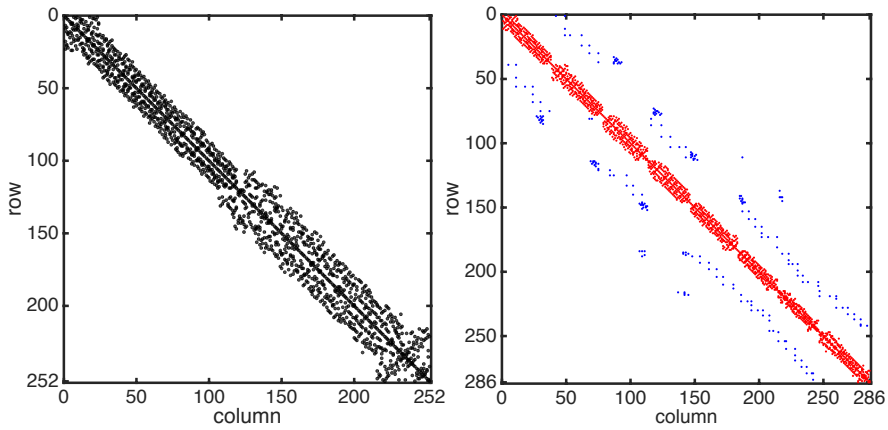


Figure 4.2: Global FE mesh (grid of solid lines) generated over Ω and domain decomposition into 8 overlapping subdomains (dashed polygons). The position of each sensor is denoted by $*$.

Here we use continuous piecewise linear functions defined on each element as $\psi_{\mathcal{E}}(\xi,\eta) = a + b\,\xi + c\,\eta$ with $(\xi,\eta) \in \mathcal{E}$ and $a, b, c \in \mathbb{R}$, so that each function is uniquely determined by its three nodal values $x_i = \psi_{\mathcal{E}}(\xi_i, \eta_i)$, $i \in \mathcal{E}$.

Basis functions are used off-line by the FE centralized filter and in the distributed setup for the element-by-element construction (described in Section 3.4) of matrices $\mathbf{S}$ and $\mathbf{M}$, introduced in (4.6). Then, the state dynamics of the centralized filter can be directly computed, whereas local estimators first need to extract matrices $\mathbf{M}^{mm}, \mathbf{S}^{mm}$ and $\mathbf{M}^{mj}$, $\mathbf{S}^{mj}$ in order to calculate $\mathbf{A}^m, \mathbf{A}^{mj}$ and $\bar{\mathbf{A}}^{mj}$ which finally provide the finite-dimensional model of temperature evolution in Ω_m through (4.22). Notice that these matrices are evaluated for a fixed sampling interval $\delta = T_s/L$, where L denotes the number of distributed prediction iterations L_q introduced in Section 4.4, here assumed constant in each sampling interval q. For a fair comparison between centralized and distributed approaches, a constant time integration interval $\Delta = 10\,[s]$ has been chosen for the centralized filter.

Notice that, being $\{\phi_j(\xi,\eta)\}_{j=1}^{n}$ functions with a small support defined by



(a) $\mathbf{S}$: 1626 nonzero elements

(b) $\tilde{\mathbf{S}}_D$: 1632 nonzero elements (red); $\tilde{\mathbf{S}}_F$: 223 nonzero elements (blue)

Figure 4.3: Sparsity pattern of 252×252 matrix $\mathbf{S}$ (a), and 286×286 matrix $\tilde{\mathbf{S}} = \tilde{\mathbf{S}}_D + \tilde{\mathbf{S}}_F$ (b).

the set of triangles sharing node j, the resulting *mass* and *stiffness* matrices will be sparse, with the same pattern shown in Fig. 4.3a. In Fig. 4.3b it can be seen how the structure of the stiffness matrix changes when considering the augmented system (4.25). The distributed pattern of the networked system is highlighted in Fig. 4.4, where $\tilde{\mathbf{A}}_D$ represents each subsystem as isolated, though affected by the evolution of neighbors through $\tilde{\mathbf{A}}_F$.

In the following experiments, both FE filters assume the initial temperature field of the plate uniform at $x_0(\xi, \eta) = 300\,[K]$, and the a-priori estimate taken as first guess $\hat{x}_{1|0}(\xi, \eta) = 305\,[K]$, with diagonal covariance $\mathbf{P}_{1|0} = 20\,\mathbf{I}$. Moreover, a zero-mean white noise process has been assumed, with covariance $\mathbf{Q} = \sigma_w^2\,\mathbf{I}$, where $\sigma_w = 3\,[K]$. Taking into consideration model uncertainty, the *ground truth* of the experiments is represented by a real process simulator implementing a finer mesh (915 vertices, 1695 elements) of size $b = 0.1$ instead of $b = 0.2$, running at a higher sampling rate ($1\,Hz$), and aware of the possibly time-varying boundary conditions of the system. On the other hand, both distributed and centralized filters have no knowledge of the real system boundary conditions, so they simply assume the plate adiabatic on each side.

The performance of the novel distributed *FE Kalman filter* has been

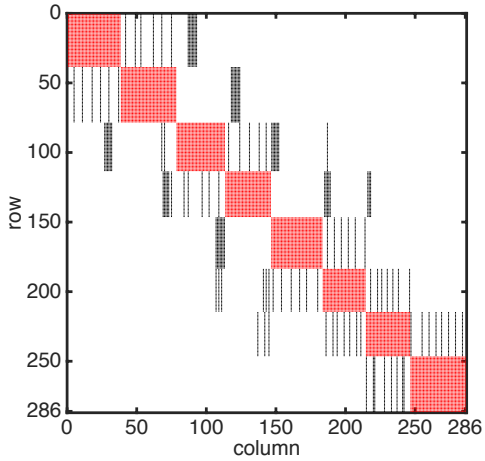


Figure 4.4: Sparsity pattern of $\tilde{\mathbf{A}}_D$ (red) and $\tilde{\mathbf{A}}_F$ (black).

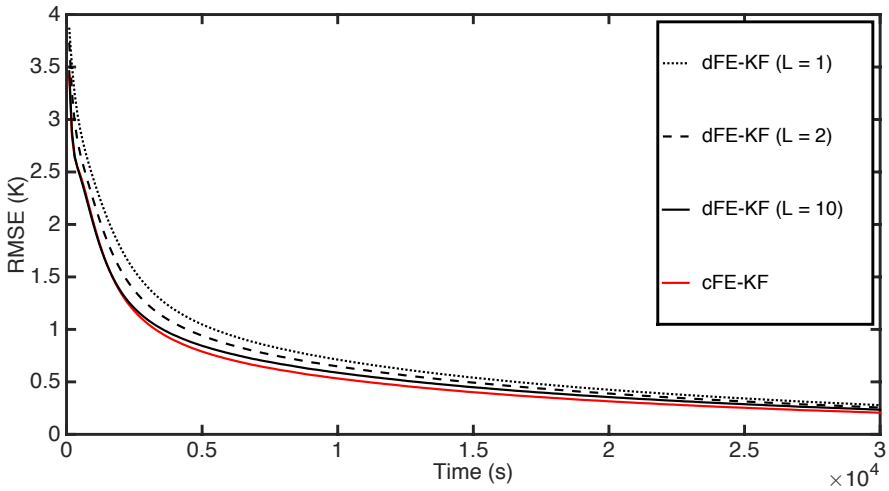


Figure 4.5: Scenario 1: Comparison of performance of centralized and distributed FE-KF ($\gamma = 1.1$).

evaluated in terms of Root Mean Square Error (RMSE) of the estimated temperature field, averaged over a set of about 300 sampling points uniformly spread within the domain Ω, and 500 independent Monte Carlo realizations.

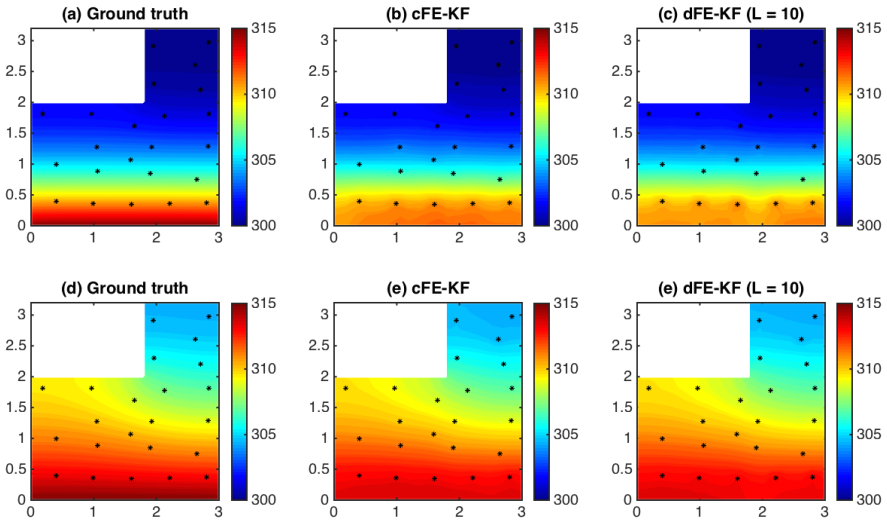


Figure 4.6: Scenario 1: *True* and estimated temperature fields in Kelvin (K) at time steps $q = 50$ (a,b,c) and $q = 200$ (d,e,f).

Scenario 1

In the first example, transient analysis is performed on a thin adiabatic L-shaped[3] plate (seen in Fig. 4.2) with a fixed temperature along the bottom edge. This is a problem with mixed boundary conditions, namely a non-homogeneous Dirichlet condition on the bottom edge of the plate $\partial\Omega_1$, i.e.

$$x = T_1 \qquad \text{on } \partial\Omega_1, \tag{4.48}$$

where $T_1 = 315\,[K]$, and natural homogeneous Neumann boundary conditions on the remaining insulated sides $\partial\Omega_2 = \partial\Omega \setminus \partial\Omega_1$, so that

$$\partial x/\partial \mathbf{n} = 0 \qquad \text{on } \partial\Omega_2. \tag{4.49}$$

[3]An L-shaped domain is traditionally used in boundary-value problems as a basic yet challenging example. It is the simplest geometry for which solutions to the wave equation cannot be expressed analytically, and thus numerical computation is necessary. Furthermore, the non-convex corner causes a singularity in the solution. This singularity limits the accuracy of finite difference methods with uniform grids. Anecdote: Cleve Moler, cofounder of MathWorks, used the L-shaped region as the primary example in his doctoral thesis. This is why MathWorks has adopted a modified surface plot of the first eigenfunction as the company logo.

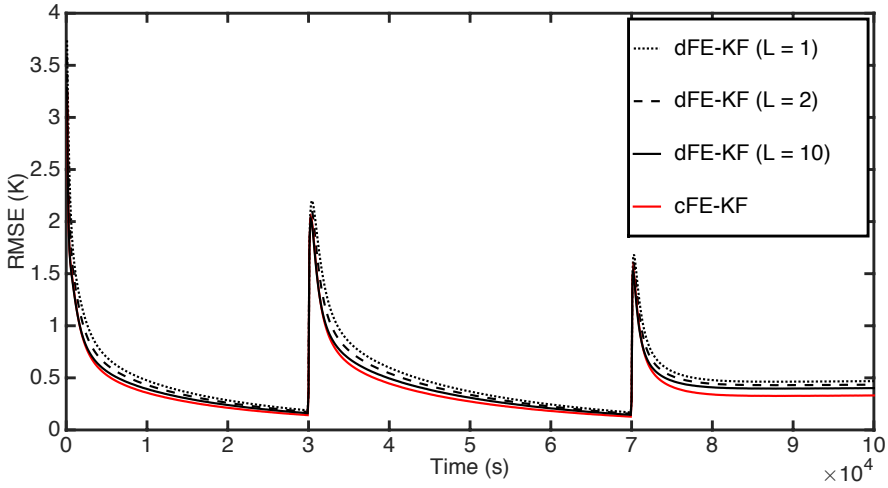


Figure 4.7: Scenario 2: Comparison of performance of centralized and distributed FE-KF ($\gamma = 1.1$).

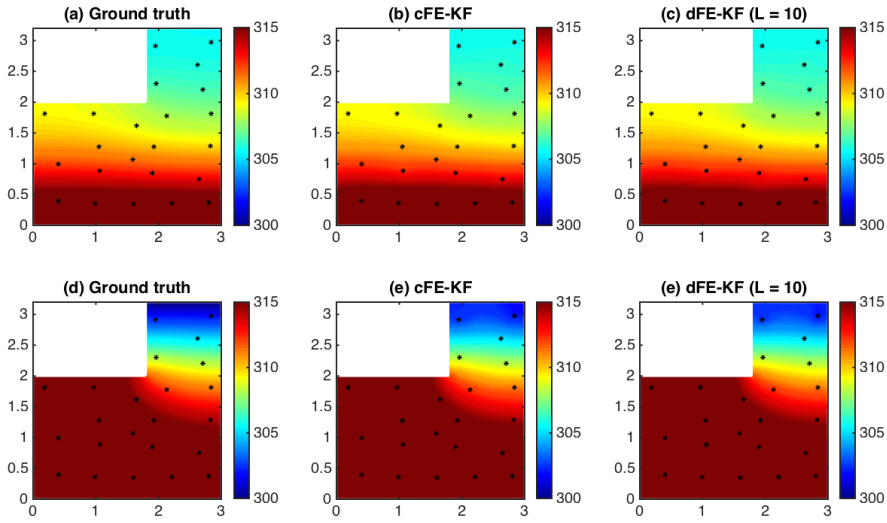


Figure 4.8: Scenario 2: *True* and estimated temperature fields in Kelvin (K) at time steps $q = 350$ (a,b,c) and $q = 900$ (d,e,f).

The duration of each Monte Carlo run is fixed to $3 \times 10^4\,[s]$ (300 samples).

Fig. 4.5 illustrates the performance comparison between centralized (cFE-KF) and distributed (dFE-KF) filters for $\gamma = 1.1$ and for three different values of the parameter L adopted in the distributed framework. First of all, it can be seen that both FE algorithms succeed in reconstructing the *true* field of the system based on fixed, point-wise temperature observations. Moreover, the performance of the distributed FE filters is very close, even for $L = 1$, to that of the centralized filter, which collects all the data in a central node. Last but not least, in the distributed setting the RMSE behaviour improves by increasing the number L of distributed prediction steps. This is true for certain values of γ, whereas for others the difference in performance is considerably reduced, as clearly presented in Fig. 4.9. Note that the covariance boosting factor used in (4.24) is set to $\gamma_L = \sqrt[L]{\gamma}$, $\forall L = 1, 2, 10$, in order to obtain a fairly comparable effect of covariance inflation after L distributed prediction steps for different distributed filters. Further insight on the performance of the proposed FE estimators is provided in Fig. 4.6, which shows the *true* and estimated temperature fields at two different time steps $q = 50$ and $q = 200$, obtained in a single Monte Carlo experiment by using cFE-KF and dFE-KF with $L = 10$.

Scenario 2

In the second experiment, two time-varying disturbances have been added in order to test the robustness of the proposed FE estimators in a more challenging scenario. To this end, different boundary conditions are considered. Specifically, a time-dependent Dirichlet condition (6.23) with $T_1 = 310\,[K]$ for time steps $q \in \{0, ..., 299\}$, and $T_2 = 320\,[K]$ for $q \in \{300, ..., 1000\}$, is set on all nodes of the bottom edge $\partial\Omega_1$. The top edge of the plate $\partial\Omega_3$ is first assumed adiabatic for $q \in \{0, ..., 699\}$, then the inhomogeneous Robin boundary condition

$$\lambda\, \partial x / \partial \mathbf{n} + \nu\, x = \nu\, x_e \quad \text{on } \partial\Omega_3 \tag{4.50}$$

is applied for $q \in \{700, ..., 1000\}$. This models a sudden exposure of the surface to a fluid, fixed at an external temperature $x_e = 300\,[K]$, through a uniform and constant convection heat transfer coefficient $\nu = 10\,[W/m^2K]$. The remaining edges $\partial\Omega_2$ where (6.24) holds, are assumed thermally insulated for the duration of the whole experiment, lasting $10^5\,[s]$ (1000 samples).

Performance of the proposed distributed filter has been evaluated for different values of L over 500 independent Monte Carlo runs and compared to

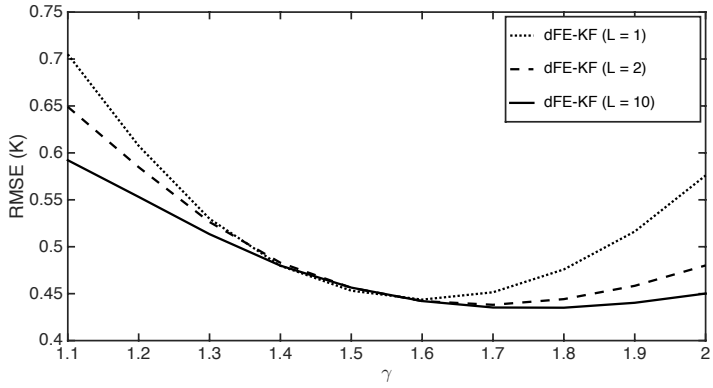


Figure 4.9: Scenario 1: Comparison of the mean value of the RMSE for different values of γ.

the behavior of the centralized FE Kalman filter. Simulation results, in Fig. 4.7, show that the proposed FE estimators provide comparable performance to the centralized filter, moreover the gap reduces as L increases. It is worth pointing out that the peaks appearing in the RMSE plot, displayed in Fig. 4.7, are due to the abrupt changes of the unknown boundary conditions, which cause considerable jumps of the estimation errors at time steps 300 and 700. Nevertheless, the filters under consideration manage to compensate for the lack of knowledge and effectively reduce the error, even if, due to persistent and cumulative disturbances on the inferred field profile, errors do not converge to zero. The original *ground truth* and the reconstructed fields are depicted in Fig. 4.8 for $q = 350$ and $q = 900$.

4.7 Conclusions

This chapter has dealt with the centralized and, especially, the decentralized estimation of a time-evolving and space-dependent field governed by a linear partial differential equation, given point-in space measurements of multiple sensors deployed over the area of interest. The originally infinite-dimensional filtering problem has been approximated into a finite-dimensional large-scale one via the *finite element* method and, further, a distributed approach inspired by the parallel Schwarz method for domain decomposition has allowed to nicely scale the overall problem complexity with respect to the number of

used processing nodes. Combining these two ingredients, a novel computationally efficient distributed finite-element Kalman filter has been proposed to solve in a decentralized and scalable fashion filtering problems involving distributed-parameter systems. Both numerical stability of the considered approximation scheme and exponential stability of the proposed distributed finite-element Kalman filter have been analysed. Simulation experiments have been presented in order to demonstrate the validity of the proposed approach. The results presented in this chapter can be applied to the estimation/localization of unknown diffusive sources, addressed in the next chapter.

Chapter 5

Unknown source in the field: detection and estimation

5.1 Introduction

The task of reconstructing the state of spatially distributed systems addressed in Chapter 4 becomes particularly challenging in the presence of unknown sources (e.g., of heat, polluting agents, toxic biochemical substances, etc.) of unknown intensity and position responsible for inducing and altering the target field. To this end, the source estimation problem is considered, which consists of detecting and localizing a concentrated diffusive source as well as estimating its intensity and monitoring the induced field.

The estimation of diffusive sources has recently received great attention within both the signal processing and control communities for at least two reasons: i) the low-cost availability of wireless sensors measuring the induced field (e.g. temperature, concentration) which can be deployed at low cost and in large number over the area to be monitored; ii) the strategic importance of such a task in homeland security, environmental and industrial monitoring, and situation awareness for a wide range of applications (e.g. fire detection, pollution monitoring, detection and localization of terrorist biochemical attacks, etc.). Two mainstream approaches to the problem of source estimation can be found in the literature. A first approach [62]- [63] models the source-induced field in steady-state, thus disregarding its transient time evolution, and therefore yields a parametric (static) estimation problem. It is worth to point out that, for slowly diffusing sources, this can

imply a very long, possibly unacceptable, detection/localization delay. Conversely, the second approach [64]- [65] is to explicitly take into account the spatiotemporal diffusion dynamics thus yielding a state (dynamic) estimation problem.

In order to allow faster detection/localization of slowly diffusing sources, here the latter, dynamic, approach will be followed. In particular, the spatiotemporal diffusion dynamics is modelled by an *advection-diffusion partial differential equation* with appropriate boundary conditions and a point (concentrated) source is considered. The *finite element* method is exploited for spatial discretization of the PDE. After time-discretization, the original infinite-dimensional boundary value problem is, therefore, transformed into a finite-dimensional, possibly large-scale, discrete-time linear system with state vector consisting of the field values in the vertices of the FE mesh, input vector representing the source intensity and input matrix depending on the source location. In this framework, we provide two major contributions to the source estimation problem. First, inspired by the classic notion of structural identifiability [66]- [67] considered as an a priori analysis for experiment design [67], this work defines the concept of *source identifiability*, i.e. the possibility of detecting the source and uniquely determining its position and intensity from available pointwise-in-time-and-space field measurements.

Specifically, system-theoretic conditions for identifiability are derived in terms of rank tests on suitable polynomial matrices for both cases in which the source intensity is regarded as an unknown input or is modeled as the output of an appropriate exosystem. Then, a multiple-model Kalman filtering approach to source estimation is undertaken by considering all hypotheses (modes) corresponding to the source location in any possible element of the FE mesh plus a further hypothesis accounting for the possible source absence. Both cases of motionless source with unknown position and of moving source are addressed, resorting to the static *Multiple Model* (MM) and, respectively, dynamic *Interacting Multiple Model* (IMM) algorithms. All the results of this chapter are presented in [68].

The rest of the chapter is organized as follows. Section 5.2 formulates the source estimation problem of interest. Section 5.3 derives a FE approximation of the original infinite-dimensional source diffusion model. Section 5.4 is devoted to the source identifiability analysis. Section 5.5 presents the multiple-model Kalman filtering approach to source estimation. Section

5.6 demonstrates the effectiveness of the proposed approach by means of a numerical example. Finally, Section 5.7 ends the chapter with concluding remarks.

5.2 Problem formulation

Let us consider a spatially distributed process governed by a PDE of the form (2.3)

$$\frac{\partial x}{\partial t} + \mathcal{L}(x) \;=\; f \quad \text{in } \Omega \tag{5.1}$$

with possibly inhomogeneous boundary condition

$$\mathcal{B}(x) \;=\; g \quad \text{on } \partial\Omega\,. \tag{5.2}$$

where: $x(\mathbf{p},t)$ is the space-time dependent scalar field of interest, defined over the space-time domain $\Omega \times \mathbb{R}$; the space domain Ω is supposed to be bounded and with smooth boundary $\partial\Omega$; $\mathbf{p} \in \Omega$ denotes the d-dimensional ($d \in \{1,2,3\}$) position vector; $t \in \mathbb{R}_+$ denotes time; $\mathcal{L}(\cdot)$ and $\mathcal{B}(\cdot)$ are the *advection-diffusion* and, respectively, *Robin* operators defined as follows

$$\mathcal{L}(x) \;\triangleq\; -\lambda\nabla^2 x + \mathbf{v}^T\nabla x \tag{5.3}$$

$$\mathcal{B}(x) \;\triangleq\; \partial x/\partial\mathbf{n} + \beta x; \tag{5.4}$$

λ is a constant diffusion coefficient; $\mathbf{v}(\mathbf{p})$ is the advection velocity vector; $\beta(\mathbf{p}) \geq 0$ is a, possibly space-dependent, coefficient; $\partial x/\partial\mathbf{n} = \mathbf{n}^T\nabla x$, $\mathbf{n}$ being the outward pointing unit normal vector of the boundary $\partial\Omega$; $g(\mathbf{p},t)$ is the forcing term acting on the boundary $\partial\Omega$; $f(\mathbf{p},t)$ is the point source input modeled as (see Section 3.4.4)

$$f(\mathbf{p},t) \;=\; \begin{cases} 0, & \text{if no source exists} \\ u(t)\;\delta\left(\mathbf{p}-\mathbf{p}^0(t)\right), & \text{otherwise} \end{cases} \tag{5.5}$$

with unknown *intensity* $u(t)$ and *position* $\mathbf{p}^0(t) \in \Omega$. The aim is to detect the source presence and jointly estimate $u(t), \mathbf{p}^0(t), x(\mathbf{p},t)$ given measurements

$$y_{k,i} \;=\; h_i\left(x\left(\mathbf{s}_i,t_k\right)\right) + v_{k,i} \tag{5.6}$$

provided by sensors $i \in \mathcal{S} \triangleq \{1,\ldots,S\}$, located at positions $\mathbf{s}_i \in \Omega$, at discrete sampling instants t_k, $k \in \mathcal{Z}_+ = \{1,2,\ldots\}$, such that $0 < t_1 < t_2 < \cdots$.

The above stated dynamic estimation problem is clearly infinite-dimensional. It will be shown in the next section how it can be approximated into a finite-dimensional one by exploiting the *finite element* (FE) method.

5.3 Finite-element approximation

As previously described in Section 2.2, equation (5.1) with boundary condition (5.2) can be recast into the following weak form:

$$\int_\Omega \frac{\partial x}{\partial t}\psi \, d\mathbf{p} \; - \; \lambda \int_\Omega \; \nabla^2 x \; \psi \, d\mathbf{p} + \int_\Omega \mathbf{v}^T \; \nabla x \; \psi \, d\mathbf{p} = \int_\Omega f\psi \, d\mathbf{p} \qquad (5.7)$$

where $\psi(\mathbf{p})$ is a generic space-dependent weight function. By applying Green's identity and thanks to (5.2), one obtains:

$$\int_\Omega \frac{\partial x}{\partial t} \, \psi \, d\mathbf{p} + \lambda \int_\Omega \nabla^T x \nabla \psi \, d\mathbf{p} + \int_\Omega \mathbf{v}^T \; \nabla x \; \psi \, d\mathbf{p} \\ - \lambda \int_{\partial\Omega} (g - \beta x) \, \psi \, d\mathbf{p} = \int_\Omega f \, \psi \, d\mathbf{p} \qquad (5.8)$$

Following the finite-element method introduced in Section 3.1, by subdividing the domain Ω into a suitable set of non overlapping elements and by defining a suitable set of basis functions $\phi_j(\mathbf{p})$, $j = 1, \ldots, n$, on them, it is possible to write the approximation (3.6) of the unknown function $x(\mathbf{p}, t)$ as

$$x(\mathbf{p}, t) = \sum_{j=1}^{n} \phi_j(\mathbf{p}) \, x_j(t) \; = \; \boldsymbol{\phi}^T(\mathbf{p}) \, \mathbf{x}(t) \qquad (5.9)$$

where: $x_j(t)$ is the unknown expansion coefficient of function $x(\mathbf{p}, t)$ relative to time t and basis function $\phi_j(\mathbf{p})$; $\boldsymbol{\phi}(\mathbf{p}) \stackrel{\triangle}{=} col\{\phi_j(\mathbf{p})\}_{j=1}^n$ and $\mathbf{x}(t) \stackrel{\triangle}{=} col\{x_j(t)\}_{j=1}^n$. The finite elements define a FE mesh with vertices $\mathbf{p}_j \in \Omega, j = 1, \ldots, n$.

By choosing the test function ψ equal to the selected basis functions, the Galerkin method, introduced in Section 3.1 is applied and the following equation is obtained

$$\underbrace{\left[\int_\Omega \boldsymbol{\phi}(\mathbf{p})\,\boldsymbol{\phi}^T(\mathbf{p})d\mathbf{p}\right]}_{\mathbf{M}}\dot{\mathbf{x}}(t)+\underbrace{\left[\lambda\int_\Omega \nabla\boldsymbol{\phi}(\mathbf{p})\,\nabla\boldsymbol{\phi}^T(\mathbf{p})d\mathbf{p}\right]}_{\mathbf{S}_\alpha}\mathbf{x}(t) \tag{5.10}$$
$$+\underbrace{\left[\int_\Omega \boldsymbol{\phi}(\mathbf{p})\;\mathbf{v}^T(\mathbf{p})\,\nabla\boldsymbol{\phi}^T(\mathbf{p})\,d\mathbf{p}\right]}_{\mathbf{S}_v}\mathbf{x}(t)+\underbrace{\left[\lambda\int_{\partial\Omega}\beta(\mathbf{p})\,\boldsymbol{\phi}(\mathbf{p})\,\boldsymbol{\phi}^T(\mathbf{p})\,d\mathbf{p}\right]}_{\mathbf{S}_\beta}\mathbf{x}(t)$$
$$=\underbrace{\left[\int_\Omega \boldsymbol{\phi}(\mathbf{p})\,\delta(\mathbf{p}-\mathbf{p}^0)d\mathbf{p}\right]}_{\boldsymbol{\phi}(\mathbf{p}^0)}u(t)+\underbrace{\left[\lambda\int_{\partial\Omega}\boldsymbol{\phi}(\mathbf{p})\,\boldsymbol{\phi}^T(\mathbf{p})d\mathbf{p}\right]}_{\mathbf{S}_g}\mathbf{g}(t)$$

where in the integrals on the contour $\partial\Omega$ it is assumed that the various functions are the restrictions to $\partial\Omega$ of the original functions defined over Ω, and that for $g(\mathbf{p},t)$ an expansion akin to (6.25) holds, i.e. $g(\mathbf{p},t) = \sum_{j=1}^{n}\phi_j(\mathbf{p})\,g_j(t) = \boldsymbol{\phi}^T(\mathbf{p})\,\mathbf{g}(t)$.

It is evident how all integrals in the LHS (5.10) depend only on basis functions and can be computed *a priori*. In particular, matrices $\mathbf{S}_\alpha, \mathbf{S}_\beta, \mathbf{S}_g, \mathbf{S}_v$ can be computed as discussed in Section 3.4.

Then, by regularly discretizing in time (5.10) with sampling interval δt (i.e. $t_k = k\,\delta t$) and approximating the time derivative with the finite difference $\dot{\mathbf{x}}(t) \simeq (\mathbf{x}_{k+1}-\mathbf{x}_k)/\delta t$, the following discrete-time linear descriptor system is obtained:

$$\mathbf{M}\left(\frac{\mathbf{x}_{k+1}-\mathbf{x}_k}{\delta t}\right)+(\mathbf{S}_\alpha+\mathbf{S}_\beta+\mathbf{S}_v)\,\mathbf{x}_{k+1}\simeq\boldsymbol{\phi}(\mathbf{p}^0)u_k+\mathbf{S}_g \tag{5.11}$$

from which one obtains the discrete-time model

$$\mathbf{x}_{k+1}=\mathbf{A}\mathbf{x}_k+\mathbf{B}(\mathbf{p}^0)u_k+\mathbf{b}_k+\mathbf{w}_k \tag{5.12}$$

where:

$$\begin{aligned} u_k &= u\,(t_{k+1})\\ \mathbf{A} &= \left[\mathbf{I}+\delta t\;\mathbf{M}^{-1}\,(\mathbf{S}_\alpha+\mathbf{S}_\beta+\mathbf{S}_v)\right]^{-1}\\ \mathbf{B}(\mathbf{p}^0) &= \left[\mathbf{I}+\delta t\;\mathbf{M}^{-1}\,(\mathbf{S}_\alpha+\mathbf{S}_\beta+\mathbf{S}_v)\right]^{-1}\mathbf{M}^{-1}\delta t\;\boldsymbol{\phi}(\mathbf{p}^0)\\ \mathbf{b}_k &= \left[\mathbf{I}+\delta t\;\mathbf{M}^{-1}\,(\mathbf{S}_\alpha+\mathbf{S}_\beta+\mathbf{S}_v)\right]^{-1}\mathbf{M}^{-1}\delta t\;\mathbf{S}_g \end{aligned} \tag{5.13}$$

and $\mathbf{w}_k$ is a process disturbance taking into account also the space-time discretization errors. For a quantitative characterization of such errors, the

reader can refer to [69]. Notice that, in equation (6.29), the intensity u_k and the position $\mathbf{p}^0$ of the point source input are unknown and hence must be estimated together with the state vector $\mathbf{x}_k$. As for the intensity u_k, different models are possible:

(a) u_k is treated as an *unknown input* for which no information on the possible time evolution is available [70, 71];

(b) u_k is unknown but a dynamic model for its time evolution is available, i.e., it is supposed that u_k is generated as the output of an auxiliary linear system (called *exosystem*).

$$\begin{aligned} \mathbf{q}_{k+1} &= \mathbf{F}\,\mathbf{q}_k + \boldsymbol{\zeta}_k \\ u_k &= \mathbf{H}\,\mathbf{q}_k \end{aligned}$$

where $\mathbf{q}_k$ is the exosystem state and $\boldsymbol{\zeta}_k$ the disturbance input. Here, without loss of generality the pair $(\mathbf{F}, \mathbf{H})$ is supposed to be observable.

For instance, if it is known that the unknown intensity u_k can vary slowly with time, its time evolution can be modeled as a random walk by letting $\mathbf{F} = \mathbf{H} = 1$ and taking $\boldsymbol{\zeta}_k$ as a zero-mean white noise. Of course, the preferable model depends on the situation under consideration and, specifically, on possible physical insights on the source intensity.

5.4 Source identifiability

In this section, an analysis on the possibility of correctly identifying the unknown source location $\mathbf{p}^0$ and intensity u_k is provided both in cases (a) and (b). As usually done in observability/identifiability analysis, the study is carried out in the ideal noise-free case by supposing that the measurements $\mathbf{y}_k$ are generated by

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}(\mathbf{p}^0)u_k \\ \mathbf{y}_k &= \mathbf{C}\,\mathbf{x}_k \end{aligned} \tag{5.14}$$

Accordingly, in case (b), the intensity dynamics will be supposed to be noise-free by letting

$$\begin{aligned} \mathbf{q}_{k+1} &= \mathbf{F}\,\mathbf{q}_k \\ u_k &= \mathbf{H}\,\mathbf{q}_k \end{aligned} \tag{5.15}$$

Notice that the known input $\mathbf{b}_k$ is not considered in equation (5.14) since, thanks to the superposition principle for linear systems, its contribution is immaterial to the source identification problem.

Let now $\mathbb{S}$ be the linear space of all real-valued sequences on the non-negative integers $\mathbb{Z}_+$ and let us denote by $\mathcal{U} \subseteq \mathbb{S}$ the set of all possible time-evolutions of the intensity u_k which are consistent with the available model. Clearly, in case (a) we simply have $\mathcal{U} = \mathbb{S}$, while in case (b) $\mathcal{U}$ is the set of all the possible output behaviors of system (5.15). Let also denote by $\mathbf{y}(\mathbf{x}_0, \mathbf{p}^0, u)$ the output behavior of system (5.14) when the initial state is $\mathbf{x}_0$, the source location is $\mathbf{p}^0$ and the source intensity evolves according to the sequence u. The following notion can be introduced.

Definition 2. *The point source is said to be* identifiable *if*

$$\mathbf{y}(\mathbf{x}_0, \mathbf{p}^0, u) \neq \mathbf{y}(\bar{\mathbf{x}}_0, \bar{\mathbf{p}}^0, \bar{u}) \tag{5.16}$$

for any pair of source locations $\mathbf{p}^0, \bar{\mathbf{p}}^0 \in \Omega$*, any pair of initial states* $\mathbf{x}_0, \bar{\mathbf{x}}_0$*, and any nonzero pair of intensity sequences* $u, \bar{u} \in \mathcal{U}$*, with* $(\mathbf{p}^0, u) \neq (\bar{\mathbf{p}}^0, \bar{u})$*.*

In words, identifiability of the point source corresponds to the fact that different sources (in terms of location and intensity) always give rise to different output behaviors or, equivalently, corresponds to the invertibility of the mapping from $(\mathbf{p}^0, u)$ to the output sequence $\mathbf{y}$. Notice that, in the above definition, we exclude the trivial case in which both u and $\bar{u}$ are zero, but we allow that either u or $\bar{u}$ be zero so as to account for the possibility of distinguishing between presence or absence of the source input.

Since only the observable part of (5.14) influences the output behavior $\mathbf{y}$, it is convenient to consider an alternative representation of system (5.14) obtained by means of the Kalman observability decomposition. This amounts to considering an invertible transformation matrix $\mathbf{T}$ such that

$$\begin{aligned} &\mathbf{T}^{-1}\mathbf{A}\mathbf{T} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{0} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}, \quad \mathbf{T}^{-1}\mathbf{B} = \begin{bmatrix} \mathbf{B}_1(\mathbf{p}^0) \\ \mathbf{B}_2(\mathbf{p}^0) \end{bmatrix} \\ &\mathbf{C}\mathbf{T} = [\mathbf{C}_1 \quad \mathbf{0}]\,. \end{aligned} \tag{5.17}$$

with $(\mathbf{A}_{11}, \mathbf{C}_1)$ observable. Of course, when $\mathbf{C}$ is chosen so that $(\mathbf{A}, \mathbf{C})$ is observable, we have $\mathbf{A}_{11} = \mathbf{A}$ and $\mathbf{C}_1 = \mathbf{C}$. All the results of this section will refer to the system (5.14) or (5.14)-(5.15) in the case (b), under the

observability decomposition (5.17). By exploiting standard results on observability and left invertibility of linear systems, the following theorem can now be stated.

Theorem 7. *For any pair of distinct source locations* $\mathbf{p}^0, \bar{\mathbf{p}}^0 \in \Omega$*, consider the polynomial matrix*

$$\mathbf{\Psi}(z, \mathbf{p}^0, \bar{\mathbf{p}}^0) = \begin{bmatrix} z\mathbf{I} - \mathbf{A}_{11} & \mathbf{B}_1(\mathbf{p}^0) & \mathbf{B}_1(\bar{\mathbf{p}}^0) \\ \mathbf{C}_1 & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

with $z \in \mathbb{C}$*. Then, the following facts hold:*

(i) in case (a), the source is identifiable if and only if

$$\operatorname{rank}\left\{\mathbf{\Psi}(z, \mathbf{p}^0, \bar{\mathbf{p}}^0)\right\} = n_o + 2. \tag{5.18}$$

for any $z \in \mathbb{C}$ *and for any* $\mathbf{p}^0, \bar{\mathbf{p}}^0 \in \Omega$ *with* $\mathbf{p}^0 \neq \bar{\mathbf{p}}^0$*. Here* n_o *is the dimension of the observable part of (5.14);*

(ii) in case (b), the source is identifiable if and only if the rank condition (5.18) holds for any $z \in \operatorname{sp}\{\mathbf{F}\}$ *and for any* $\mathbf{p}^0, \bar{\mathbf{p}}^0 \in \Omega$ *with* $\mathbf{p}^0 \neq \bar{\mathbf{p}}^0$*. Here,* $\operatorname{sp}\{\mathbf{F}\}$ *stands for the spectrum of the matrix* $\mathbf{F}$*.* □

Proof: Let $\mathbf{x}_{1,k} \in \mathbb{R}^{n_0}$ be the state vector of the observable part of (5.14). Then, we have

$$\begin{aligned} \mathbf{x}_{1,k+1} &= \mathbf{A}_{11}\mathbf{x}_{1,k} + \mathbf{B}_1(\mathbf{p}^0)u_k \\ \mathbf{y}_k &= \mathbf{C}_1\,\mathbf{x}_{1,k} \end{aligned}$$

Further, thanks to linearity, the overall output behavior can be decomposed as the sum of the natural (free) response $\mathbf{y}^{\mathrm{n}}(\mathbf{x}_{1,0})$ and the forced response $\mathbf{y}^{\mathrm{f}}(\mathbf{p}^0, u)$. As a consequence the identifiability condition (5.16) can be written as $\mathbf{y}^{\mathrm{n}}(\mathbf{x}_{1,0}) + \mathbf{y}^{\mathrm{f}}(\mathbf{p}^0, u) \neq \mathbf{y}^{\mathrm{n}}(\bar{\mathbf{x}}_{1,0}) + \mathbf{y}^{\mathrm{f}}(\bar{\mathbf{p}}^0, \bar{u})$ or equivalently as $\mathbf{y}^{\mathrm{n}}(\mathbf{x}_{1,0} - \bar{\mathbf{x}}_{1,0}) + \mathbf{y}^{\mathrm{f}}(\mathbf{p}^0, u) - \mathbf{y}^{\mathrm{f}}(\bar{\mathbf{p}}^0, \bar{u}) \neq 0$. It is now immediate to see that source identifiability is equivalent to requiring that the system

$$\begin{aligned} \mathbf{x}_{1,k+1} &= \mathbf{A}_{11}\mathbf{x}_{1,k} + \mathbf{B}_1(\mathbf{p}^0)u_k + \mathbf{B}_1(\bar{\mathbf{p}}^0)\bar{u}_k \\ \mathbf{y}_k &= \mathbf{C}_1\,\mathbf{x}_{1,k} \end{aligned} \tag{5.19}$$

does not exhibit a zero output trajectory when its input $(u, \bar{u})$ is different from 0. In case (a), when the input trajectories can be arbitrary, this latter condition is clearly equivalent to the left invertibility of system (5.19). Then,

fact (i) follows from well-known results [72] by noting that $\mathbf{\Psi}(z, \mathbf{p}^0, \bar{\mathbf{p}}^0)$ corresponds to the Rosenbrock's system matrix of (5.19). As for case (b), since both u_k and $\bar{u}_k$ are supposed to be generated by an exosystem of the form (5.15), we can consider the augmented system resulting from the cascade interconnection of (5.19) with

$$\begin{aligned} \begin{bmatrix} \mathbf{q}_{k+1} \\ \bar{\mathbf{q}}_{k+1} \end{bmatrix} &= \begin{bmatrix} \mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{F} \end{bmatrix} \begin{bmatrix} \mathbf{q}_k \\ \bar{\mathbf{q}}_k \end{bmatrix} \\ \begin{bmatrix} u_k \\ \bar{u}_k \end{bmatrix} &= \begin{bmatrix} \mathbf{H} & \mathbf{0} \\ \mathbf{0} & \mathbf{H} \end{bmatrix} \begin{bmatrix} \mathbf{q}_k \\ \bar{\mathbf{q}}_k \end{bmatrix} \end{aligned} \tag{5.20}$$

Then, the identifiability condition corresponds to the fact that the autonomous system (5.19)-(5.20) does not exhibit a zero output trajectory when its initial condition is different from 0, or in other words to the fact that (5.19)-(5.20) is observable. In order to study the observability of (5.19)-(5.20), it is convenient to consider an alternative representation of such a system in terms of the Polynomial Matrix Description (PMD)

$$\begin{bmatrix} z\mathbf{I} - \mathbf{A}_{11} & \mathbf{B}_1(\mathbf{p}^0) & \mathbf{B}_1(\bar{\mathbf{p}}^0) \\ \mathbf{0} & \Delta(z) & 0 \\ \mathbf{0} & 0 & \Delta(z) \end{bmatrix} \begin{bmatrix} \mathbf{x}_{1,k} \\ u_k \\ \bar{u_k} \end{bmatrix} = \mathbf{0}$$

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{C}_1 & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{1,k} \\ u_k \\ \bar{u_k} \end{bmatrix}$$

where $\Delta(z) = \det(z\mathbf{I} - \mathbf{F})$. In fact, we can now use the PBH observability test for PMDs (see [73], page 562) in order to conclude that (5.20) is observable if and only if

$$\operatorname{rank} \begin{bmatrix} \mathbf{C}_1 & \mathbf{0} & \mathbf{0} \\ z\mathbf{I} - \mathbf{A}_{11} & \mathbf{B}_1(\mathbf{p}^0) & \mathbf{B}_1(\bar{\mathbf{p}}^0) \\ \mathbf{0} & \Delta(z) & 0 \\ \mathbf{0} & 0 & \Delta(z) \end{bmatrix} = n_o + 2$$

for any $z \in \mathbb{C}$. In turn, the latter condition is equivalent to requiring that (5.18) holds for any $z \in \operatorname{sp}\{\mathbf{F}\}$ (recall that the roots of $\Delta(z)$ are the eigenvalues of $\mathbf{F}$). □

Notice that the only difference between cases (a) and (b) is that in the latter case only the values of z corresponding to eigenvalues of the exosystem

have to be considered in the rank test. Further, in this case, the identifiability condition can be rephrased in terms of system gains as follows.

Corollary 1. *Consider case (b) and suppose that* $\mathrm{sp}\{\mathbf{A}\}$ *and* $\mathrm{sp}\{\mathbf{F}\}$ *are disjoint. Then, the source is identifiable if and only if*

$$\mathrm{rank}\left[\ \mathbf{C}(z\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}(\mathbf{p}^0),\quad \mathbf{C}(z\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}(\bar{\mathbf{p}}^0)\ \right]=2$$

for any $z\in\mathrm{sp}\{\mathbf{F}\}$ *and any* $\mathbf{p}^0,\bar{\mathbf{p}}^0\in\Omega$ *with* $\mathbf{p}^0\neq\bar{\mathbf{p}}^0$. □

From Theorem 7 and Corollary 1, it is evident that at least 2 sensors are needed in order to guarantee distinguishability of two source locations $\mathbf{p}^0,\bar{\mathbf{p}}^0$. However, this is just a lower bound since, in general, a larger number of sensors may be needed. The main drawback of the derived rank conditions is that they have to be satisfied for any pair of source locations $\mathbf{p}^0,\bar{\mathbf{p}}^0$ belonging to the space domain Ω. This means that, in order to verify whether a given set of sensors ensures source identifiability, an infinite number of conditions have to be checked, which clearly makes the test impractical. A first, approximated, approach to sidestep such a difficulty would amount to restricting the attention only to a finite number of possible source locations, for example corresponding to the vertices $\mathbf{p}_i$ of the FE mesh. A second, more theoretically sound approach consists in looking for alternative conditions by exploiting the structure of the system matrices resulting from application of the FE method.

With this respect, consider the most typical situation in which the elements are chosen as d-dimensional simplexes (i.e., intervals when $d=1$, triangles when $d=2$, or tetrahedrons when $d=3$) and the basis functions $\phi_j(\mathbf{p})$ are piecewise linear. Further, let $\mathcal{E}_1,\ldots,\mathcal{E}_v$ denote the elements of the considered mesh and, for a generic element $\mathcal{E}_j$, let $\mathcal{V}_j\subset\{1,\ldots,n\}$ be the set of indices corresponding to the vertices of $\mathcal{E}_j$. Notice that, for a d-dimensional simplex, $\mathcal{V}_j$ contains exactly $d+1$ elements. Then, in this case, the input matrix $\mathbf{B}(\mathbf{p}^0)$ can be written as a convex combination of the input matrices associated to the vertices of the element $\mathcal{E}_j$ containing $\mathbf{p}^0$, i.e.,

$$\mathbf{B}(\mathbf{p}^0)=\sum_{i\in\mathcal{V}_j}\omega_i(\mathbf{p}^0)\mathbf{B}(\mathbf{p}_i) \tag{5.21}$$

where $\omega_i(\mathbf{p}^0)\geq 0$ and $\sum_{i\in\mathcal{V}_j}\omega_i(\mathbf{p}^0)=1$. This makes it possible to derive sufficient conditions for source identifiability to be checked only for each pair

of elements. To this end, for a generic element $\mathcal{E}_j$, let $\mathbf{B}(\mathcal{E}_j)$ denote the matrix obtained by juxtaposition of the column matrices $\mathbf{B}(\mathbf{p}_i)$ with $i \in \mathcal{V}_j$, and let $\mathbf{B}_1(\mathcal{E}_j)$ be obtained in an analogous way from the matrices $\mathbf{B}_1(\mathbf{p}_i)$ with $i \in \mathcal{V}_j$. Then, the following result holds.

Theorem 8. *Let the input matrices be as in (5.21) and, for any pair of distinct elements $\mathcal{E}_j, \mathcal{E}_\ell$ of the mesh, consider the polynomial matrix*

$$\mathbf{\Psi}_{j\ell}(z) = \begin{bmatrix} z\mathbf{I} - \mathbf{A}_{11} & \mathbf{B}_1(\mathcal{E}_j) & \mathbf{B}_1(\mathcal{E}_\ell) \\ \mathbf{C}_1 & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

with $z \in \mathbb{C}$. Then, the following facts hold:

(i) in case (a), the source is identifiable if

$$\operatorname{rank}\{\mathbf{\Psi}_{j\ell}(z)\} = n_o + |\mathcal{V}_j \cup \mathcal{V}_\ell| \tag{5.22}$$

for any $z \in \mathbb{C}$ and for any $\mathcal{E}_j, \mathcal{E}_\ell$ with $j \neq \ell$;

(ii) in case (b), the source is identifiable if the rank condition (5.22) holds for any $z \in \operatorname{sp}\{\mathbf{F}\}$ and for any $\mathcal{E}_j, \mathcal{E}_\ell$ with $j \neq \ell$. □

Proof: Consider a pair of distinct elements $\mathcal{E}_j, \mathcal{E}_\ell$ and suppose that condition (5.22) holds for some z. Clearly, this implies that

$$\operatorname{rank}\begin{bmatrix} \mathbf{B}_1(\mathcal{E}_j) & \mathbf{B}_1(\mathcal{E}_\ell) \end{bmatrix} = |\mathcal{V}_j \cup \mathcal{V}_\ell|\,. \tag{5.23}$$

Notice also that, for any $\mathbf{p}^0 \in \mathcal{E}_j$, one has $\mathbf{B}_1(\mathbf{p}^0)u_k = \mathbf{B}_1(\mathcal{E}_j)\boldsymbol{\eta}_k$ with $\boldsymbol{\eta}_k = \operatorname{col}\big(\omega_i(\mathbf{p}^0)u_k, i \in \mathcal{V}_j\big)$. Hence, it is an easy matter to verify that condition (5.23) implies that one can have $\mathbf{B}_1(\mathbf{p}^0)u_k = \mathbf{B}_1(\bar{\mathbf{p}}^0)\bar{u}_k$, or equivalently $\mathbf{B}_1(\mathcal{E}_j)\boldsymbol{\eta}_k = \mathbf{B}_1(\mathcal{E}_\ell)\bar{\boldsymbol{\eta}}_k$, if and only if $\mathbf{p}^0 = \bar{\mathbf{p}}^0$ and $u_k = \bar{u}_k$. Then, fact (i) follows from the observation that, when condition (5.22) holds for any $z \in \mathbb{C}$, the output behavior of the system

$$\begin{aligned} \mathbf{x}_{1,k+1} &= \mathbf{A}_{11}\mathbf{x}_{1,k} + \mathbf{B}_1(\mathcal{E}_j)\boldsymbol{\eta}_k - \mathbf{B}_1(\mathcal{E}_\ell)\bar{\boldsymbol{\eta}}_k \\ \mathbf{y}_k &= \mathbf{C}_1\,\mathbf{x}_{1,k} \end{aligned} \tag{5.24}$$

can be zero for any k if and only if $\mathbf{B}_1(\mathcal{E}_j)\boldsymbol{\eta}_k = \mathbf{B}_1(\mathcal{E}_\ell)\bar{\boldsymbol{\eta}}_k$ for any k, which as pointed out above implies $\mathbf{p}^0 = \bar{\mathbf{p}}^0$ and $u_k = \bar{u}_k$. As for fact (ii), it can be proved along the same lines of the proof of Theorem 1 by considering the cascade interconnection of (5.24) with (5.20). □

Notice that in condition (5.22), the term $|\mathcal{V}_j \cup \mathcal{V}_\ell|$ represents the number of distinct vertices in $\mathcal{E}_j \cup \mathcal{E}_\ell$. As a consequence, $|\mathcal{V}_j \cup \mathcal{V}_\ell| \leq 2(d+1)$ where the equality holds if and only if $\mathcal{E}_j$ and $\mathcal{E}_\ell$ have no common vertices. Hence, Theorem 8 suggests that, in the d-dimensional case, $2(d+1)$ sensors may be needed in order to have source identifiability.

Remark 4. *The rank condition of Theorem 8 provides a computationally feasible way to verify whether a given set of sensors guarantees source identifiability, since it has to be checked only for a finite number of cases, i.e., for each pair of distinct elements. Additional insights can be gained by recalling that system (5.14) can also be written as a linear descriptor system [see (5.11)]*

$$\begin{array}{rcl} \mathbf{E}\mathbf{x}_{k+1} & = & \mathbf{M}\mathbf{x}_k + \delta t\, \boldsymbol{\phi}(\mathbf{p}^0) u_k \\ \mathbf{y}_k & = & \mathbf{C}\,\mathbf{x}_k \end{array} \tag{5.25}$$

where the matrices $\mathbf{E}$, $\mathbf{M}$ *and* $\boldsymbol{\phi}(\mathbf{p}^0)$ *have very specific structures. In particular, the non-zero elements of the matrices* $\mathbf{E}$, $\mathbf{M}$ *correspond to connected vertices in the graph associated to the FE mesh. Further, for any mesh vertex* $\mathbf{p}_i$, $\boldsymbol{\phi}(\mathbf{p}_i)$ *coincides with* $\mathbf{e}_i$, *the* i*-th vector of the canonical basis. If it is assumed that system (5.25) is observable, it can be easily shown that in condition (5.22) the matrix* $\boldsymbol{\Psi}_{j\ell}(z)$ *can be replaced by*

$$\tilde{\boldsymbol{\Psi}}_{j\ell}(z) = \begin{bmatrix} z\mathbf{E} - \mathbf{M} & \boldsymbol{\phi}(\mathcal{E}_j) & \boldsymbol{\phi}(\mathcal{E}_\ell) \\ \mathbf{C} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

where $\boldsymbol{\phi}(\mathcal{E}_j)$ *denotes the matrix obtained by juxtaposition of the column matrices* $\boldsymbol{\phi}(\mathbf{p}_i) = \mathbf{e}_i$ *with* $i \in \mathcal{V}_j$. *Hence, if it is further assumed that the sensor locations coincide with vertices of the FE mesh, it is possible to relate the rank of the matrix* $\tilde{\boldsymbol{\Psi}}_{j\ell}(z)$ *to the topology of the FE mesh as well as to the sensor locations. In fact, results on the rank of matrices of the form of* $\tilde{\boldsymbol{\Psi}}_{j\ell}(z)$ *for systems like (5.25) defined over graphs have been recently obtained in the literature [74, 75]. While such results are only generic (i.e., they hold for almost all the dynamical systems compatible with the graph topology, but counterexamples can exist), nevertheless they provide useful guidelines on where to place the sensors inside the domain* Ω. *The interested reader is referred to [74, 75] for further details on this issue.*

5.5 Source estimation

Based on the fact that distinct source locations correspond to different process behaviors, and thanks to finite element approximation, the key idea of the proposed source estimation algorithms relies upon the assumption that system (6.29), at each time step, obeys to one of a finite set of diffusion models. To this end, the *Multiple Model* (MM) approach [76] provides a suitable tool, as it accounts for the uncertainty about the system input location, assuming that the real evolution of the system follows one of the possible modes of operation. In particular, the idea is to match each hypothesis of source being located in a generic element of the mesh, to a distinct operating mode of the system. This makes it possible to run in parallel a finite number of mode-matched Kalman filters, one for each element of the generated mesh. Hence, each mode j associated to the hypothesis that a point source is located in $\mathbf{p}^0$, contained in element $\mathcal{E}_j$, is characterised by the following mode-matched model

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}(\mathcal{E}_j)\boldsymbol{\omega}(\mathbf{p}^0)u_k + \mathbf{w}_k, \quad j = 1, 2, ..., v \\ \mathbf{y}_k &= \mathbf{C}\,\mathbf{x}_k + \mathbf{v}_k \end{aligned} \tag{5.26}$$

where $\boldsymbol{\omega}(\mathbf{p}^0)$ is the $(d{+}1)$-dimensional column vector of coefficients $\omega_i(\mathbf{p}^0)$, $i \in \mathcal{V}_j$, introduced in Section 5.4. It is worth noting that in order to be able to detect new sources, an extra *source-free* operating mode, based on the assumption that no point source is present, needs to be added to the set of possible modes of the MM algorithm. Thus, including the no-source mode, from now on denoted as $\bar{v}$ for $\bar{v} = v + 1$, and recalling the mesh generates elements $\mathcal{E}_1, ..., \mathcal{E}_v$, the set of possible modes becomes $\bar{v}$-dimensional.

Bearing in mind the previous points about source detection and localisation, the additional joint source intensity and state estimation can be carried out by constructing an augmented system for the MM estimator, as the aggregate of the original system (5.26) and a suitable model for the unknown input time evolution. To this end, let us introduce $\boldsymbol{\eta}_k = \mathrm{col}\left(\omega_i(\mathbf{p}^0)u_k,\, i \in \mathcal{V}_j\right)$ so that $\mathbf{B}(\mathbf{p}^0)u_k = \mathbf{B}(\mathcal{E}_j)\boldsymbol{\eta}_k$. Then, the augmented system for a generic mode j originated from (5.26), takes the following form for $j = 1, 2, ..., v$

$$\begin{aligned} \begin{bmatrix} \mathbf{x}_{k+1} \\ \boldsymbol{\eta}_{k+1} \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & \mathbf{B}(\mathcal{E}_j) \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \boldsymbol{\eta}_k \end{bmatrix} + \begin{bmatrix} \mathbf{w}_k \\ \boldsymbol{\zeta}_k \end{bmatrix} \\ \mathbf{y}_k &= [\mathbf{C} \quad \mathbf{0}] \begin{bmatrix} \mathbf{x}_k \\ \boldsymbol{\eta}_k \end{bmatrix} + \mathbf{v}_k \end{aligned} \tag{5.27}$$

whereas, for $j = \bar{v}$

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{w}_k \\ \mathbf{y}_k &= \mathbf{C}\,\mathbf{x}_k + \mathbf{v}_k \end{aligned} \tag{5.28}$$

Note that in (5.27) the dynamics of the source intensity u_k are assumed to follow a discrete-time random walk. As a result, the joint source and field estimation problem can be reduced to the joint estimation of $\mathbf{x}_k$ and $\boldsymbol{\eta}_k$ for each time step k. Assuming the source may move within the domain, i.e. the correct operating mode may switch over time, it is convenient to employ a dynamic MM technique, e.g. *Interacting Multiple Model* (IMM) [76], which allows for mode jumps, limiting at the same time the number of hypotheses to the number of filters. Otherwise, the source of interest is assumed motionless, i.e. fixed in an unknown position of the monitored area. In this case a *static* MM estimator, which assumes there is a single operating mode throughout the entire process, can be suitably employed to address the considered estimation problem. Next, a brief summary of a centralised approach to the static MM called *Finite Element Static Multiple Model* (FE-SMM) and dynamic FE-IMM is shown, wherein the measurements of all sensors are collected and jointly processed in the correction step of each mode-matched filter. For further details on multiple-model filtering algorithms, the reader is referred to [76].

***A*. Static case: FE-SMM**

The FE-SMM algorithm runs a bank of $\bar{v}$ FE Kalman filters matched to the modes j in (5.27), for $1 \leq j < \bar{v}$, or (5.28) for $j = \bar{v}$. Each filter updates the state estimate, covariance and mode probability relative to mode j by processing the entire set of gathered measurements $y_{k,i}$, $i \in \mathcal{S}$. Assume available the state estimate $\hat{\mathbf{x}}_{k-1|k-1}$, the covariance $\mathbf{P}_{k-1|k-1}$, and the mode probabilities μ^j_{k-1} at time step $k-1$, then the estimator recursion is the following.

1. ***Mode-matched filtering:*** a mode-matched Kalman filter for each $j \in \{1, ..., \bar{v}\}$ carries out the prediction and correction steps, processing the entire set of gathered measurements $y_{k,i}$, $i \in \mathcal{S}$. The bank of filters produces mode-conditioned state estimates $\hat{\mathbf{x}}^j_{k|k}$ and covariances $\mathbf{P}^j_{k|k}$ for each mode. In addition, assuming Gaussian noises, the mode likelihoods are evaluated as follows

$$\Lambda^j_k = \mathcal{N}(\zeta^j_k; 0; \mathfrak{S}^j_k), \quad j = 1, ..., \bar{v} \tag{5.29}$$

where $\zeta_k^j \triangleq \mathbf{y}_k^j - \mathbf{C}\hat{\mathbf{x}}_{k|k-1}^j$ is the innovation at time k of mode j, and $\mathfrak{S}_k^j$ the associated covariance.

2. ***Mode probability update:*** mode probabilities are updated by means of the mode likelihoods as follows

$$\mu_k^j = \frac{1}{c}\,\Lambda_k^j\,\mu_{k-1}^j \tag{5.30}$$

where $c = \sum_{i=1}^{\bar{v}} \Lambda_k^i\,\mu_{k-1}^i$ is the normalization constant.

Once initialised, mode-matched filters run independently with no interaction. At the end of each cycle, the mode with maximum probability will be considered as the operating one. As a consequence, the associated mode-conditioned estimate will be directly used for field and source intensity estimation. Further, exploiting the structure of the FE approximation, the source location can be estimated as a convex combination of the position of the vertices of the element $\mathcal{E}_j$ matched to the estimated operating mode, i.e.

$$\begin{aligned} \hat{\mathbf{p}}^0 &= \sum_{i\in\mathcal{V}_j} \hat{\omega}_i\,\mathbf{p}_i \\ \hat{\omega}_i &= \frac{\hat{\eta}_k^i}{\hat{u}_k}, \quad i \in \mathcal{V}_j, \quad \hat{u}_k = \sum_{i\in\mathcal{V}_j} \hat{\eta}_k^i \end{aligned} \tag{5.31}$$

***B.* Dynamic case: FE-IMM**

The idea is to run an IMM estimator for the augmented system (5.27) with mode-to-mode transitions modelled by means of a homogeneous Markov chain with known constant transition probabilities

$$\pi_{ij} = \text{prob}\left(\nu_k = i \,|\, \nu_{k-1} = j\right), \qquad i,j \in \{1,2,...,\bar{v}\} \tag{5.32}$$

where ν_k represents the *modal state* (i.e. the mode in operation) at time k. Differently from the static MM algorithm, at the beginning of each sampling interval, the $\bar{v}$ filters interact in a mixing step which produces the so-called *mixed initial conditions*, i.e. different combinations of the previous model-conditioned estimates and associated covariances. It must also be noted that, since the source-free mode $\bar{v}$ has a different (lower) state dimension with respect to modes $j \neq \bar{v}$, the state estimate and covariance of the former must be padded with zeros in order to match the higher dimension of the latter during the mixing step. Hence the recursion of the dynamic estimator is the following.

1. ***Calculation of the mixing probabilities:*** in order to calculate the mixed initial conditions, the mixing probabilities are first updated as follows

$$\mu^{i|j}_{k-1|k-1} \triangleq \frac{\pi_{ji}\,\mu^{i}_{k-1}}{\sum_{\ell=1}^{\bar{v}} \pi_{j\ell}\,\mu^{\ell}_{k-1}} \tag{5.33}$$

2. ***Mixing:*** the mixed state estimates and covariances for modes $j \neq \bar{v}$, are computed as follows

$$\begin{aligned}
\hat{\mathbf{x}}^{0j}_{k-1|k-1} &= \sum_{i=1}^{v} \hat{\mathbf{x}}^{i}_{k-1|k-1}\mu^{i|j}_{k-1|k-1} + \hat{\mathbf{x}}^{\bar{v}}_{k-1|k-1}\mu^{\bar{v}|j}_{k-1|k-1} \\
\mathbf{P}^{0j}_{k-1|k-1} &= \sum_{i=1}^{v} \mu^{i|j}_{k-1|k-1}\left[\mathbf{P}^{i}_{k-1|k-1} + \tilde{\mathbf{x}}^{ij}(\tilde{\mathbf{x}}^{ij})^T\right] + \\
&\quad + \mu^{\bar{v}|j}_{k-1|k-1}\left[\mathbf{P}^{\bar{v}}_{k-1|k-1} + \tilde{\mathbf{x}}^{\bar{v}j}(\tilde{\mathbf{x}}^{\bar{v}j})^T\right]
\end{aligned} \tag{5.34}$$

where $\hat{\mathbf{x}}^{\bar{v}}_{k-1|k-1} = [(\bar{\mathbf{x}}^{\bar{v}}_{k-1|k-1})^T\ \mathbf{0}^T]^T$ and

$$\mathbf{P}^{\bar{v}}_{k-1|k-1} = \begin{bmatrix} \bar{\mathbf{P}}^{\bar{v}}_{k-1|k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

denote respectively the source-free model-conditioned augmented state estimate and covariance at time step $k-1$. Note that the state estimate $\bar{\mathbf{x}}^{\bar{v}}_{k-1|k-1}$ of mode $\bar{v}$ has been simply augmented assuming a zero-intensity source. Furthermore, $\tilde{\mathbf{x}}^{ij}$ in (5.34) is the so called *spread of the means* $\tilde{\mathbf{x}}^{ij} = \hat{\mathbf{x}}^{i}_{k-1|k-1} - \hat{\mathbf{x}}^{0j}_{k-1|k-1}$. Mixed initial conditions for the no-source filter are computed by extracting from the mode-conditioned state estimates $\hat{\mathbf{x}}^{i}_{k-1|k-1}$ and covariances $\mathbf{P}^{i}_{k-1|k-1}$, $i \neq \bar{v}$, the lower dimensional $\bar{\mathbf{x}}^{i}_{k-1|k-1}$ and $\bar{\mathbf{P}}^{i}_{k-1|k-1}$ corresponding to the original state. Then, the mixing is computed as above

$$\begin{aligned}
\bar{\mathbf{x}}^{0\bar{v}}_{k-1|k-1} &= \sum_{i=1}^{\bar{v}} \bar{\mathbf{x}}^{i}_{k-1|k-1}\mu^{i|\bar{v}}_{k-1|k-1} \\
\bar{\mathbf{P}}^{0\bar{v}}_{k-1|k-1} &= \sum_{i=1}^{\bar{v}} \mu^{i|\bar{v}}_{k-1|k-1}\left[\bar{\mathbf{P}}^{i}_{k-1|k-1} + \tilde{\mathbf{x}}^{i\bar{v}}(\tilde{\mathbf{x}}^{i\bar{v}})^T\right]
\end{aligned} \tag{5.35}$$

3. ***Mode-matched filtering:*** this step is identical to the one discussed above for the static MM algorithm.

4. ***Mode probability update:*** assuming that mode transitions are modelled by (5.32), the mode probabilities are evaluated as follows

$$\mu_k^j = \frac{1}{c_1} \Lambda_k^j \sum_{i=1}^{\bar{v}} \pi_{ji} \, \mu_{k-1}^i \tag{5.36}$$

where $c_1 = \sum_{j=1}^{\bar{v}} \Lambda_k^j \sum_{i=1}^{\bar{v}} \pi_{ji} \, \mu_{k-1}^i$ is the normalization constant.

It is worth mentioning that the proposed source estimator uses $v+1$ Kalman filters, where v is the number of elements of the FE mesh. More precisely, v out of the $v+1$ Kalman filters have an $(n+d+1)$-dimensional state, where n is the number of vertices of the FE grid and $d \in \{1,2,3\}$ the dimension of the domain of interest, while the remaining Kalman filter associated to the no-source mode has n-dimensional state. Since $v = O(n)$, $d \ll n$ and $|\mathcal{S}| \ll n$, each Kalman filter has $O(n^3)$ complexity, and the overall computational complexity is $O(n^4)$. Since modern computers are characterized by computing power in the order of Gigaflops, problems with hundreds or thousands of state variables can be handled, depending on the required sampling rate. It is also worth pointing out that the modal Kalman filters can be run in a fully parallel fashion, being mindful, however, that the IMM mixing step requires an exchange of information between the bank of filters.

5.6 Numerical examples

The proposed FE-IMM, described in Section 5.5, is validated via simulations. Consider a scenario concerning a moving source estimation for a diffusion process governed by the 2D case of (5.1) with

$$\mathcal{L}(x) = -\lambda \left(\partial^2 x / \partial \xi^2 + \partial^2 x / \partial \eta^2 \right)$$

and mixed boundary conditions (see Section 2.6)

$$\partial x / \partial \mathbf{n} + \beta_1 x = g_1 \quad \text{on } \partial\Omega_1 \tag{5.37}$$

$$\partial x / \partial \mathbf{n} = 0 \quad \text{on } \partial\Omega_2 \tag{5.38}$$

This model describes, for instance, transient contaminant transport in water bodies. Parameters $\beta_1 = \frac{\nu}{\lambda}$ and $g_1 = \frac{\nu}{\lambda} x_e$ are such that (5.37) describes an outward/inward diffusive flux across $\partial\Omega_1$ (boundary 10 in Fig.5.1), proportional to the concentration difference $x - x_e$ between internal and external

environments (external concentration $x_e = 0$ is assumed known, $\nu = 1$). The homogeneous Neumann boundary condition (5.38) assumes there is no flux across $\partial\Omega_2$, i.e. it is considered impermeable to the contaminant. Further, (5.1) implicitly assumes λ is constant, here taken as $\lambda = 0.1$. A network of 6 sensors is randomly deployed inside the spatial 2D domain Ω to sample the concentration field of interest, with sampling interval $T_s = 1\,[s]$ and standard deviation of measurement noise $\sigma_v = 0.005$. As shown in Fig. 5.1, a triangular mesh (116 nodes, 196 elements) is generated over Ω for the finite-dimensional approximation of the monitored field. As *true* initial field condition, we consider $\mathbf{x}_0 = \mathbf{0}$, whereas the estimator starts from $\hat{\mathbf{x}}_{1|0} = 10\,\mathbf{1}$ with covariance $\mathbf{P}_{1|0} = 100^2\,\mathbf{I}$. Moreover, the source intensity estimate is initialised as $\hat{u}_{1|0} = 10$, with associated initial covariance matrix $\mathbf{P}^u_{1|0} = 100^2\,\mathbf{I}$, while the *true* average intensity of the source is 30. The standard deviation of the process noise for the simulator is set as $\sigma_w = 1.5$, whereas for the MM filters $\hat{\sigma}_w = 5$. The variance of the disturbance input is set to $\sigma^2_\zeta = 0.04$. All simulation results are averaged over 100 Monte Carlo trials.

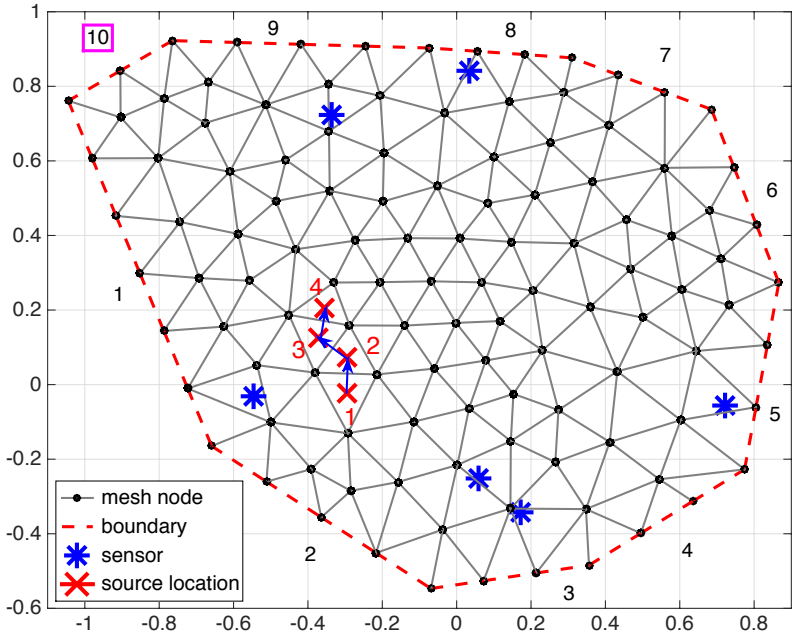
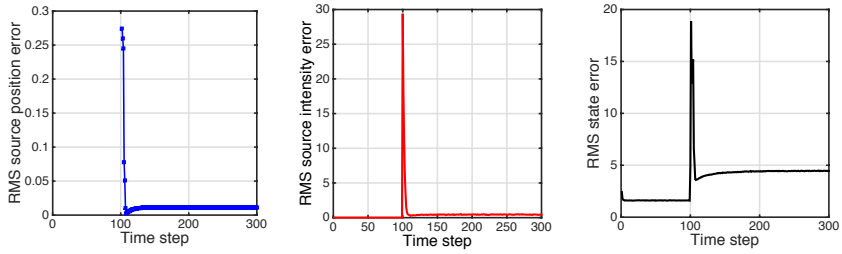


Figure 5.1: Static case: fixed source in 1. Dynamic case: source moves from 1 to 4 in an area monitored by 6 sensors.



(a) Static case: RMS source position error.

(b) Static case: RMS source intensity error.

(c) Static case: RMS state error.

Figure 5.2: Simulation results in the case of static source.

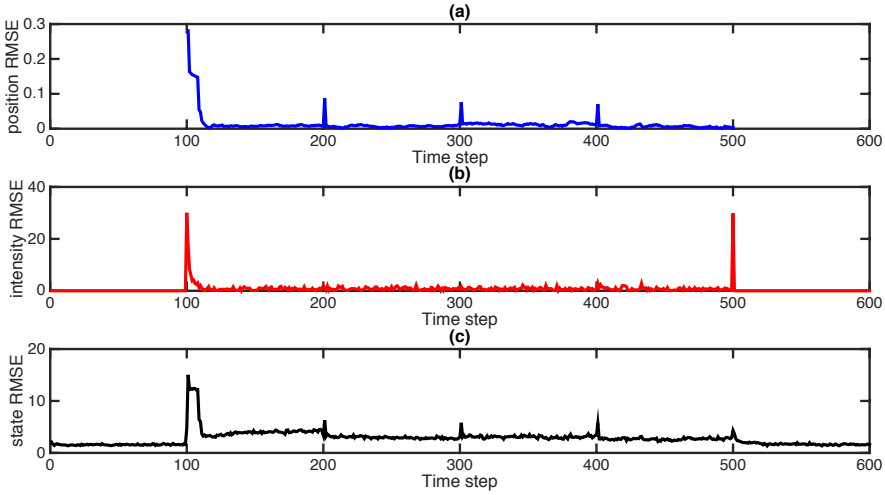


Figure 5.3: Simulation results in the case of dynamic source.

5.6.1 Static source: FE-SMM

In the first scenario, no source is active when the 6 sensing devices start the monitoring activity. After 100 $[s]$ a fixed source located in $\mathbf{p}^0 = [-0.296, -0.0237]^T$ (location 1 in Fig. 5.1) activates. The total simulation time is 300 $[s]$ (300 samples), and the standard deviation of the intensity increment $\boldsymbol{\zeta}_k$ is chosen as $\sigma_\zeta = 0.02$ for the simulator and $\hat{\sigma}_\zeta = 1$ for the filters. Simulation results

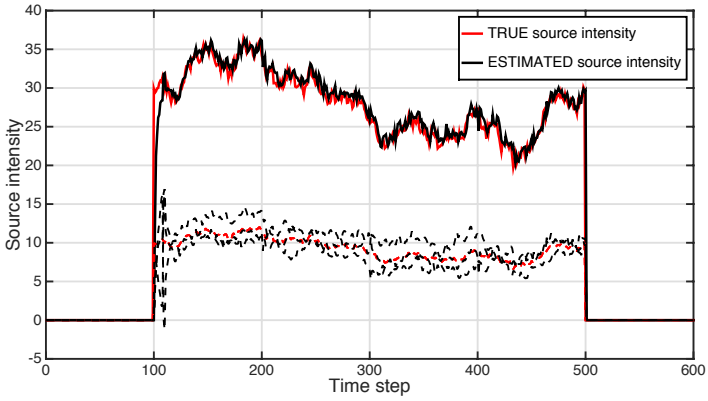


Figure 5.4: Dynamic case: *true* (u_k) and estimated ($\hat{u}_k$) source intensity (solid lines), and *true* and estimated ($\hat{\eta}_k^i$, $i \in \mathcal{V}_j$) intensity components (dotted lines).

relative to a static source are shown in Fig. 5.2.

5.6.2 Dynamic source: FE-IMM

In the second scenario the source, activated at time $100\,[s]$, is moving along the path $1 \to 2 \to 3 \to 4$ (see Fig. 5.1), sojourning 100 time steps in each intermediate location, before turning off at time 500. From 500 to 600 the simulation continues with no source. Jump probabilities are $\pi_{ii} = 0.85\,\forall i = 1, ..., \bar{v}$, $\pi_{i\bar{v}} = 0.05, i = 1, ..., v$, while the remaining 0.1 probability is equally distributed among all elements $\mathcal{E}_j$ adjacent to $\mathcal{E}_i$. The Root Mean Square Errors (RMSE) relative to the source position, intensity and source-induced field, are reported, as performance indices, in Fig. 5.3. These are obtained via comparison of the estimated quantities with a simulated system (*ground truth*), which implements a finer mesh (427 nodes, 784 elements) and runs at a higher sampling frequency of $10\,Hz$, in order to take into account model uncertainty. Results show that both the estimators succeed in localizing the unknown source (Fig. 5.3a) and estimating the corresponding intensity (Fig. 5.3b) in a very short time and with remarkable accuracy. The field estimation error (Fig. 5.3c) presents visible peaks in correspondence of either source activation or change of position, but it promptly stabilizes once the input has been detected. Fig. 5.4 displays, for the maximum probability

element $\mathcal{E}_j$, the estimates $\hat{u}_k = \sum_{i \in \mathcal{V}_j} \hat{\eta}_k^i$ and $\hat{\eta}_k^i = \hat{\omega}_i \, \hat{u}_k$ from which it is possible to obtain via (5.31) the estimate of the source location.

5.7 Conclusions

In this chapter, the problem of detecting a diffusive point source and jointly estimating its location, intensity and induced field from pointwise-in-time-and-space field measurements of sensors deployed over the monitored area, has been addressed. This has been made possible by combining the finite-element method for discretising in space the diffusion dynamics and the multiple-model Kalman filtering approach.

Chapter 6

Dynamic field estimation over binary sensor networks

6.1 Introduction

In monitoring applications *wireless sensor networks* (WSNs) must usually operate with a large number of geographically distributed devices, characterized by limited power as well as limited communication and computational resources. As reported in [77], simple, inexpensive devices deployed in large numbers are likely to be the cutting-edge technology for WSNs, as it is not practical to rely on sophisticated sensors with large power supply and communication demands. Motivated by the challenges that the progress in WSN technology will pose, the aim of this chapter is to investigate how to perform real-time dynamic field estimation using minimum-cost binary sensor networks, which convey a minimal amount of information.

Binary sensors, whose output just indicates whether the noisy measurement of the sensed variable (analog measurement) exceeds or not a given threshold, have increasingly been employed in monitoring and control applications [78]- [79]. The idea is that by a multitude of low-cost and low-resolution sensing devices it is possible to achieve the same estimation accuracy that a few (possibly a single one) expensive high-resolution sensors could provide, with significant practical benefits in terms of ease of sensor deployment and minimization of communication requirements. The fact that a binary (threshold) measurement just conveys a minimal amount (i.e. a single bit) of information, while implying communication bandwidth savings

and consequently greater energy efficiency, makes of paramount importance to fully exploit the little available information by means of smart estimation algorithms. In this respect, some work has recently addressed system identification [78]- [80], parameter [81]- [82] or state estimation [83]- [79] with binary measurements by following either a deterministic [78]- [80], [83]- [84] or a probabilistic [81]- [82], [85]- [79] approach.

In a deterministic context, the available information is essentially concentrated at the sampling instants in which some binary measurement signal has switched value [83, 86]. As shown in [86], some additional information can be exploited in the other (non switching) sampling instants by penalizing values of the estimated quantity such that the corresponding predicted measurement is on the opposite side, with respect to a binary sensor reading, far away from the threshold. Nevertheless, it is clear that there is no or very little information available for estimation purposes whenever no or very few binary sensor switchings occur. Hence, a possible way to achieve high estimation accuracy is to have many binary sensors measuring the same variable with different thresholds as this would clearly increase the number of switchings, actually emulating, when the number of sensors tends to infinity, the availability of a single continuous-valued (analog) measurement.

Conversely, following a probabilistic approach, binary sensor readings could be exploited to infer information about the probability distribution of the variable of interest. To clarify this point, let us assume that a very large number of binary sensors of the same type (i.e. measuring the same variable with the same threshold) be available and the distribution of their measurement noise (e.g. Gaussian with zero mean and given standard deviation) be known. Then, thanks to the numerosity of measurements, the relative frequency of 1 (or 0) values occurring in the sensor readings could be considered as a reasonable estimate of the probability that the sensed variable is above (or below) the threshold and this, in turn, exploiting the knowledge of the measurement noise distribution allows to extract information about the location of the value of the sensed variable with respect to the threshold. If, for example, it is found that the binary measurement is equal to 1 for 70% of the sensors and Gaussian measurement noise is hypothesized, it turns out that the expected measurement of the sensed variable is above the threshold of an amount equal to 0.525 times the standard deviation of the measurement noise. Notice that if the sensors are noiseless, they all provide either 0 or 1 output and, paradoxically, in this case minimal information, i.e.

that the sensed variable takes values in a semi-infinite interval (either below or above the threshold), is extracted from the set of binary measurements. The above arguments suggest that, adopting a probabilistic approach to estimation using binary measurements, the presence of measurement noise can be a helpful source of information. In other words, it can be said that noise-aided procedures can be devised for estimation with binary measurements by exploiting the fact that the measurement noise randomly shifts the analog measurement thus making possible to infer statistical information on the sensed variable.

Relying on the above stated *noise-aided* paradigm, this chapter presents a novel approach to recursive estimation of the state of a discrete-time dynamical system given binary measurements. The proposed approach is based on a *moving-horizon* (MH) approximation of the *Maximum A-posteriori Probability* (MAP) estimation and extends previous work [81]- [87] concerning parameter estimation to recursive state estimation. A further contribution is to show that for a linear system the optimization problem arising from the MH-MAP formulation turns out to be convex and, hence, practically feasible for real-time implementation. All the results of this chapter are presented in [88].

The rest of the chapter is organized as follows. Section 6.2 introduces the MAP problem formulation of state estimation with binary measurements. Section 6.3 presents a MH approximation of MAP estimation, referred to as MH-MAP algorithm, and analyzes the properties of the resulting optimization problem. Section 6.4 discusses a possible application of the proposed approach to the dynamic estimation of a diffusion field from binary pointwise-in-space-and-time field measurements. Section 6.5 presents simulation results relative to the dynamic field estimation case-study.

6.2 MAP state estimation with binary sensors

The following notation will be used throughout the chapter: $col(\cdot)$ denotes the matrix obtained by stacking its arguments one on top of the other; $diag(m^1, \dots, m^q)$ denotes the diagonal matrix with diagonal entries $m^1, \dots, m^q$; 0_n, 1_n indicate the $n-$dimensional vectors, respectively, with all zero and unit entries.

Let us consider the problem of recursively estimating the state of the

discrete-time nonlinear dynamical system

$$x_{t+1} = f(x_t, u_t) + w_t \tag{6.1}$$

$$z_t^i = h^i(x_t) + v_t^i, \quad i = 1, \ldots, l \tag{6.2}$$

from a set of measurements provided by binary sensors

$$y_t^i = g^i(z_t^i) = \begin{cases} 1, & \text{if } z_t^i \geq \tau^i \\ 0, & \text{if } z_t^i < \tau^i \end{cases} \tag{6.3}$$

where $x_t \in \mathbb{R}^n$ is the state to be estimated, $u_t \in \mathbb{R}^m$ is a known input, and τ^i is the threshold of the $i-$th binary sensor. For the sake of simplicity, we define $z_t = col\left(z_t^i\right)_{i=1}^l \in \mathbb{R}^l$ and $y_t = col\left(y_t^i\right)_{i=1}^l \in \mathbb{R}^l$. The vector $w_t \in \mathbb{R}^n$ is an additive disturbance affecting the system dynamics which accounts for uncertainties in the mathematical model, while $v_t = col\left(v_t^i\right)_{i=1}^l \in \mathbb{R}^l$ is the measurement noise vector.

Let $\mathcal{N}(\mu, \Sigma)$ denote as usual the normal distribution with mean μ and variance Σ. The statistical behaviour of the system is characterized by

$$x_0 \sim \mathcal{N}(\overline{x}_0, P^{-1}), \quad w_t \sim \mathcal{N}(0, Q^{-1}), \quad v_t \sim \mathcal{N}(0, R) \tag{6.4}$$

where: $R = diag(r^1, \ldots, r^p)$; $\mathbb{E}[w_j w_k'] = 0$ and $\mathbb{E}[v_j v_k'] = 0$ if $j \neq k$; $\mathbb{E}[w_j v_k'] = 0$, $\mathbb{E}[w_j x_0'] = 0$, $\mathbb{E}[v_j x_0'] = 0$ for any j, k. Notice from (6.2)-(6.3) that sensor i produces a binary measurements $y_t^i \in \{0, 1\}$ depending on whether the noisy system output z_t^i is below or above the threshold τ^i.

According to the available probabilistic description (6.4), the problem of estimating the state of system (6.1) under the binary measurement model (6.2)-(6.3) is formulated hereafter in a Bayesian framework by resorting to a maximum a posteriori probability (MAP) criterion. In the remainder of this section, as a preliminary step, the *full-information* MAP state estimation problem is formulated.

To this end, notice that each binary measurement y_t^i provides intrinsically relevant information on the state x_t which can be taken into account by means of the a posteriori probabilities $p(y_t^i|x_t)$. In particular, the binary measurement y_t^i is a Bernoulli random variable such that, for any binary sensor i and any time instant t, the a posteriori probability $p(y_t^i|x_t)$ is given by

$$p(y_t^i|x_t) = p(y_t^i = 1|x_t)^{y_t^i} \; p(y_t^i = 0|x_t)^{1-y_t^i} \tag{6.5}$$

where

$$p(y_t^i = 1|x_t) = F^i(\tau^i - h^i(x_t)) \tag{6.6}$$

and $p(y_t^i = 0|x_t) = 1 - p(y_t^i = 1|x_t) \triangleq \Phi^i(\tau^i - h^i(x_t))$. The function $F^i(\tau^i - h^i(x_t))$ is the complementary cumulative distribution function (CDF) of the random variable $\tau^i - h^i(x_t)$. Since $v_t^i \sim \mathcal{N}(0, r^i)$, the conditional probability $p(y_t^i = 1|x_t) = F^i(\tau^i - h^i(x_t))$ can be written in terms of the Q-function as follows

$$F^i(\tau^i - h^i(x_t)) = \frac{1}{\sqrt{2\pi r^i}} \int_{\tau^i - h^i(x_t)}^{\infty} e^{-\frac{u^2}{2r^i}} du = Q\left(\frac{\tau^i - h^i(x_t)}{\sqrt{r^i}}\right). \quad (6.7)$$

Let us now denote by $Y_t = \text{col}(y_0, \ldots, y_t)$ the vector of all binary measurements collected up to time t and by $X_t \triangleq \text{col}(x_0, \ldots, x_t)$ the vector of the state trajectory. Further, let us denote by $\hat{X}_{t|t} \triangleq \text{col}(\hat{x}_{0|t}, \ldots, \hat{x}_{t|t})$ the estimates of X_t to be made at any stage t. Then, at each time instant t, given the a posteriori probability $p(X_N|Y_N)$, the estimate of the state trajectory can be obtained by solving the following MAP estimation problem:

$$\hat{X}_{t|t} = \arg\max_{X_t} p(X_t|Y_t) = \arg\min_{X_t} -\ln p(X_t|Y_t). \quad (6.8)$$

From the Bayes rule

$$p(X_t|Y_t) \;\propto\; p(Y_t|X_t)\; p(X_t), \quad (6.9)$$

where

$$\begin{aligned} p(X_t) &= \prod_{k=0}^{t-1} p(x_{t-k}|x_{t-k-1}, \ldots, x_0)\; p(x_0) \\ &= \prod_{k=0}^{t-1} p(x_{t-k}|x_{t-k-1})\; p(x_0). \end{aligned} \quad (6.10)$$

Notice that in the latter equation we have considered the Markov property for the dynamical system state. As x_0 and w_t are normally distributed vectors, we have

$$p(x_0) \propto e^{-\frac{1}{2}\|x_0 - \overline{x}_0\|_P^2} \quad (6.11)$$

$$p(x_{k+1}|x_k) \propto e^{-\frac{1}{2}\|x_{k+1} - f(x_k, u_k)\|_Q^2}. \quad (6.12)$$

Moreover, the likelihood function $p(Y_t|X_t)$ of the binary measurement

vector Y_t can be written as

$$\begin{aligned} p(Y_t|X_t) &= \prod_{k=0}^{t} p(y_k|x_k) = \prod_{k=0}^{t} \prod_{i=1}^{l} p(y_k^i|x_k) \\ &= \prod_{k=0}^{t} \prod_{i=1}^{l} F^i(\tau^i - h^i(x_k))^{y_k^i} \; \Phi^i(\tau^i - h^i(x_k))^{1-y_k^i} \end{aligned} \tag{6.13}$$

where in the latter equality we have exploited the statistical independence of the binary sensors. Accordingly, the log-likelihood is

$$\begin{aligned} \ln p(Y_t|X_t) = \sum_{k=0}^{t} \sum_{i=1}^{l} \big[& y_k^i \ln F^i(\tau^i - h^i(x_k)) \\ & + (1 - y_k^i) \ln \Phi^i(\tau^i - h^i(x_k)) \big] , \end{aligned} \tag{6.14}$$

and the cost function $-\ln p(X_t|Y_t) = -\ln p(Y_t|X_t) - \ln p(X_t)$ to be minimized in the MAP estimation problem (6.8) turns out to be, up to additive constant terms,

$$\begin{aligned} J_t(X_t) &= \|x_0 - \overline{x}_0\|_P^2 + \sum_{k=0}^{t} \|x_{k+1} - f(x_k, u_k)\|_Q^2 \\ &- \sum_{k=0}^{t} \sum_{i=1}^{l} \big[y_k^i \ln F^i(\tau^i - h^i(x_k)) + (1 - y_k^i) \ln \Phi^i(\tau^i - h^i(x_k)) \big] \; . \end{aligned} \tag{6.15}$$

Unfortunately, a closed-form expression for the global minimum of (6.15) does not exist and, hence, the optimal MAP estimate $\hat{X}_{t|t}$ has to be determined by resorting to some numerical optimization routine. With this respect, the main drawback is that the number of optimization variables grows linearly with time, since the vector X_t has size $(t+1)\,n$. As a consequence, as t grows the solution of the full information MAP state estimation problem (6.8) becomes eventually unfeasible, and some approximation has to be introduced.

6.3 Moving-horizon approximation

In this section, an approximate solution to the MAP state estimation problem is proposed by resorting to the MHE approach [89]- [90]. Accordingly, by defining a sliding window $\mathfrak{W}_t = \{t - N, t - N + 1, \ldots, t\}$, the goal is to

find an estimate of the partial state trajectory $X_{t-N:t} \triangleq \mathrm{col}(x_{t-N}, \ldots, x_t)$ by using the information available in $\mathfrak{W}_t$. Then, in place of the full information cost $J_t(X_t)$, at each time instant t the minimization of the following *moving-horizon cost* is addressed:

$$\begin{aligned} J_t^{\mathrm{MH}}(X_{t-N:t}) = \Gamma_{t-N}(x_{t-N}) + \sum_{k=t-N}^{t} \|x_{k+1} - f(x_k, u_k)\|_Q^2 \\ - \sum_{k=t-N}^{t} \sum_{i=1}^{l} \left[y_k^i \ln F^i(\tau^i - h^i(x_k)) + (1 - y_k^i) \ln \Phi^i(\tau^i - h^i(x_k)) \right] \end{aligned} \tag{6.16}$$

where the non-negative initial penalty function $\Gamma_{t-N}(x_{t-N})$, known in the MHE literature as *arrival cost* [91, 92], is introduced so as to summarize the past data $y_0, \ldots, y_{t-N-1}$ not explicitly accounted for in the objective function.

As a matter of fact, the form of the arrival cost plays an important role in the behavior and performance of the overall estimation scheme. While in principle $\Gamma_{t-N}(x_{t-N})$ could be chosen so that minimization of (6.16) yields the same estimate that would be obtained by minimizing (6.15), an algebraic expression for such a true arrival cost seldom exists, even when the sensors provide continuous (non-binary) measurements [91]. Hence, some approximation must be used. With this respect, a common choice [92, 93], also followed in the present work, consists of assigning to the arrival cost a fixed structure penalizing the distance of the state x_{t-N} at the beginning of the sliding window from some prediction $\bar{x}_{t-N}$ computed at the previous time instant, thus making the estimation scheme recursive. A natural choice is then a quadratic arrival cost of the form

$$\Gamma_{t-N}(x_{t-N}) = \|x_{t-N} - \bar{x}_{t-N}\|_\Psi^2 , \tag{6.17}$$

which, from the Bayesian point of view, corresponds to approximating the PDF of the state x_{t-N} conditioned to all the measurements collected up to time $t-1$ with a Gaussian having mean $\bar{x}_{t-N}$ and covariance Ψ^{-1}. As for the choice of the weight matrix Ψ, in the case of continuous measurements it has been shown that stability of the estimation error dynamics can be ensured provided that Ψ is not too large (so as to avoid an overconfidence on the available estimates) [92, 93]. Recently [86], similar results have been proven to hold also in the case of binary sensors in a deterministic context. In practice, Ψ can be seen as a design parameter which has to be tuned by

pursuing a suitable tradeoff between such stability considerations and the necessity of not neglecting the already available information (since in the limit for Ψ going to zero the approach becomes a finite memory one).

Summing up, at any stage $t = N, N+1, \ldots$, the following problem has to be solved.

Problem E_t: Given the prediction $\bar{x}_{t-N}$, the input sequence $\{u_{t-N}, \ldots, u_{t-1}\}$, the measurement sequences $\{y^i_{t-N}, \ldots, y^i_t,\ i = 1, \ldots, l\}$, find the optimal estimates $\hat{x}_{t-N|t}, \ldots, \hat{x}_{t|t}$ that minimize the cost function (6.16) with arrival cost (6.17).

Concerning the propagation of the estimation procedure from Problem E_{t-1} to Problem E_t, the prediction $\bar{x}_{t-N}$ is set equal to the value of the estimate of x_{t-N} made at time instant $t-1$, i.e., $\bar{x}_{t-N} = \hat{x}_{t-N|t-1}$. Clearly, the recursion is initialized with the a priori expected value $\bar{x}_0$ of the initial state vector.

In general, solving Problem E_t entails the solution of a non-trivial optimization problem. However, when both equations (6.1) and (6.2) are linear, the resulting optimization problem turns out to be convex so that standard optimization routines can be used in order to find the global minimum. To see this, let us consider the following assumption.

A1 The functions $f(\cdot)$ and $h^i(\cdot)$, $i = 1, \ldots, l$, are linear, i.e., $f(x_t, u_t) = Ax_t + Bu_t$ and $h^i(x_t) = C^i x_t$, $i = 1, \ldots, l$, where A, B, C^i are constant matrices of suitable dimensions.

Proposition 2. *If assumption A1 holds, the CDF $\Phi^i(\tau^i - C^i x_t)$ and its complementary function $F^i(\tau^i - C^i x_t)$ are log-concave. Hence, the cost function (6.16) with arrival cost (6.17) is convex.*

□

Proof: Under assumption A1, the cost function (6.16) is convex if and only if $F^i(\tau^i - C^i x_t)$ and $\Phi^i(\tau^i - C^i x_t)$ are log-concave functions, $\forall i = 1, \ldots, p$. A function $f : \mathbb{R}^n \to \mathbb{R}$ is log-concave if $f(x) > 0$ for all x in its domain and $ln\ f(x)$ is concave [94], namely

$$\nabla^2 ln\ f(x) = \frac{1}{f^2(x)} \left[\frac{\partial^2 f(x)}{\partial x^2} f(x) - \left(\frac{\partial f(x)}{\partial x} \right)' \left(\frac{\partial f(x)}{\partial x} \right) \right] < 0. \qquad (6.18)$$

Let us now consider the CDF $\Phi^i(\tau^i - C^i x_t)$ and its complementary function $F^i(\tau^i - C^i x_t)$, that are positive functions for all $d_t^i \triangleq \tau^i - C^i x_t$, $i = 1, \ldots, l$. From the fundamental theorem of calculus, namely $\frac{\partial}{\partial x}\left(\int_{b(x)}^{a(x)} f(x)dx\right) = f(a(x))\frac{\partial a(x)}{\partial x} - f(b(x))\frac{\partial b(x)}{\partial x}$ where $a(x)$ and $b(x)$ are arbitrary functions of x, the first and the second derivatives of the function $F^i(\tau^i - C^i x_t)$ with respect to x_t are, respectively, equal to

$$\frac{\partial F^i(\tau^i - C^i x_t)}{\partial x_t} = \frac{C^i}{\sqrt{2\pi r^i}} e^{-\frac{(\tau^i - C^i x_t)^2}{2r^i}} \tag{6.19}$$

and

$$\frac{\partial^2 F^i(\tau^i - C^i x_t)}{\partial x_t^2} = \frac{(C^i)' C^i}{r^i\sqrt{2\pi r^i}}(\tau^i - C^i x_t) e^{-\frac{(\tau^i - C^i x_t)^2}{2r^i}}. \tag{6.20}$$

If $\tau^i - C^i x_t \leq 0$, then $\frac{\partial^2 F^i(\tau^i - C^i x_t)}{\partial x_t^2} \leq 0$. Hence $\frac{\partial^2 F^i}{\partial x^2} F^i \leq 0$ and, from (6.18), it follows that the Q-function F^i is log-concave. Conversely, if $\tau^i - C^i x_t > 0$, the log-concavity of F^i depends on the sign of the term

$$\frac{\partial^2 F^i}{\partial x^2} F^i - \left(\frac{\partial F^i}{\partial x}\right)' \left(\frac{\partial F^i}{\partial x}\right) =$$
$$\frac{(C^i)' C^i}{2\pi r^i} e^{-\frac{(\tau^i - C^i x_t)^2}{2r^i}} \left[\frac{\tau^i - C^i x_t}{r^i}\left(\int_{\tau^i - C^i x_t}^{\infty} e^{-\frac{u^2}{2r^i}} du\right) - e^{-\frac{(\tau^i - C^i x_t)^2}{2r^i}}\right].$$

From the convexity properties of the function $f(x) = x^2/2$, it can be easily verified for any variables s, k that $s^2/2 \geq -k^2/2 + sk$, and hence $e^{-s^2/2} \leq e^{-sk + k^2/2}$ [94]. Then, if $k > 0$, it holds that

$$\int_k^{\infty} e^{-\frac{s^2}{2}} ds \leq \int_k^{\infty} e^{-sk + \frac{k^2}{2}} ds = \frac{e^{-\frac{k^2}{2}}}{k}.$$

Since $\tau^i - C^i x_t > 0$, with a simple change of variable, it can be stated that

$$\frac{\tau^i - C^i x_t}{r^i}\left(\int_{\tau^i - C^i x_t}^{\infty} e^{-\frac{u^2}{2r^i}} du\right) \leq e^{-\frac{(\tau^i - C^i x_t)^2}{2r^i}}, \tag{6.21}$$

proving, as a consequence, the log-concavity of the Q-function $F^i(\tau^i - C^i x_t)$.

By using the complement rule, the cumulative distribution function can be written as $\Phi^i(\tau^i - C^i x_t) = 1 - F^i(\tau^i - C^i x_t) \geq 0$ and $\frac{\partial^2 \Phi^i(\tau^i - C^i x_t)}{\partial x_t^2} = -\frac{\partial^2 F^i(\tau^i - C^i x_t)}{\partial x_t^2}$. If $\tau^i - C^i x_t > 0$, then $\frac{\partial^2 \Phi^i}{\partial x^2}\Phi^i < 0$ such that Φ^i is log-concave. In the remaining case, i.e. $\tau^i - C^i x_t \leq 0$, noting that

$$\Phi^i = \frac{1}{\sqrt{2\pi r^i}} \int_{-\infty}^{\tau^i - C^i(x_t)} e^{-\frac{u^2}{2r^i}} du = \frac{1}{\sqrt{2\pi r^i}} \int_{-(\tau^i - C^i x_t)}^{\infty} e^{-\frac{u^2}{2r^i}} du$$

it can be observed that the sign of the term

$$\frac{\partial^2 \Phi^i}{\partial x^2}\Phi^i - \left(\frac{\partial \Phi^i}{\partial x}\right)'\left(\frac{\partial \Phi^i}{\partial x}\right) = \frac{(C^i)'C^i}{2\pi r^i} e^{-\frac{(\tau^i - C^i x_t)^2}{2r^i}}$$
$$\times \left[\frac{-(\tau^i - C^i x_t)}{r^i}\left(\int_{-(\tau^i - C^i x_t)}^{\infty} e^{-\frac{u^2}{2r^i}} du\right) - e^{-\frac{(\tau^i - C^i x_t)^2}{2r^i}}\right]$$

is negative, thus proving the log-concavity of the CDF $\Phi^i(\tau^i - C^i x_t)$ and the convexity of the whole cost function.

□

Remark 5. *Under assumption A1, the convexity of the cost function (6.16) is guaranteed also in the more general case in which the statistical behaviour of the random variables x_0, w_t, v_t is described by logarithmically concave distribution functions. Indeed, if a PDF is log-concave, also its cumulative distribution function is log-concave; hence the contribution related to the binary measurements in (6.16) turns out to be convex.*

In the next section we will focus on the case of a discrete-time linear system, in particular considering a diffusion process governed by a *partial differential equation* (PDE) and spatially discretized by means of the *finite element method* (FEM).

6.4 Dynamic field estimation with binary sensors

In this section, we consider the problem of reconstructing a two-dimensional diffusion field, sampled with a network of binary sensors arbitrarily deployed over the spatial domain of interest Ω. The diffusion process is governed by the following parabolic PDE:

$$\frac{\partial c}{\partial t} - \lambda \nabla^2 c \;=\; 0 \quad \text{in } \Omega \tag{6.22}$$

which models various physical phenomena such as the spread of a pollutant in a fluid. In this case, $c(\xi, \eta, t)$ represents the space-time dependent substance concentration, λ denotes the constant diffusivity of the medium, and $\nabla^2 = \partial^2/\partial\xi^2 + \partial^2/\partial\eta^2$ is the Laplace operator, $(\xi, \eta) \in \Omega$ being the 2D spatial variables. Furthermore, let us assume mixed boundary conditions (see

Section 2.6), i.e. an inhomogeneous Dirichlet condition

$$c = \psi \quad \text{on } \partial\Omega_D, \tag{6.23}$$

which specifies a constant-in-time value of concentration on the boundary $\partial\Omega_D$, and a homogeneous Neumann condition on $\partial\Omega_N = \partial\Omega \setminus \partial\Omega_D$, assumed impermeable to the contaminant, so that

$$\partial c/\partial \upsilon = 0 \quad \text{on } \partial\Omega_N, \tag{6.24}$$

where υ is the outward pointing unit normal vector of $\partial\Omega_N$.

The objective is to estimate the values of the dynamic field of interest $c(\xi, \eta, t)$, given the binary measurements (6.3). The PDE system (6.22)-(6.24) is simulated with a mesh of finite elements over Ω via the finite-element (FE) approximation described in Chapter 3. Specifically, the domain Ω is subdivided into a suitable set of non overlapping regions, or elements, and a suitable set of basis functions $\phi_j(\xi, \eta)$, $j = 1, \ldots, n_\phi$, is defined on such elements. In the specific case under investigation, the elements are triangles in 2D and define a FE mesh with vertices $(\xi_j, \eta_j) \in \Omega, j = 1, \ldots, n_\phi$. In order to account for the mixed boundary conditions, the basis functions are supposed to be ordered so that the first n correspond to vertices of the mesh which lie either in the interior of Ω or on $\partial\Omega_N$, while the last $n_\phi - n$ correspond to the vertices lying on $\partial\Omega_D$.

Accordingly, the unknown function $c(\xi, \eta, t)$ is approximated as

$$c(\xi, \eta, t) \approx \sum_{j=1}^{n} \phi_j(\xi, \eta)\, c_j(t) + \sum_{j=n+1}^{n_\phi} \phi_j(\xi, \eta)\, \psi_j \tag{6.25}$$

where $c_j(t)$ is the unknown expansion coefficient of the function $c(\xi, \eta, t)$ relative to time t and basis function $\phi_j(\xi, \eta)$, and ψ_j is the known expansion coefficient of the function $\psi(\xi, \eta)$ relative to the basis function $\phi_j(\xi, \eta)$. Notice that the second summation in (6.25) is needed so as to impose the inhomogeneous Dirichlet condition (6.23) on the boundary $\partial\Omega_D$.

The PDE (6.22) can be recast into the following integral form:

$$\int_\Omega \frac{\partial c}{\partial t} \varphi \, d\xi d\eta \; - \; \lambda \int_\Omega \nabla^2 c \; \varphi \, d\xi d\eta = 0 \tag{6.26}$$

where $\varphi(\xi, \eta)$ is a generic space-dependent weight function. By applying Green's identity, one obtains:

$$\int_\Omega \frac{\partial c}{\partial t} \varphi \, d\xi d\eta + \lambda \int_\Omega \nabla^T c \; \nabla\varphi \, d\xi d\eta - \lambda \int_{\partial\Omega} \frac{\partial c}{\partial \upsilon} \varphi \, d\xi d\eta = 0 \, . \tag{6.27}$$

By choosing the test function φ equal to the selected basis functions and exploiting the approximation (6.25), the Galerkin weighted residual method is applied and the following equation is obtained

$$\sum_{i=1}^{n} \int_{\Omega} \phi_i \phi_j \, d\xi d\eta \, \dot{c}_i(t) + \lambda \sum_{i=1}^{n} \int_{\Omega} \nabla^T \phi_i \, \nabla \phi_j \, d\xi d\eta \, c_i(t) + \lambda \sum_{i=n+1}^{n_\phi} \int_{\Omega} \nabla^T \phi_i \, \nabla \phi_j \, d\xi d\eta \, \psi_i = 0 \tag{6.28}$$

for $j = 1, \ldots, n$. Notice that in the latter equation the boundary integral of equation (6.27) is omitted since it is equal to 0 thanks to the homogeneous Neumann condition (6.24) on $\partial\Omega_N$ and to the fact that, by construction, the basis functions ϕ_j, $j = 1, \ldots, n$, vanish on $\partial\Omega_D$.

By defining the state vector $x = \mathrm{col}(c_1, \ldots, c_n)$ and the vector of boundary conditions $\gamma = \mathrm{col}(\psi_{n+1}, \ldots, \psi_{n_\phi})$, equation (6.28) can be written in the more compact form as

$$M\dot{x}(t) + Sx(t) + S_D\gamma = 0$$

where S is the so-called stiffness matrix representing diffusion, M is the mass matrix, and S_D captures the physical interconnections among the vertices affected by boundary condition (6.23) and the remaining nodes of the mesh.

By applying for example the implicit Euler method, the latter equation can be discretized in time, thus obtaining the linear discrete-time model

$$x_{t+1} = A\, x_t + B\, u + w_t \tag{6.29}$$

where

$$A = \left[I + \delta t \; M^{-1} S\right]^{-1}$$
$$B = \left[I + \delta t \; M^{-1} S\right]^{-1} M^{-1} \delta t$$
$$u = -S_D \; \gamma$$

δt is the time integration interval, and w_t is the process disturbance taking into account also the space-time discretization errors.

Notice that the linear system (6.29) has dimension n equal to the number of vertices of the mesh not lying on $\partial\Omega_D$. The linear system (6.29) is assumed to be monitored by a network of l threshold sensors. Each sensor, before binary quantization is applied, directly measures the pointwise-in-time-and-space concentration of the contaminant in a point of the spatial domain

Ω. By exploiting (6.25), such a concentration can be written as a linear combination of the concentrations on the grid points, so that the resulting output function takes the form

$$z_t^i = C^i x_t + v_t^i, \quad i = 1, \dots, l \tag{6.30}$$

and assumption A1 is fulfilled.

6.5 Numerical example

In this section, we present the simulation results of the proposed approach applied to the problem of state estimation of spatially distributed processes, discussed in the previous section. We consider the simulated system (6.29)-(6.30) with 1695 triangular elements, 915 vertices, $\lambda = 0.01$ $[m^2/s]$, fixed integration step length $\delta t = 1$ $[s]$, $\gamma = 30$ $[g/m^2]$, and initial condition of the field vector $x_0 = 0_n$ $[g/m^2]$. The field of interest is defined over a bounded 2D spatial domain Ω which covers an area of 7.44 $[m^2]$ (see Fig. 6.1), with boundary condition (6.23) on the bottom edge and no-flux condition (6.24) on the remaining portions of $\partial\Omega$. Compared to the *ground truth*

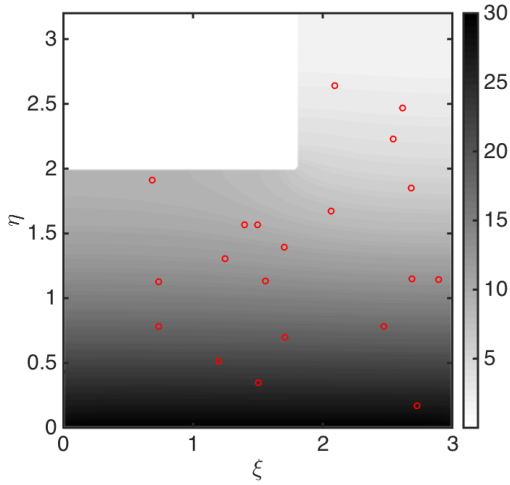


Figure 6.1: Concentration field at time $t = 100$ $[s]$ monitored by a random network of 20 binary sensors (red $\circ$).

simulator, the proposed MH-MAP estimator implements a coarser mesh (see

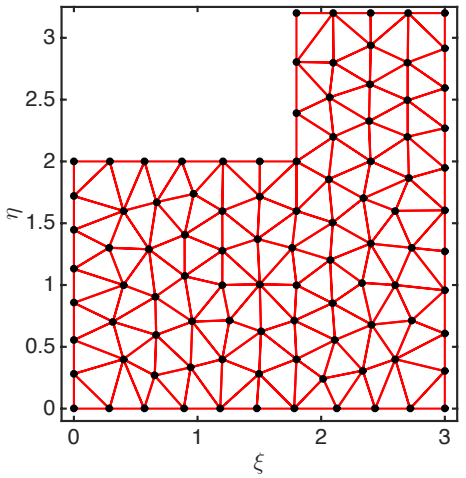


Figure 6.2: Mesh used by the MH-MAP estimator (152 elements, 97 nodes).

Fig. 6.2) of $n_\phi = 97$ vertices ($n = 89$), and runs at a slower sample rate (0.1 $[Hz]$), so that model uncertainty is taken into account. The initial condition of the estimated dynamic field is set to $\overline{x}_0 = 5 \cdot 1_n$ $[g/m^2]$, the moving window has size $N = 5$, and the weight matrices in (6.4) are chosen as $\Psi = 10^3\ I_n$ and $Q = 10^2\ I_n$. The *true* concentrations from (6.29) are first corrupted with a Gaussian noise with variance r^i, then binary observations are obtained by applying a different threshold τ^i for each sensor i of the network. Note that, in order to receive informative binary measurements, $\tau^i,\ i = 1, ..., l$, are generated as uniformly distributed random numbers in the interval $(0.05, 29.95)$, being $(0, 30)$ the range of nominal concentration values throughout each experiment. The duration of each simulation experiment is fixed to 1200 $[s]$ (120 samples).

Fig. 6.3 shows the performance of the novel MH-MAP state estimator implemented in Matlab, in terms of Root Mean Square Error (RMSE) of the estimated concentration field, i.e.:

$$\mathrm{RMSE}(t) = \left(\sum_{j=1}^{\alpha} \frac{\|e_{t,j}\|^2}{\alpha} \right)^{\frac{1}{2}}, \tag{6.31}$$

where $\|e_{t,j}\|$ is the norm of the estimation error at time t in the $j-$th simulation run, averaged over 304 sampling points (evenly spread within Ω) and

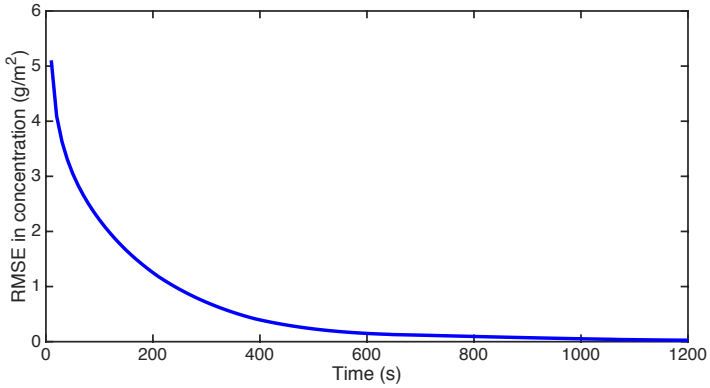


Figure 6.3: RMSE in concentration of the MH-MAP state estimator as a function of time, for a random network of 5 threshold sensors.

$\alpha = 100$ independent Monte Carlo realizations. The estimation error is computed at time t on the basis of the estimate $\hat{x}_{t-N|t}$. It can be observed

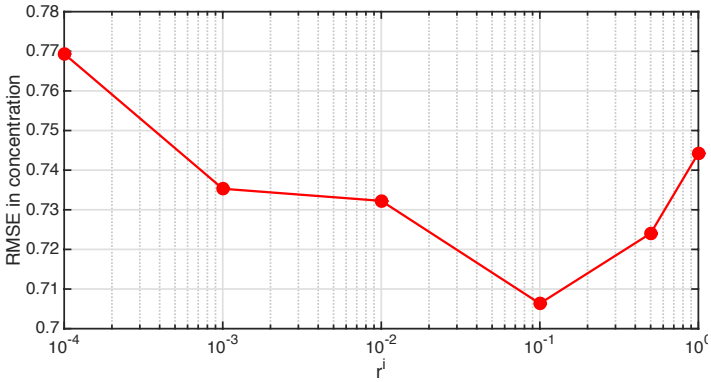


Figure 6.4: RMSE in concentration as a function of the measurement noise variance, for a fixed constellation of 20 binary sensors. It is shown here that operating in a noisy environment turns out to be beneficial, for certain values of r^i, to the state estimation problem.

that the proposed estimator successfully estimates the dynamic field, even when observed by a network of $l = 5$ randomly deployed binary sensors, with $r^i = 0.25\ [g/m^2]$ for $i = 1, ..., l$. The effect of measurement noise on the mean

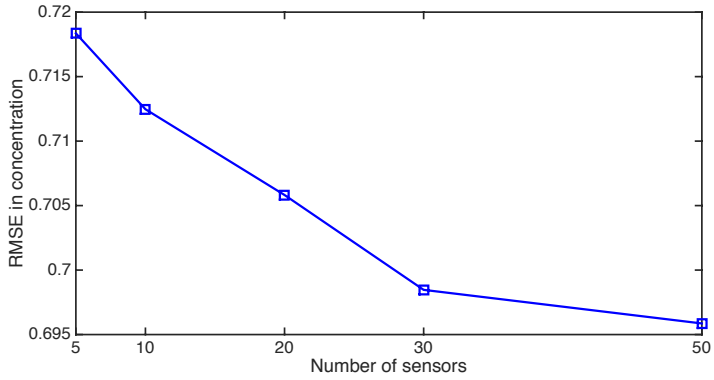


Figure 6.5: RMSE of the concentration estimates as a function of the number of sensors deployed over the monitoring area.

value of the RMSE can be seen in Fig. 6.4, in which it becomes apparent how for certain values of r^i, including an observation noise with higher variance, can actually improve the quality of the overall estimates. The results in Fig. 6.4 numerically demonstrate the validity of the above stated noise-aided paradigm in the recursive state estimation with binary measurements and, thus, represent an interesting contribution of this work. Finally, Fig. 6.5 shows the evolution of the RMSE as a function of the number of binary observations available at the fusion center.

6.6 Conclusions

State estimation with binary sensors has been formulated as a *Moving Horizon* (MH) *Maximum A posteriori Probability* (MAP) optimization problem and it has been shown how such a problem turns out to be convex in the linear system case. Simulation results relative to a dynamic field estimation case-study have exhibited the conjectured noise-aided feature of the proposed estimator in that the estimation accuracy improves, starting from a null measurement noise, until the variance of the latter achieves an optimal value beyond which estimation performance decays.

Chapter 7

Dynamic field estimation in adversarial environments

7.1 Introduction

The aim of this chapter is to address new challenges introduced in the environment under consideration by the progress of the so-called *Cyber-Physical Systems* (CPS). In particular, next-generation monitoring/control systems of spatially distributed processes are typical examples of CPS, i.e. complex systems integrating computation, networking capabilities and physical processes. Due to their strategic importance in homeland security, situation awareness, environmental and industrial monitoring, etc. such systems are nowadays subject to the potential threat of *cyber-physical* attacks. Indeed, while advances in CPS technology will provide enhanced autonomy, efficiency, seamless interoperability and cooperation, the tighter interaction between cyber and physical realms is unavoidably introducing novel security vulnerabilities, which make CPS subject to non-standard malicious threats.

Recent real-world attacks, such as the Maroochy Shire sewage spill where a hacker caused the release of $800,000$ liters of untreated sewage into waterways, the Stuxnet worm which targeted nuclear industry software and equipment in Iran, and the lately reported massive power outage against Ukrainian electric grid, have brought the attention of the engineering community towards the urgency of designing secure CPS. There exist a wide range of cyber-physical attacks and a variety of approaches to handle them, i.e. to detect the attack outbreak as well as to correctly estimate the system

state even in presence of the attack.

Preliminary studies addressed the problems of attack detection/identification, and proposed attack monitors for deterministic control systems [74]. In addition, active detection methods have been designed in order to reveal stealthy attacks via manipulation of e.g. control inputs [95] and dynamics [96]. In recent times, the problem of secure state estimation, i.e. capable of reconstructing the state even when the CPS of interest is under attack, has gained considerable attention [97], [98]. Under the assumption of linear systems subject to an unknown, but bounded, number of *false-data injection* attacks, the problem for a noise-free system has been cast into an ℓ_0-optimization problem, which can be relaxed as a more efficient convex problem [99], and later adapted to systems with bounded noise [100]. Further advances try to tackle the combinatorial complexity of the problem [101]. Lately, the most popular types of attack have been modeled based on adversary's resources and system knowledge [102], and resilient state estimation has been also addressed for noisy systems under both data injection and switching attacks [103].

This chapter specifically focuses on dynamic field estimation in *adversarial* environments. A typical example of such an *adversarial* setting is the problem of detecting and localizing an unknown malicious source (e.g. a biochemical attack) inducing and/or altering the field to be monitored (the similar problem in a non-malicious setup has been presented in Chapter 5). In this specific case, the objective of secure estimation becomes the joint task of detecting the presence of the source, localizing it, estimating its intensity, and monitoring the induced field. A possible solution to the above problem can be found by exploiting the approach presented in Section 5.5, i.e. by modeling the intensity of the malicious source as an unknown input (model (a) in Section 5.3). In the sequel it will be shown how the above source attack can be modeled as a more general *switching mode attack* by which the attacker can switch the currently operating mode of the CPS within a finite set of possible attack modes. This can be achieved, for instance, by altering the network's topology in a power system through breaker control signals [104]; the same type of attack has been also studied on water distribution systems [105], where the water outflow can be influenced via boundary control actions.

In particular, this chapter aims to address the general problem of jointly detecting a signal attack (e.g., the intensity of a malicious source) and esti-

mating both the attack mode (e.g., the position of the source) and system state (e.g., the source-induced field) from the available observations. The overall problem is formulated in a stochastic random set Bayesian framework by exploiting Bernoulli modeling for the signal attack presence/absence and multiple models to account for the different attack modes. It is worth to highlight that the adopted approach exhibits the following positive features: 1) can deal with nonlinear systems; 2) takes into account the presence of disturbances and noise; 3) can encompass in a unique framework different types of attacks (switching signal and mode attacks, extra packet injection, packet substitution, etc.); 4) provides (discrete or continuous) probability distributions of the attack existence, attack mode, attack signal and system state which are very useful for taking decisions. Preliminary results on the single-model case of this topic are presented in [106].

The rest of the chapter is organized as follows. Section 7.2 introduces the considered attack models and provides the necessary background. Section 7.3 formulates and solves the joint *attack detection and mode-state estimation* problem of interest in the Bayesian framework. Section 7.4 discusses the Gaussian-mixture implementation of the joint attack detector and mode-state estimator derived in Section 7.3. Then, Section 7.5 demonstrates the effectiveness of the proposed approach via a simulation example concerning a power network. Finally, Section 7.6 ends the chapter with concluding remarks.

7.2 Problem formulation and preliminaries

7.2.1 System and attack model

Let the discrete-time cyber-physical system in *adversarial* environment be modeled by

$$x_{k+1} = \begin{cases} f_k^0(\nu_k, x_k) + w_k, & \text{under no signal attack} \\ f_k^1(\nu_k, x_k, a_k) + w_k, & \text{under signal attack} \end{cases} \tag{7.1}$$

where: k is the time index; $\nu_k \in \mathfrak{M} = \{1, 2, ..., \mathfrak{m}\}$ is the mode variable in operation at time k; $x_k \in \mathbb{R}^n$ is the state vector to be estimated; $a_k \in \mathbb{R}^m$, called attack vector, is an unknown input affecting the system only when it is under attack; $f_k^0(\nu_k, \cdot)$ and $f_k^1(\nu_k, \cdot, \cdot)$ are known mode-matched state transition functions that describe the system evolution under a specific

mode ν_k, in the *no signal attack* and, respectively, *signal attack* cases; w_k is a random process disturbance, with probability density function (PDF) $p_w(\nu_k, \cdot)$, also affecting the system. For monitoring purposes, the state of the above system is observed through the measurement model

$$y_k = \begin{cases} h_k^0(\nu_k, x_k) + v_k, & \text{under no signal attack} \\ h_k^1(\nu_k, x_k, a_k) + v_k, & \text{under signal attack} \end{cases} \tag{7.2}$$

where: $h_k^0(\nu_k, \cdot)$ and $h_k^1(\nu_k, \cdot, \cdot)$ are known mode-matched measurement functions that refer to the *no signal attack* and, respectively, *signal attack* cases; v_k is a random measurement noise with PDF $p_v(\nu_k, \cdot)$. It is assumed that the measurement y_k is actually delivered to the system monitor with probability $p_d \in (0, 1]$, where the non-unit probability might be due to a number of reasons (e.g. temporary denial of service, packet loss, sensor inability to detect or sense the system, etc.).

Jump Markov models (7.1)-(7.2) allow to describe cyber-physical systems subject to two different types of switching attacks, as considered in [103]: (i) switching mode attacks, and (ii) switching signal attacks. The former class of attacks is capable of switching the ongoing mode of the system between a finite set of possible models $\mathfrak{M}$, by e.g. altering the state transition of the system (in [104] the topology of a power network). Moreover, a change in the system mode might represent a modification of the set of corruptible actuators/sensors, i.e. a change of the structure under which the signal attack enters the system. In other words, switching mode attacks model every possible cyber-physical adversary's action causing a change of the functions f^0, f^1 governing the system dynamics and/or of the functions h^0, h^1 describing the observation process. On the other hand, a signal attack (ii), modeled in (7.1)-(7.2) via the attack vector a_k, is a time-varying signal of arbitrary magnitude and location injected into the system to corrupt sensor/actuator data (also known as *false-data injection attack*), here modeled as an unknown input. Specifically, as in [110] for unknown inputs, a_k is treated as a white stochastic process $\{a_k\}$, independent of $x_0, \{w_k\}$ and $\{v_k\}$. This means that a_k and a_l are independent random variables for $k \neq l$, and a_k is independent of x_k and y^{k-1}. Such an assumption accounts for the fact that a_k may assume all possible values, being completely unknown (we consider the most general model for signal attacks where any value can be injected via the compromised actuators/sensors), and the knowledge of a_k adds no information on a_l, if $k \neq l$. At each time instant k, the signal attack can

be present or not, according to the binary hypothesis 1 or 0, respectively, in (7.1)-(7.2).

Besides the above switching attacks (i) and (ii), the proposed attack model takes into account the presence of malicious *extra packet injections*, already addressed in [107], [108], and [106]. This means that, in addition to the system-originated measurement y_k in (7.2), it is assumed that the system monitor might receive from some cyber-attacker one or multiple extra fake measurements indistinguishable (e.g. with same time stamp and sender id) from the system-originated one. For the subsequent developments, it is convenient to introduce the *attack set* at time k, $\mathcal{A}_k$, which is either equal to the empty set if the system is not under signal attack at time k or to the singleton $\{a_k\}$ otherwise, i.e.

$$\mathcal{A}_k \;=\; \begin{cases} \emptyset, & \text{if the system is not under signal attack} \\ \{a_k\}, & \text{otherwise.} \end{cases}$$

Due to the possible presence of the *extra packet injection* attack, it is also convenient to define the *measurement set* at time k

$$\mathcal{Z}_k \;=\; \mathcal{Y}_k \cup \mathcal{F}_k \tag{7.3}$$

where

$$\mathcal{Y}_k \;=\; \begin{cases} \emptyset & \text{with probability } 1 - p_d \\ \{y_k\} & \text{with probability } p_d \end{cases} \tag{7.4}$$

is the set of system-originated measurements and $\mathcal{F}_k$ the finite set of fake measurements. It is worth mentioning that the above attack model could be extended to include the case of *packet substitution*, see [106].

7.2.2 Multiple model approach

In order to handle switching attacks that can change the model in effect of the cyber-physical system (7.1)-(7.2), single-model approaches to secure state estimation, like the one proposed in [106], need to be accomodated for switching systems. To this end, the idea is to rely on the Multiple Model (MM) approach [76]. For state estimation problems in jump Markov systems with known inputs, the MM framework provides, in theory, Bayes-optimal solutions by running in parallel a bank of $\mathtt{m}$ mode-matched Bayesian filters. In simple terms, each filter, at each time instant, provides the mode-conditioned PDFs of the state given the observations, and recursively computes the modal

probabilities for each mode ν_k. In this way, the MM approach can infer the *best* model of the current system's mode of operation as well as estimate the state of the system, based on the mode estimate. The MM approach commonly assumes that the true mode of the system switches according to a (homogeneous) Markov chain with known transition probabilities

$$\pi_{ji} = \text{prob}\left(\nu_k = i \,|\, \nu_{k-1} = j\right), \quad i, j \in \mathfrak{M} \tag{7.5}$$

where $\sum_{i=1}^{\mathfrak{m}} \pi_{ji} = 1$. This assumption leads to the so called *Dynamic MM estimator*. A particular case of the aforementioned filter is the *Static MM estimator* which conversely assumes a constant mode variable $\nu_k \in \mathfrak{M}$, i.e. $\nu_k = \nu$, and hence $\pi_{ji} = 0 \; \forall j, i \in \mathfrak{M}$ with $j \neq i$. In the special case of joint mode, state, and attack estimation in cyber-physical systems of the form (7.1)-(7.2), the simultaneous presence of both the unknown mode and input affecting the system poses new challenges. In particular, the signal attack vector a_k can be considered as either a mode-dependent vector with possibly different dimension within distinct system modes, or a vector with fixed dimension for each possible mode (as assumed in [103]). Furthermore, two possible approaches can be undertaken for solving the above problem, depending on the available knowledge of mode transitions. In this respect, although in adversarial environments it is usually more realistic to assume no prior knowledge of the mode transition model, and hence the *Static MM* approach provides the most suitable tool, there also exist cases where a *Dynamic MM* approach turns out to be preferable. A typical example of such a case is the problem of detecting and localizing an unknown malicious source which will be described next.

7.2.3 Malicious source estimation

Since the unknown location of the source corresponds to a specific mode of the system, and the source intensity can be treated as an unknown signal attack a_k, the problem of malicious source estimation can be recast as a joint mode, state, and attack estimation in jump Markov systems (7.1)-(7.2). Notice that in this case, the attack vector a_k (intensity) has fixed dimension for each mode, and prior knowledge of the modal transitions can be assumed (since at each time instant they depend on the current location of the moving source, and hence modes corresponding to locations close to this position will clearly have higher probability to become the active mode of the system at the next step). This is the reason why in these types of

problems a *Dynamic MM* approach is preferable to undertake with respect to a static filtering.

Let us consider an *advection-diffusion* process described by a PDE of the form (5.1), i.e.

$$\frac{\partial x}{\partial t} - \lambda \nabla^2 x + v^T \nabla x \;=\; f^a \quad \text{in } \Omega \tag{7.6}$$

Here $f^a(p,t)$ represents the forcing term modeling a malicious point source injected by an attacker within the monitored domain Ω as

$$f^a(p,t) \;=\; \begin{cases} 0, & \text{under no source attack} \\ a(t)\;\delta\left(p - p^a(t)\right), & \text{under source attack} \end{cases} \tag{7.7}$$

The considered diffusive source is characterized by unknown *intensity* $a(t)$ and *position* $p^a(t) \in \Omega$. The aim is to detect the presence of the malicious source and jointly estimate $a(t), p^a(t), x(p,t)$ given measurements

$$y_{k,i} \;=\; h_i\left(x\left(s_i, t_k\right)\right) + v_{k,i} \tag{7.8}$$

The aforementioned problem of infinite dimension can be approximated, as presented in Section 5.3, via the *finite element* method so as to obtain a discrete-time model for the system under source attack of the form (6.29)

$$x_{k+1} = Ax_k + B(p^a)a_k + b_k + w_k \tag{7.9}$$

Thus, the discrete-time CPS model (7.1) for dynamic field estimation under source attack can be rewritten as

$$x_{k+1} = \begin{cases} Ax_k + b_k + w_k, & \text{under no source attack} \\ Ax_k + B(p^a)a_k + b_k + w_k, & \text{under source attack} \end{cases} \tag{7.10}$$

so that the same attack model discussed in Section 7.2.1 can be adopted. To sum up, following the key idea of the source estimation approach in Section 5.5, it is possible to derive, under the assumption of a signal attack vector $a_k \in \mathbb{R}^m$ with fixed dimension, a Bayesian recursion based on the *Dynamic MM* filtering.

Next, the general problem of joint signal attack detection and simultaneous mode–state estimation will be addressed. This amounts to jointly estimating, at each time k, the mode ν_k modeling switching mode attacks, the state x_k, and the signal attack set $\mathcal{A}_k$ given the set of measurements $\mathcal{Z}^k \stackrel{\triangle}{=} \cup_{i=1}^{k} \mathcal{Z}_i$ up to time k. Note that, differently from the general framework where the two types of switching attack will be treated separately, the

model of source attack considered in this section usually implies a combined action of *signal* and *mode* attacks, i.e. the position of the switching malicious source is not known *a priori* and hence its presence will directly correspond to a source-induced field behavior that can follow multiple models.

7.2.4 Joint input and state estimation

In this section we review the formulation of the *Joint Input and State Estimation* (JISE) problem [109], [110] in the Bayesian framework. To this end, let us consider a system with direct feedthrough of the form

$$\begin{cases} x_{k+1} &= f(x_k, u_k) + w_k \\ y_k &= h(x_k, u_k) + v_k \end{cases} \tag{7.11}$$

where u_k is the unknown input vector. The goal of stochastic Bayesian filtering is to recursively estimate the time-varying posterior PDF of the unknown variables conditioned on all the information available up to that time. Hence, when the objective is the simultaneous input and state estimation, at each time instant k, the estimates of u_k and x_k can be obtained by solving the following problem.

JISE problem: For the system (7.11), given the measurement set $y^k = \{y_1, y_2, \ldots, y_k\}$, sequentially compute the joint conditional PDF $p(u_k, x_k|y^k)$ from $p(u_{k-1}, x_{k-1}|y^{k-1})$.

Assuming that the initial density $p(u_0, x_0)$ is given, the solution can be described as a two-step procedure of prediction and correction. Let $p(u_{k-1}, x_{k-1}|y^{k-1})$ denote the posterior PDF at $k-1$. The prediction step computes the conditional PDF $p(x_k|y^{k-1})$ via the Chapman-Kolmogorov equation:

$$p(x_k|y^{k-1}) = \iint p(x_k|u_{k-1}, x_{k-1}) p(u_{k-1}, x_{k-1}|y^{k-1})\, du_{k-1} dx_{k-1} \tag{7.12}$$

Then, at time instant k, the observed output y_k is available and can be used to update $p(x_k|y^{k-1})$ and jointly estimate the conditional PDF of u_k, y_k being the first measurement containing information about the unknown signal. The correction step is then performed by applying the Bayes rule:

$$p(u_k, x_k|y^k) = \frac{p(y_k|u_k, x_k)\, p(x_k|y^{k-1})\, p(u_k)}{p(y_k|y^{k-1})} \tag{7.13}$$

Note that in (7.13) the unknown input is treated as a white stochastic process $\{u_k\}$, independent of $x_0, \{w_k\}$ and $\{v_k\}$. This means that u_k and u_l are independent random variables for $k \neq l$, and u_k is independent of x_k and y^{k-1}. With the derived Bayesian solution to JISE in the presence of direct feedthrough, optimal (with respect to any criterion) point estimates of the input and state can be obtained from this PDF, e.g. the Maximum A-posteriori Probability (MAP) estimate.

7.2.5 Random set estimation

An RFS (*Random Finite Set*) $\mathcal{X}$ over $\mathbb{X}$ is a random variable taking values in $\mathcal{F}(\mathbb{X})$, the collection of all finite subsets of $\mathbb{X}$. The mathematical background needed for Bayesian random set estimation can be found in [111]; here, the basic concepts needed for the subsequent developments are briefly reviewed. From a probabilistic viewpoint, an RFS $\mathcal{X}$ is completely characterized by its *set density* $f(\mathcal{X})$, also called FISST (*FInite Set STatistics*) probability density. In fact, given $f(\mathcal{X})$, the cardinality *probability mass function* $p(n)$ that $\mathcal{X}$ have $n \geq 0$ elements and the joint PDFs $f(x_1, x_2, \ldots, x_n|n)$ over $\mathbb{X}^n$ given that $\mathcal{X}$ have n elements, are obtained as follows:

$$\begin{aligned} p(n) &= \int_{\mathbb{X}^n} f(\{x_1, \ldots, x_n\})\, dx_1 \cdots dx_n \\ f(x_1, x_2, \ldots, x_n|n) &= \frac{1}{n! p(n)}\, f(\{x_1, \ldots, x_n\}) \end{aligned}$$

In order to measure probability over subsets of $\mathbb{X}$ or compute expectations of random set variables, [111] introduced the notion of *set integral* for a generic real-valued function $g(\mathcal{X})$ of an RFS $\mathcal{X}$ as

$$\int g(\mathcal{X})\, \delta\mathcal{X} = g(\emptyset) + \sum_{n=1}^{\infty} \frac{1}{n!} \int g(\{x_1, \ldots, x_n\})\, dx_1 \cdots dx_n \tag{7.14}$$

Two specific types of RFSs, i.e. Bernoulli and Poisson RFSs, will be considered in this work.

Bernoulli RFS

A Bernoulli RFS is a random set which can be either empty or, with some probability $r \in [0, 1]$, a singleton $\{x\}$ distributed over $\mathbb{X}$ according to the

PDF $p(x)$. Accordingly, its set density is defined as follows:

$$f(\mathcal{X}) = \begin{cases} 1-r, & \text{if } \mathcal{X} = \emptyset \\ r \cdot p(x), & \text{if } \mathcal{X} = \{x\} \end{cases} \tag{7.15}$$

Poisson RFS

A Poisson RFS is a random finite set with Poisson-distributed cardinality, i.e.

$$p(n) = \frac{e^{-\xi}\xi^n}{n!}, \quad n = 0, 1, 2, \ldots \tag{7.16}$$

and elements independently distributed over $\mathbb{X}$ according to a given spatial density $p(\cdot)$. Accordingly, its set density is defined as follows:

$$f(\mathcal{X}) = e^{-\xi} \prod_{x \in \mathcal{X}} \xi\, p(x). \tag{7.17}$$

7.3 Bayesian random set filter for secure estimation

Let the signal attack input at time k be modeled as a Bernoulli random set $\mathcal{A}_k \in \mathcal{B}(\mathbb{A})$, where $\mathcal{B}(\mathbb{A}) = \emptyset \cup \mathcal{S}(\mathbb{A})$ is a set of all finite subsets of the attack probability space $\mathbb{A} \subseteq \mathbb{R}^m$, and $\mathcal{S}$ denotes the set of all singletons (i.e., sets with cardinality 1) $\{a\}$ such that $a \in \mathbb{A}$. Further, let $\mathbb{X} \subseteq \mathbb{R}^n$ denote the Euclidean space for the system state vector. Then, a new state variable $(\mathcal{A}, x)$, referred to as *Hybrid Bernoulli Random Set* (HBRS), which incorporates the Bernoulli attack random set $\mathcal{A}$ and the random state vector x can be defined (see [106]). The HBRS can be subsequently augmented in order to include the hidden mode (or discrete state) in the new state variable $(\mathcal{A}, x, \nu)$, that we refer to as *Multiple Model Hybrid Bernoulli Random Set* (MM-HBRS), which takes values in the hybrid space $\mathcal{B}(\mathbb{A}) \times \mathbb{X} \times \mathfrak{M}$. An MM-HBRS is fully specified by the (signal attack) probability r of $\mathcal{A}$ being a singleton, the mode-conditioned PDF $p^0(x, \nu)$, and the mode-conditioned joint PDF $p^1(a, x, \nu)$, i.e.

$$p(\mathcal{A}, x, \nu) = \begin{cases} (1-r)\, p^0(x, \nu), & \text{if } \mathcal{A} = \emptyset \\ r \cdot p^1(a, x, \nu), & \text{if } \mathcal{A} = \{a\} \end{cases} \tag{7.18}$$

with integration over the new state space

$$\sum_{i=1}^{\mathfrak{m}} \mu^i \int_{\mathcal{F}(\mathcal{B})\times\mathbb{X}} p(\mathcal{A}, x|\nu_k = i)\, \delta\mathcal{A}\, dx \tag{7.19}$$

where

$$\int_{\mathcal{F}(\mathcal{B})\times\mathbb{X}} p(\mathcal{A}, x|\nu_k = i)\, \delta\mathcal{A}\, dx = \tag{7.20}$$
$$\int p(\emptyset, x, \nu_k = i)\, dx + \iint p(\{a\}, x, \nu_k = i)\, da\, dx$$

and $\mu^i \triangleq \text{prob}(\nu_k = i|\mathcal{Z})$ is the mode probability of mode i, given the measurement set $\mathcal{Z}$. The set integration with respect to $\mathcal{A}$ is defined according to (7.14) while the integration with respect to x is an ordinary one. Notice that in (7.19) $p(\mathcal{A}, x, \nu)$ integrates to one, since integration with respect to $\mathcal{A}$ and x equals 1, $p^0(x)$ and $p^1(a, x)$ being conventional probability density functions, and $\sum_{i=1}^{\mathfrak{m}} \mu^i = 1$. Thus, (7.18) turns out to be a FISST probability density for the MM-HBRS $(\mathcal{A}, x, \nu)$, which will be referred to as *multiple model hybrid Bernoulli density* throughout the rest of the chapter.

An MM-HBRS can be corrected and predicted in a recursive fashion so as to form a novel *Multiple Model Hybrid Bernoulli Filter* (MM-HBF).

7.3.1 Measurement model and correction

Due to the possible presence of *extra packet injection*, whose attack model has been introduced in Section 7.2.1, the measurement set defined in (7.3) is given by the union of two independent random sets. As it is clear from (7.4), $\mathcal{Y}_k$ is a Bernoulli random set (with cardinality $|\mathcal{Y}_k|$ at most 1) which depends on whether the system-originated measurement y_k is delivered or not. Conversely, $\mathcal{F}_k$ is the random set of fake measurements that will be modeled hereafter as a Poisson random set, such that the number of counterfeit measurements is Poisson-distributed according to (7.16) and the FISST PDF of fake-only measurements $\gamma(\mathcal{F}_k)$ is given by (7.17) with spatial distribution $\kappa(\cdot)$ in place of $p(\cdot)$. For the measurement set (7.3), the aim is to find the expression of the likelihood function $\lambda(\mathcal{Z}_k|\mathcal{A}_k, x_k, \nu_k)$. To this end, let us first introduce the following FISST PDF for $\mathcal{A}_k = \emptyset$:

$$\eta(\mathcal{Y}_k|\emptyset, x_k, \nu_k) = \begin{cases} 1 - p_d, & \text{if } \mathcal{Y}_k = \emptyset \\ p_d\ \ell(y_k|x_k, \nu_k), & \text{if } \mathcal{Y}_k = \{y_k\} \end{cases} \tag{7.21}$$

and for $\mathcal{A}_k = \{a_k\}$:

$$\eta(\mathcal{Y}_k|\{a_k\}, x_k, \nu_k) = \begin{cases} 1 - p_d, & \text{if } \mathcal{Y}_k = \emptyset \\ p_d\, \ell(y_k|a_k, x_k, \nu_k), & \text{if } \mathcal{Y}_k = \{y_k\} \end{cases}. \tag{7.22}$$

Then, using the convolution formula [111, p. 385], it follows that

$$\lambda(\mathcal{Z}_k|\mathcal{A}_k, x_k, \nu_k) = \sum_{\mathcal{Y}_k \subseteq \mathcal{Z}_k} \eta(\mathcal{Y}_k|\mathcal{A}_k, x_k, \nu_k)\, \gamma(\mathcal{Z}_k \setminus \mathcal{Y}_k). \tag{7.23}$$

Hence, the likelihood corresponding to $\mathcal{A}_k = \emptyset$ is given by

$$\begin{aligned} \lambda(\mathcal{Z}_k|\emptyset, x_k, \nu_k) &= \eta(\emptyset|\emptyset, x_k, \nu_k)\, \gamma(\mathcal{F}_k) \\ &\quad + \sum_{y_k \in \mathcal{Z}_k} \eta(\{y_k\}|\emptyset, x_k, \nu_k)\, \gamma(\mathcal{Z}_k \setminus \{y_k\}) \\ &= \gamma(\mathcal{F}_k) \left[1 - p_d + p_d \sum_{y_k \in \mathcal{Z}_k} \frac{\ell(y_k|x_k, \nu_k)}{\xi\, \kappa(y_k)}\right] \end{aligned} \tag{7.24}$$

where (7.21) and (7.17) have been used, while for $\mathcal{A}_k = \{a_k\}$ we have

$$\begin{aligned} \lambda(\mathcal{Z}_k|\{a_k\}, x_k, \nu_k) &= \eta(\emptyset|\{a_k\}, x_k, \nu_k)\, \gamma(\mathcal{F}_k) \\ &\quad + \sum_{y_k \in \mathcal{Z}_k} \eta(\{y_k\}|\{a_k\}, x_k, \nu_k)\, \gamma(\mathcal{Z}_k \setminus \{y_k\}) \\ &= \gamma(\mathcal{F}_k) \left[1 - p_d + p_d \sum_{y_k \in \mathcal{Z}_k} \frac{\ell(y_k|a_k, x_k, \nu_k)}{\xi\, \kappa(y_k)}\right]. \end{aligned} \tag{7.25}$$

Using the above measurement model, exact correction equations of the multiple-model Bayesian random set filter for joint signal attack detection, mode and state estimation in the case of extra packet injection attack are obtained as follows.

Note that from now on the notation $\langle \alpha, \beta \rangle = \int \alpha(x)\beta(x)\mathrm{d}x$ will be used for the inner product of two functions.

Theorem 9. *Assume that the prior density at time k is multiple model hybrid Bernoulli of the form*

$$p(\mathcal{A}_k, x_k, \nu_k|\mathcal{Z}^{k-1}) = \begin{cases} (1 - r_{k|k-1}) p^0_{k|k-1}(x_k, \nu_k), & \text{if } \mathcal{A}_k = \emptyset \\ r_{k|k-1} \cdot p^1_{k|k-1}(a_k, x_k, \nu_k), & \text{if } \mathcal{A}_k = \{a_k\} \end{cases}. \tag{7.26}$$

Then, given the measurement random set $\mathcal{Z}_k$ defined in (7.3), also the posterior density at time k turns out to be multiple model hybrid Bernoulli of the form

$$p(\mathcal{A}_k, x_k, \nu_k|\mathcal{Z}^k) = \begin{cases} (1 - r_{k|k})\, p^0_{k|k}(x_k, \nu_k), & \text{if } \mathcal{A}_k = \emptyset \\ r_{k|k} \cdot p^1_{k|k}(a_k, x_k, \nu_k), & \text{if } \mathcal{A}_k = \{a_k\} \end{cases} \tag{7.27}$$

with parameters

$$r_{k|k} = \frac{1 - p_d\,(1 - \Gamma_1)}{1 - p_d(1 - \Gamma_0 + r_{k|k-1}\Gamma)}\, r_{k|k-1} \tag{7.28}$$

$$p^0_{k|k}(x_k, \nu_k) = \frac{1 - p_d\left[1 - \sum\limits_{y_k \in \mathcal{Z}_k} \dfrac{\ell(y_k|x_k, \nu_k)}{\xi\, \kappa(y_k)}\right]}{1 - p_d\,(1 - \Gamma_0)} p^0_{k|k-1}(x_k, \nu_k) \tag{7.29}$$

$$p^1_{k|k}(a_k, x_k, \nu_k) = \frac{1 - p_d\left[1 - \sum\limits_{y_k \in \mathcal{Z}_k} \dfrac{\ell(y_k|a_k, x_k, \nu_k)}{\xi\, \kappa(y_k)}\right]}{1 - p_d\,(1 - \Gamma_1)} p^1_{k|k-1}(a_k, x_k, \nu_k) \tag{7.30}$$

where

$$\Gamma_0 \triangleq \sum_{y_k \in \mathcal{Z}_k} \frac{\left\langle \ell(y_k|x_k, \nu_k), p^0_{k|k-1}(x_k, \nu_k) \right\rangle}{\xi\, \kappa(y_k)} \tag{7.31}$$

$$\Gamma_1 \triangleq \sum_{y_k \in \mathcal{Z}_k} \frac{\left\langle \ell(y_k|a_k, x_k, \nu_k), p^1_{k|k-1}(a_k, x_k, \nu_k) \right\rangle}{\xi\, \kappa(y_k)} \tag{7.32}$$

and $\Gamma \triangleq \Gamma_0 - \Gamma_1$.

Proof: The correction equation of the multiple model Bayes random set filter for secure state estimation under switching attacks with possible extra packet injection follows from a generalization of the Bayes rule (7.13), which yields

$$p(\mathcal{A}_k, x_k, \nu_k|\mathcal{Z}^k) = \frac{\lambda(\mathcal{Z}_k|\mathcal{A}_k, x_k, \nu_k)\, p(\mathcal{A}_k, x_k, \nu_k|\mathcal{Z}^{k-1})}{p(\mathcal{Z}_k|\mathcal{Z}^{k-1})} \tag{7.33}$$

where $\lambda(\mathcal{Z}_k|\mathcal{A}_k, x_k, \nu_k)$ is given by (7.24) and (7.25), while

$$\begin{aligned} p(\mathcal{Z}_k|\mathcal{Z}^{k-1}) &= \left\langle \lambda(\mathcal{Z}_k|\mathcal{A}_k, x_k, \nu_k), p(\mathcal{A}_k, x_k, \nu_k|\mathcal{Z}^{k-1}) \right\rangle \\ &= \left\langle \lambda(\mathcal{Z}_k|\emptyset, x_k, \nu_k), p(\emptyset, x_k, \nu_k|\mathcal{Z}^{k-1}) \right\rangle \\ &+ \left\langle \lambda(\mathcal{Z}_k|\{a_k\}, x_k, \nu_k), p(\{a_k\}, x_k, \nu_k|\mathcal{Z}^{k-1}) \right\rangle . \end{aligned} \tag{7.34}$$

The posterior probability of signal attack existence $r_{k|k}$ can be obtained from the posterior density (7.33) with $\mathcal{A}_k = \emptyset$ via

$$r_{k|k} = 1 - \iint p(\emptyset, x_k, \nu_k|\mathcal{Z}^k)\, \mathrm{d}x_k \mathrm{d}\nu_k \tag{7.35}$$

where - using (7.24), (7.26) and (7.34) in (7.33) - we have

$$p(\emptyset, x_k, \nu_k|\mathcal{Z}^k) = (1 - r_{k|k-1})\, p^0_{k|k-1}(x_k, \nu_k). \tag{7.36}$$

From (7.34), combining (7.24), (7.25), and (7.26), we obtain

$$p(\mathcal{Z}_k|\mathcal{Z}^{k-1}) = \gamma(\mathcal{F}_k) \left[1 - p_d + p_d(1 - r_{k|k-1})\Gamma_1 + p_d r_{k|k-1}\Gamma_2 \right] \tag{7.37}$$

which is subsequently used together with (7.36) and (7.24) to obtain $p(\emptyset, x_k, \nu_k|Z^k)$ from (7.33), and finally (7.28) via (7.35). Moreover, $p^0_{k|k}(x_k, \nu_k) = p(\emptyset, x_k, \nu_k|\mathcal{Z}^k)/(1 - r_{k|k})$, and the joint density for the system under attack can be easily derived from the posterior density with $\mathcal{A}_k = \{a_k\}$ by recalling that $p^1_{k|k}(a_k, x_k, \nu_k) = p(\{a_k\}, x_k, \nu_k|\mathcal{Z}^k)/r_{k|k}$, where

$$p(\{a_k\}, x_k, \nu_k|\mathcal{Z}^k) = r_{k|k-1} \cdot p^1_{k|k-1}(a_k, x_k, \nu_k)\, . \tag{7.38}$$

■

7.3.2 Dynamic model and prediction

Let us next introduce the dynamic model of the MM-HBRS $(\mathcal{A}, x, \nu)$ essential to derive the prediction equations. First, it is assumed that, in the case of a system under normal operation at time k, an attack a_{k+1} will be launched to the system by an adversary during the sampling interval with probability p_b. On the other hand, if the system is under attack (i.e., $\mathcal{A}_k$ is a singleton), it is supposed that the adversarial action will endure from time step k to time step $k + 1$ with probability p_s. It is further assumed that $(\mathcal{A}, x, \nu)$ is a Markov process with joint transitional density

$$p(\mathcal{A}_{k+1}, x_{k+1}, \nu_{k+1}|\mathcal{A}_k, x_k, \nu_k) = p(x_{k+1}, \nu_{k+1}|\mathcal{A}_k, x_k, \nu_k)\, p(\mathcal{A}_{k+1}|\mathcal{A}_k) \tag{7.39}$$

which ensues from considering the attack as a stochastic process independent of the system state. In addition, note that

$$p(x_{k+1}, \nu_{k+1}|\mathcal{A}_k, x_k, \nu_k) = \begin{cases} p(\nu_{k+1}|\nu_k)\, p(x_{k+1}|x_k, \nu_k), & \text{if } \mathcal{A}_k = \emptyset \\ p(\nu_{k+1}|\nu_k)\, p(x_{k+1}|a_k, x_k, \nu_k), & \text{if } \mathcal{A}_k = \{a_k\} \end{cases} \tag{7.40}$$

and that the dynamics of the Markov process $\mathcal{A}_k$ in (7.39) is Bernoulli, i.e.

$$p(\mathcal{A}_{k+1}|\emptyset) = \begin{cases} 1 - p_b, & \text{if } \mathcal{A}_{k+1} = \emptyset \\ p_b\, p(a_{k+1}), & \text{if } \mathcal{A}_{k+1} = \{a_{k+1}\} \end{cases} \tag{7.41}$$

$$p(\mathcal{A}_{k+1}|\{a_k\}) = \begin{cases} 1 - p_s, & \text{if } \mathcal{A}_{k+1} = \emptyset \\ p_s\, p(a_{k+1}), & \text{if } \mathcal{A}_{k+1} = \{a_{k+1}\} \end{cases} \tag{7.42}$$

where $p(a_{k+1})$ is the PDF of the attack input vector. Clearly, when the attack vector is completely unknown, a non-informative PDF (e.g., uniform in the attack space) can be used as $p(a_{k+1})$. Under the above assumptions, an exact recursion for the prior density can be obtained, as stated in the following theorem.

Theorem 10. *Given the posterior multiple model hybrid Bernoulli density $p(\mathcal{A}_k, x_k, \nu_k|\mathcal{Z}^k)$ at time k of the form (7.27), fully characterized by the parameter triplet $\left(r_{k|k}, p^0_{k|k}(x_k, \nu_k), p^1_{k|k}(a_k, x_k, \nu_k)\right)$, also the predicted density turns out to be multiple model hybrid Bernoulli of the form*

$$p(\mathcal{A}_{k+1}, x_{k+1}, \nu_{k+1}|\mathcal{Z}^k) \tag{7.43}$$
$$= \begin{cases} (1 - r_{k+1|k})\, p^0_{k+1|k}(x_{k+1}, \nu_{k+1}), & \textit{if } \mathcal{A}_{k+1} = \emptyset \\ r_{k+1|k} \cdot p^1_{k+1|k}(a_{k+1}, x_{k+1}, \nu_{k+1}), & \textit{if } \mathcal{A}_{k+1} = \{a_{k+1}\} \end{cases}$$

with parameters

$$r_{k+1|k} = (1 - r_{k|k})\, p_b + r_{k|k}\, p_s \tag{7.44}$$

$$p^0_{k+1|k}(x_{k+1}, \nu_{k+1}) = \frac{(1 - r_{k|k})(1 - p_b)\, p_{k+1|k}(x_{k+1}|\emptyset)}{1 - r_{k+1|k}} + \frac{r_{k|k}(1 - p_s)\, p_{k+1|k}(x_{k+1}|\{a_k\})}{1 - r_{k+1|k}} \tag{7.45}$$

$$p^1_{k+1|k}(a_{k+1}, x_{k+1}, \nu_{k+1}) = \frac{(1 - r_{k|k})\, p_b\, p_{k+1|k}(x_{k+1}|\emptyset)\, p(a_{k+1})}{r_{k+1|k}} + \frac{r_{k|k}\, p_s\, p_{k+1|k}(x_{k+1}|\{a_k\})\, p(a_{k+1})}{r_{k+1|k}} \tag{7.46}$$

where

$$p_{k+1|k}(x_{k+1}|\emptyset) = \left\langle p(x_{k+1}|x_k, \nu_k), p^0_{k|k}(x_k, \nu_k) \right\rangle \tag{7.47}$$

$$p_{k+1|k}(x_{k+1}|\{a_k\}) = \left\langle p(x_{k+1}|a_k, x_k, \nu_k), p^1_{k|k}(a_k, x_k, \nu_k) \right\rangle. \tag{7.48}$$

Proof: The prediction equation of the multiple model Bayes random set filter is given by the following generalization of the Chapman-Kolmogorov equation (7.12)

$$p(\mathcal{A}_{k+1}, x_{k+1}, \nu_{k+1}|\mathcal{Z}^k) = \left\langle p(\mathcal{A}_{k+1}, x_{k+1}, \nu_{k+1}|\mathcal{A}_k, x_k, \nu_k), p(\mathcal{A}_k, x_k, \nu_k|\mathcal{Z}^k) \right\rangle$$
$$= (1 - r_{k|k}) \left\langle p(\mathcal{A}_{k+1}, x_{k+1}, \nu_{k+1}|\emptyset, x_k, \nu_k), p^0_{k|k}(x_k, \nu_k) \right\rangle$$
$$+ r_{k|k} \left\langle p(\mathcal{A}_{k+1}, x_{k+1}, \nu_{k+1}|\{a_k\}, x_k, \nu_k), p^1_{k|k}(a_k, x_k, \nu_k) \right\rangle$$

where the set integral definition (7.14) and (7.27) have been used. Then, we solve for $\mathcal{A}_{k+1} = \emptyset$. From (7.39), (7.40), and (7.41), one has

$$p(\emptyset, x_{k+1}, \nu_{k+1}|\mathcal{Z}^k) \tag{7.49}$$
$$= (1 - r_{k|k})(1 - p_b)\, p(\nu_{k+1}|\nu_k) \left\langle p(x_{k+1}|x_k, \nu_k), p^0_{k|k}(x_k, \nu_k) \right\rangle$$
$$+ r_{k|k}(1 - p_s)\, p(\nu_{k+1}|\nu_k) \left\langle p(x_{k+1}|a_k, x_k, \nu_k), p^1_{k|k}(a_k, x_k, \nu_k) \right\rangle.$$

Next, using (7.47) and (7.48), (7.49) becomes

$$p(\emptyset, x_{k+1}, \nu_{k+1}|\mathcal{Z}^k) \tag{7.50}$$
$$= (1 - r_{k|k})\, (1 - p_b)\, p(\nu_{k+1}|\nu_k)\, p_{k+1|k}(x_{k+1}|\emptyset)$$
$$+ r_{k|k}\, (1 - p_s)\, p(\nu_{k+1}|\nu_k)\, p_{k+1|k}(x_{k+1}|\{a_k\})$$

Analogously, for $\mathcal{A}_{k+1} = \{a_{k+1}\}$ we obtain

$$p(\{a_{k+1}\}, x_{k+1}, \nu_{k+1}|\mathcal{Z}^k) = \Big[(1 - r_{k|k})\, p_b\, p(\nu_{k+1}|\nu_k)\, p_{k+1|k}(x_{k+1}|\emptyset) + r_{k|k}\, p_s\, p(\nu_{k+1}|\nu_k)\, p_{k+1|k}(x_{k+1}|\{a_k\})\Big]\, p(a_{k+1})\,.$$

■

7.4 Gaussian-mixture implementation

Although no closed-form solution to the Bayes optimal recursion is admitted in general, for the special class of linear Gaussian models it is possible to analytically propagate in time the posterior densities $p^0_{k|k}(\cdot)$ and $p^1_{k|k}(\cdot)$ in the form of Gaussian mixtures (weights, means and covariances), and the probability of a signal attack. Note that in the case of nonlinear models and/or non-Gaussian noises, the solution can be obtained via nonlinear extensions of the GM approximation (e.g. Unscented/Extended GM) or sequential Monte Carlo methods (i.e. particle filter).

Denoting by $\mathcal{N}(x; m, P)$ a Gaussian PDF in the variable x, with mean m and covariance P, the closed-form solution assumes linear Gaussian observation and transition models conditioned on the modal state, i.e. for each mode $i \in \mathfrak{M}$ one has

$$\ell(y_k|x_k, \nu_k = i) = \mathcal{N}(y; C^i x_k, R^i) \tag{7.51}$$

$$\ell(y_k|a_k, x_k, \nu_k = i) = \mathcal{N}(y; C^i x_k + H^i a_k, R^i) \tag{7.52}$$

$$p(x_{k+1}|x_k, \nu_k = i) = \mathcal{N}(x; A^i x_k, Q^i) \tag{7.53}$$

$$p(x_{k+1}|a_k, x_k, \nu_k = i) = \mathcal{N}(x; A^i x_k + G^i a_k, Q^i) \tag{7.54}$$

In addition, the (a priori) signal attack model can be expressed as a Gaussian mixture of the form

$$p(a) = \sum_{j=1}^{J^a} \tilde{\omega}^{a,j} \mathcal{N}(a; \tilde{a}^j, \tilde{P}^{a,j}) \tag{7.55}$$

and the probabilities of signal attack survival p_s and measurement delivery p_d are assumed independent of both the system state and mode, i.e.

$$p_s(x, \nu) = p_s \tag{7.56}$$

$$p_d(x, \nu) = p_d\,. \tag{7.57}$$

In the GM implementation, each probability density at time k conditioned on mode $\nu_k = i$ is represented by the following set of parameters

$$\begin{aligned}&\Big(r_{k|k}, p^{0,i}_{k|k}(x_k), p^{1,i}_{k|k}(a_k, x_k)\Big)\\ &= \Big(r_{k|k}, \{\omega^{0,ij}_{k|k}, m^{0,ij}_{k|k}, P^{0,ij}_{k|k}\}_{j=1}^{J^0_{k|k}}, \{\omega^{1,ij}_{k|k}, m^{1,ij}_{k|k}, P^{1,ij}_{k|k}\}_{j=1}^{J^1_{k|k}}\Big), \qquad i \in \mathfrak{M}\end{aligned}$$

where symbols ω and J denote, respectively, weights and number of mixture components such that

$$\begin{aligned}p^{0,i}_{k|k}(x_k) &= p^0_{k|k}(x_k, \nu_k = i) = \sum_{j=1}^{J^0_{k|k}} \omega^{0,ij}_{k|k} \mathcal{N}(m^{0,ij}_{k|k}, P^{0,ij}_{k|k})\\ p^{1,i}_{k|k}(a_k, x_k) &= p^1_{k|k}(a_k, x_k, \nu_k = i) = \sum_{j=1}^{J^1_{k|k}} \omega^{1,ij}_{k|k} \mathcal{N}(m^{1,ij}_{k|k}, P^{1,ij}_{k|k}).\end{aligned}$$

In the above equation we defined $m^0_{k|k} = \hat{x}^0_{k|k}$, $m^1_{k|k} = [\hat{x}^{1^T}_{k|k}, \hat{a}^T_k]^T$, $P^0_{k|k} \triangleq \mathbb{E}[(x_k - \hat{x}^0_{k|k})(x_k - \hat{x}^0_{k|k})^T]$, $P^1_{k|k} = \begin{bmatrix} P^{1x}_{k|k} & P^{xa}_k \\ P^{ax}_k & P^a_k \end{bmatrix}$, and $P^{1x}_{k|k} \triangleq \mathbb{E}[(x_k - \hat{x}^1_{k|k})(x_k - \hat{x}^1_{k|k})^T]$, $(P^{xa}_k)^T = P^{ax}_k \triangleq \mathbb{E}[(a_k - \hat{a}_k)(x_k - \hat{x}^1_{k|k})^T]$, $P^a_k \triangleq \mathbb{E}[(a_k - \hat{a}_k)(a_k - \hat{a}_k)^T]$. The weights are such that $\sum_{j=1}^{J^0_{k|k}} \omega^{0,j}_{k|k} = 1$, and $\sum_{j=1}^{J^1_{k|k}} \omega^{1,j}_{k|k} = 1$.

The Gaussian Mixture implementation of the *Multiple Model Hybrid Bernoulli Filter* (GM-MM-HBF) is described as follows.

7.4.1 GM-MM-HBF correction

Proposition 3. *Suppose assumptions* (7.51)-(7.55) *hold, the predicted FISST density at time k is fully specified by the triplet*

$$\big(r_{k|k-1}, p^0_{k|k-1}(x_k, \nu_k), p^1_{k|k-1}(a_k, x_k, \nu_k)\big)$$

and $p^0_{k|k-1}(\cdot)$, $p^1_{k|k-1}(\cdot)$ *for each* $i \in \mathfrak{M}$ *are Gaussian mixtures of the form*

$$p^{0,i}_{k|k-1}(x_k) = \sum_{j=1}^{J^{0,i}_{k|k-1}} \omega^{0,ij}_{k|k-1} \mathcal{N}(m^{0,ij}_{k|k-1}, P^{0,ij}_{k|k-1}) \tag{7.58}$$

$$p^{1,i}_{k|k-1}(a_k, x_k) = \sum_{j=1}^{J^{1,i}_{k|k-1}} \omega^{1,ij}_{k|k-1} \mathcal{N}(m^{1,ij}_{k|k-1}, P^{1,ij}_{k|k-1}) \tag{7.59}$$

where $m^{0,ij}_{k|k-1} = \hat{x}^{0,ij}_{k|k-1}$, $m^{1,ij}_{k|k-1} = [(\hat{x}^{1,ij}_{k|k-1})^T, (\hat{a}^j_k)^T]^T$, $\sum_{j=1}^{J^{0,i}_{k|k-1}} \omega^{0,ij}_{k|k-1} = 1$, *and* $\sum_{j=1}^{J^{1,i}_{k|k-1}} \omega^{1,ij}_{k|k-1} = 1$.
Then the posterior FISST density $\left(r_{k|k}, p^0_{k|k}(x_k, \nu_k), p^1_{k|k}(a_k, x_k, \nu_k)\right)$ *for each mode* i *is given by*

$$r_{k|k} = \frac{1 - p_d + p_d\,\Gamma_1}{1 - p_d + p_d(1 - r_{k|k-1})\Gamma_0 + p_d\, r_{k|k-1}\Gamma_1}\, r_{k|k-1} \tag{7.60}$$

$$p^{0,i}_{k|k}(a_k, x_k) = \sum_{j=1}^{J^{0,i}_{k|k}} \omega^{0,ij}_{k|k}\mathcal{N}(m^{0,ij}_{k|k}, P^{0,ij}_{k|k}) \tag{7.61}$$

$$= \sum_{j=1}^{J^{0,i}_{k|k-1}} \omega^{0,ij}_{\bar{D},k|k}\mathcal{N}(m^{0,ij}_{k|k-1}, P^{0,ij}_{k|k-1}) + \sum_{y_k \in \mathcal{Z}_k} \sum_{j=1}^{J^{0,i}_{k|k-1}} \omega^{0,ij}_{D,k|k}\mathcal{N}(m^{0,ij}_{k|k}, P^{0,ij}_{k|k})$$

$$p^{1,i}_{k|k}(a_k, x_k) = \sum_{j=1}^{J^{1,i}_{k|k}} \omega^{1,ij}_{k|k}\mathcal{N}(m^{1,ij}_{k|k}, P^{1,ij}_{k|k}) \tag{7.62}$$

$$= \sum_{j=1}^{J^{1,i}_{k|k-1}} \omega^{1,ij}_{\bar{D},k|k}\mathcal{N}(m^{1,ij}_{k|k-1}, P^{1,ij}_{k|k-1}) + \sum_{y_k \in \mathcal{Z}_k} \sum_{j=1}^{J^{1,i}_{k|k-1}} \omega^{1,ij}_{D,k|k}\mathcal{N}(m^{1,ij}_{k|k}, P^{1,ij}_{k|k})$$

where we denote, for $b = 0, 1$*:*

$$\omega^{b,ij}_{\bar{D},k|k} = \frac{(1 - p_d)\,\omega^{b,ij}_{k|k-1}}{\Delta_b}, \quad \omega^{b,ij}_{D,k|k} = \frac{p_d\,\omega^{b,ij}_{k|k-1} q^{b,ij}_k(y_k)}{\Delta_b\,\xi\,\kappa(y_k)} \tag{7.63}$$

$$\Delta_b = 1 - p_d + p_d \sum_{y_k \in \mathcal{Z}_k} \sum_{h \in \mathfrak{M}} \sum_{l=1}^{J^{1,h}_{k|k-1}} \frac{\omega^{b,hl}_{k|k-1}}{\xi\,\kappa(y_k)} q^{b,hl}_k(y_k)$$

and

$$\begin{aligned} q^{0,ij}_k(y_k) &= \mathcal{N}(y_k; C^i m^{0,ij}_{k|k-1}, C^i P^{0,ij}_{k|k-1} C^{i^T} + R^i) \\ q^{1,ij}_k(y_k) &= \mathcal{N}(y_k; \tilde{C}^i m^{1,ij}_{k|k-1}, \tilde{C}^i P^{1,ij}_{k|k-1} \tilde{C}^{i^T} + R^i) \end{aligned}$$

with $\tilde{C}^i \triangleq [C^i, H^i]$.

Proof: We first derive the posterior density $p^{1,i}_{k|k}(\cdot)$, then $p^{0,i}_{k|k}(\cdot)$ can be obtained analogously.

From Theorem 3:

$$p^{1,i}_{k|k}(a_k,x_k) = \frac{1-p_d}{1-p_d+p_d\Gamma_1}\, p^{1,i}_{k|k-1}(a_k,x_k) \tag{7.64}$$
$$+\frac{p_d}{1-p_d+p_d\Gamma_1}\sum_{y_k\in\mathcal{Z}_k}\frac{\ell(y_k|a_k,x_k,\nu_k=i)}{\xi\,\kappa(y_k)}p^{1,i}_{k|k-1}(a_k,x_k)$$

where

$$\Gamma_1 = \sum_{y_k\in\mathcal{Z}_k}\frac{\left\langle \ell(y_k|a_k,x_k,\nu_k=i), p^{1,i}_{k|k-1}(a_k,x_k)\right\rangle}{\xi\,\kappa(y_k)}. \tag{7.65}$$

By substituting (7.59) and (7.52) into (7.65), we obtain

$$p^{1,i}_{k|k}(a_k,x_k) = \sum_{j=1}^{J^{1,i}_{k|k-1}}\frac{1-p_d}{1-p_d+p_d\Gamma_1}\,\omega^{1,ij}_{k|k-1}\mathcal{N}(m^{1,ij}_{k|k-1},P^{1,ij}_{k|k-1})$$

$$+\sum_{y_k\in\mathcal{Z}_k}\sum_{j=1}^{J^{1,i}_{k|k-1}}\omega^{1,ij}_{k|k-1}\frac{p_d}{1-p_d+p_d\Gamma_1}\frac{\mathcal{N}(y;C^ix_k+H^ia_k,R^i)}{\xi\,\kappa(y_k)}\mathcal{N}(m^{1,ij}_{k|k-1},P^{1,ij}_{k|k-1}).$$

Then, by applying a standard result for Gaussian functions [112, Lemma 2], we can write

$$\mathcal{N}(y;C^ix_k+H^ia_k,R^i)\mathcal{N}(m^{1,ij}_{k|k-1},P^{1,ij}_{k|k-1}) = q^{1,ij}_k(y_k)\mathcal{N}(m^{1,ij}_{k|k},P^{1,ij}_{k|k})$$

where

$$q^{1,ij}_k(y_k) = \mathcal{N}(y_k;\tilde{C}^im^{1,ij}_{k|k-1},\tilde{C}^iP^{1,ij}_{k|k-1}\tilde{C}^{i^T}+R^i). \tag{7.66}$$

In the special case of linear Gaussian models, $m^{1,ij}_{k|k}$ and $P^{1,ij}_{k|k}$ can be calculated following the correction step of the filter for joint input and state estimation of linear discrete-time systems [109], introduced in Section 7.2.4. In particular, $m^{1,ij}_{k|k}$ consists of

$$\begin{aligned}\hat{x}^{1,ij}_{k|k} &= \hat{x}^{1,ij}_{k|k-1}+\tilde{L}^{1,ij}_k(y_k-C^i\hat{x}^{1,ij}_{k|k-1}-H^i\hat{a}^{ij}_k) = L^{1,ij}_k(y_k-C^i\hat{x}^{1,ij}_{k|k-1})\\ \hat{a}^{ij}_k &= M^{ij}_k(y_k-C^i\hat{x}^{1,ij}_{k|k-1})\end{aligned} \tag{7.67}$$

where

$$L^{1,ij}_k = \tilde{L}^{1,ij}_k(I-H^iM^{ij}_k) \tag{7.68}$$
$$\tilde{L}^{1,ij}_k = P^{1,ij}_{k|k-1}C^{i^T}(S^{1,ij}_k)^{-1} \tag{7.69}$$
$$S^{1,ij}_k = C^iP^{1,ij}_{k|k-1}C^{i^T}+R^i \tag{7.70}$$
$$M^{ij}_k = \left[H^{i^T}(S^{1,ij}_k)^{-1}H^i\right]^{-1}H^{i^T}(S^{1,ij}_k)^{-1}. \tag{7.71}$$

The elements composing $P_{k|k}^{1,ij}$ can be computed as

$$P_k^{a,ij} = (H^{i^T}(S_k^{1,ij})^{-1}H^i)^{-1} \tag{7.72}$$

$$P_{k|k}^{1x,ij} = (I - L_k^{1,ij}C^i)P_{k|k-1}^{1,ij} \tag{7.73}$$

$$P_k^{xa,ij} = (P_k^{ax,ij})^T = -\tilde{L}_k^{1,ij}H^i P_k^{a,ij}. \tag{7.74}$$

In addition, Γ_1 is obtained by substituting (7.52) and (7.59) in (7.65), and by using integration (7.19) we obtain

$$\Gamma_1 = \sum_{y_k \in \mathcal{Z}_k} \sum_{h \in \mathfrak{M}} \sum_{l=1}^{J_{k|k-1}^{1,h}} \frac{\omega_{k|k-1}^{1,hl}}{\xi\, \kappa(y_k)} q_k^{1,hl}(y_k). \tag{7.75}$$

Thus, by substituting (7.66) and (7.75) in (7.66), with means and covariances given by (7.67) and (7.72)-(7.74), we can write

$$p_{k|k}^{1,i}(a_k, x_k) = \sum_{j=1}^{J_{k|k}^{1,i}} \omega_{k|k}^{1,ij} \mathcal{N}(m_{k|k}^{1,ij}, P_{k|k}^{1,ij}) \tag{7.76}$$

which comprises $J_{k|k-1}^{1,i}(1 + |\mathcal{Z}_k|)$ components, i.e.

$$p_{k|k}^{1,i}(a_k, x_k) = \sum_{j=1}^{J_{k|k-1}^{1,i}} \omega_{\bar{D},k|k}^{1,ij} \mathcal{N}(m_{k|k-1}^{1,ij}, P_{k|k-1}^{1,ij}) + \sum_{y_k \in \mathcal{Z}_k} \sum_{j=1}^{J_{k|k-1}^{1,i}} \omega_{D,k|k}^{1,ij} \mathcal{N}(m_{k|k}^{1,ij}, P_{k|k}^{1,ij})$$

with weights given by (7.63). Note that, as we can see from above, it turns out that $J_{k|k}^{1,i} = J_{k|k-1}^{1,i} + |\mathcal{Z}_k|\, J_{k|k-1}^{1,i} = J_{k|k-1}^{1,i}(1 + |\mathcal{Z}_k|)$, where the first *legacy* components correspond to the fact that no measurement has been delivered, while the remaining components are the ones corrected when one or multiple measurements are received.

Following the same rationale, analogous results can be obtained for $p_{k|k}^{0,i}$, with the exception that no signal attack estimate is obviously performed. The corrected probability of signal attack existence can be written, from (7.28), as

$$r_{k|k} = \frac{1 - p_d + p_d\, \Gamma_1}{1 - p_d + p_d(1 - r_{k|k-1})\Gamma_0 + p_d\, r_{k|k-1}\Gamma_1}\, r_{k|k-1} \tag{7.77}$$

where Γ_1 is given in (7.75), and

$$\Gamma_0 = \sum_{y_k \in \mathcal{Z}_k} \sum_{h \in \mathfrak{M}} \sum_{l=1}^{J_{k|k-1}^{0,h}} \frac{\omega_{k|k-1}^{0,hl}}{\xi\, \kappa(y_k)} q_k^{0,hl}(y_k)\,. \tag{7.78}$$

7.4.2 GM-MM-HBF prediction

Proposition 4. *Suppose assumptions* (7.51)-(7.55) *hold, the posterior FISST density at time k is fully specified by the triplet* $\big(r_{k|k}, p^0_{k|k}(x_k,\nu_k), p^1_{k|k}(a_k,x_k,\nu_k)\big)$, *and* $p^0_{k|k}(\cdot)$, $p^1_{k|k}(\cdot)$ *for each* $i \in \mathfrak{M}$ *are Gaussian mixtures of the form*

$$p^{0,i}_{k|k}(x_k) = \sum_{j=1}^{J^{0,i}_{k|k}} \omega^{0,ij}_{k|k} \mathcal{N}(m^{0,ij}_{k|k}, P^{0,ij}_{k|k}) \tag{7.79}$$

$$p^{1,i}_{k|k}(a_k,x_k) = \sum_{j=1}^{J^{1,i}_{k|k}} \omega^{1,ij}_{k|k} \mathcal{N}(m^{1,ij}_{k|k}, P^{1,ij}_{k|k}). \tag{7.80}$$

Then the predicted FISST density

$$\big(r_{k+1|k}, p^0_{k+1|k}(x_{k+1},\nu_{k+1}), p^1_{k+1|k}(a_{k+1},x_{k+1},\nu_{k+1})\big)$$

for each mode i is given by

$$r_{k+1|k} = (1-r_{k|k})p_b + r_{k|k}p_s \tag{7.81}$$

$$p^{0,i}_{k+1|k}(x_{k+1}) = \sum_{j=1}^{J^{0,i}_{k+1|k}} \omega^{0,ij}_{k+1|k} \mathcal{N}(m^{0,ij}_{k+1|k}, P^{0,ij}_{k+1|k}) \tag{7.82}$$

$$p^{1,i}_{k+1|k}(a_{k+1},x_{k+1}) = \sum_{j=1}^{J^{1,i}_{k+1|k}} \omega^{1,ij}_{k+1|k} \mathcal{N}(m^{1,ij}_{k+1|k}, P^{1,ij}_{k+1|k}) \tag{7.83}$$

where (7.82) *can be written as*

$$p^{0,i}_{k+1|k}(x_{k+1}) = \underbrace{\sum_{h\in\mathfrak{M}} \sum_{j=1}^{J^{0,h}_{k|k}} \omega^{0,hj}_{\bar{B},k+1|k} \mathcal{N}(m^{0,hj}_{\bar{B},k+1|k}, P^{0,hj}_{\bar{B},k+1|k})}_{no\ attack\text{-}birth} + \underbrace{\sum_{h\in\mathfrak{M}} \sum_{j=1}^{J^{1,h}_{k|k}} \omega^{0,hj}_{\bar{S},k+1|k} \mathcal{N}(m^{0,hj}_{\bar{S},k+1|k}, P^{0,hj}_{\bar{S},k+1|k})}_{no\ attack\text{-}survival} \tag{7.84}$$

with

$$\begin{aligned} m^{0,hj}_{\bar{B},k+1|k} &= A^h m^{0,hj}_{k|k} \\ P^{0,hj}_{\bar{B},k+1|k} &= A^h P^{0,hj}_{k|k} A^{h^T} + Q^h \\ \omega^{0,hj}_{\bar{B},k+1|k} &= \frac{(1-r_{k|k})(1-p_b)}{1-r_{k+1|k}}\,\pi_{hi}\,\omega^{0,hj}_{k|k} \end{aligned}$$

$$\begin{aligned} m^{0,hj}_{\bar{S},k+1|k} &= \tilde{A}^h m^{1,hj}_{k|k} \\ P^{0,hj}_{\bar{S},k+1|k} &= \tilde{A}^h P^{1,hj}_{k|k} \tilde{A}^{h^T} + Q^h \\ \omega^{0,hj}_{\bar{S},k+1|k} &= \frac{r_{k|k}(1-p_s)}{1-r_{k+1|k}}\,\pi_{hi}\,\omega^{1,hj}_{k|k}. \end{aligned}$$

Moreover, (7.83) *can be written as*

$$\begin{aligned} & p^{1,i}_{k+1|k}(a_{k+1}, x_{k+1}) \\ &= \underbrace{\sum_{h\in\mathfrak{M}} \sum_{j=1}^{J^{0,h}_{k|k}} \sum_{l=1}^{J^a} \omega^{1,hjl}_{B,k+1|k}\, \mathcal{N}(m^{1,hjl}_{B,k+1|k}, P^{1,hjl}_{B,k+1|k})}_{attack\text{-}birth} \\ &+ \underbrace{\sum_{h\in\mathfrak{M}} \sum_{j=1}^{J^{1,h}_{k|k}} \sum_{l=1}^{J^a} \omega^{1,hjl}_{S,k+1|k}\, \mathcal{N}(m^{1,hjl}_{S,k+1|k}, P^{1,hjl}_{S,k+1|k})}_{attack\text{-}survival} \end{aligned} \tag{7.85}$$

where

$$\begin{aligned} m^{1,hjl}_{B,k+1|k} &= \begin{bmatrix} A^h m^{0,hj}_{k|k} \\ \tilde{a}^l \end{bmatrix} \\ P^{1,hjl}_{B,k+1|k} &= \begin{bmatrix} A^h P^{0,hj}_{k|k} A^{h^T} + Q^h & 0 \\ 0 & \tilde{P}^{a,l} \end{bmatrix} \\ \omega^{1,hjl}_{B,k+1|k} &= \frac{(1-r_{k|k})\,p_b}{r_{k+1|k}}\,\pi_{hi}\,\omega^{0,hj}_{k|k}\,\tilde{\omega}^{a,l} \end{aligned}$$

$$
\begin{aligned}
m^{1,hjl}_{S,k+1|k} &= \begin{bmatrix} \tilde{A}^h m^{1,hj}_{k|k} \\ \tilde{a}^l \end{bmatrix} \\
P^{1,hjl}_{S,k+1|k} &= \begin{bmatrix} \tilde{A}^h P^{1,hj}_{k|k} \tilde{A}^{h^T} + Q^h & 0 \\ 0 & \tilde{P}^{a,l} \end{bmatrix} \\
\omega^{1,hjl}_{S,k+1|k} &= \frac{r_{k|k}\, p_s}{r_{k+1|k}}\, \pi_{hi}\, \omega^{1,hj}_{k|k} \tilde{\omega}^{a,l}.
\end{aligned}
$$

Proof: The predicted probability of signal attack existence comes directly from (7.44). Let us now derive the posterior density $p^{1,i}_{k|k}(\cdot)$.
From (7.46) in Theorem 4:

$$
\begin{aligned}
&p^{1,i}_{k+1|k}(a_{k+1}, x_{k+1}) \qquad (7.86)\\
&\quad = \frac{(1 - r_{k|k})\, p_b\, p(\nu_{k+1} = i|\nu_k)}{r_{k+1|k}} \Big\langle p(x_{k+1}|x_k, \nu_k = i), p^0_{k|k}(x_k, \nu_k = i) \Big\rangle\, p(a) \\
&\quad + \frac{r_{k|k}\, p_s\, p(\nu_{k+1} = i|\nu_k)}{r_{k+1|k}} \Big\langle p(x_{k+1}|a_k, x_k, \nu_k = i), p^1_{k|k}(a_k, x_k, \nu_k = i) \Big\rangle\, p(a).
\end{aligned}
$$

Using (7.53), (7.79) in the first term, and (7.54), (7.80) in the second term, and recalling the definition of transitional probabilities (7.5), we can rewrite

$$
\begin{aligned}
p^{1,i}_{k+1|k}(a_{k+1}, x_{k+1}) &= \frac{(1 - r_{k|k})\, p_b}{r_{k+1|k}} \sum_{h \in \mathfrak{M}} \pi_{hi} \int \mathcal{N}(x; A^h x_k, Q^h) \qquad (7.87)\\
&\quad \times \sum_{j=1}^{J^{0,h}_{k|k}} \omega^{0,hj}_{k|k}\, \mathcal{N}(m^{0,hj}_{k|k}, P^{0,hj}_{k|k})\, \mathrm{d}x_k \sum_{l=1}^{J^a} \tilde{\omega}^{a,l}\, \mathcal{N}(a; \tilde{a}^l, \tilde{P}^{a,l}) \\
&\quad + \frac{r_{k|k}\, p_s}{r_{k+1|k}} \sum_{h \in \mathfrak{M}} \pi_{hi} \iint \mathcal{N}(x; A^h x_k + G^h a_k, Q^h) \\
&\quad \times \sum_{j=1}^{J^{1,h}_{k|k}} \omega^{1,hj}_{k|k}\, \mathcal{N}(m^{1,hj}_{k|k}, P^{1,hj}_{k|k})\, \mathrm{d}a_{k+1} \mathrm{d}x_k \sum_{l=1}^{J^a} \tilde{\omega}^{a,l}\, \mathcal{N}(a; \tilde{a}^l, \tilde{P}^{a,l}).
\end{aligned}
$$

Hence, using Lemma 1 in [113], we finally derive (7.85).

In a similar fashion, we can obtain $p^{0,i}_{k+1|k}$. From (7.45) in Theorem 4:

$$
\begin{aligned}
&p^{0,i}_{k+1|k}(x_{k+1}) \qquad (7.88)\\
&\quad = \frac{(1 - r_{k|k})(1 - p_b)\, p(\nu_{k+1} = i|\nu_k)}{1 - r_{k+1|k}} \Big\langle p(x_{k+1}|x_k, \nu_k = i), p^0_{k|k}(x_k, \nu_k = i) \Big\rangle \\
&\quad + \frac{r_{k|k}\, (1 - p_s)\, p(\nu_{k+1} = i|\nu_k)}{1 - r_{k+1|k}} \Big\langle p(x_{k+1}|a_k, x_k, \nu_k = i), p^1_{k|k}(a_k, x_k, \nu_k = i) \Big\rangle
\end{aligned}
$$

which leads to

$$p^{0,i}_{k+1|k}(x_{k+1}) = \frac{(1-r_{k|k})(1-p_b)}{1-r_{k+1|k}} \sum_{h\in\mathfrak{M}} \pi_{hi} \int \mathcal{N}(x; A^h x_k, Q^h) \quad (7.89)$$
$$\times \sum_{j=1}^{J^{0,h}_{k|k}} \omega^{0,hj}_{k|k}\, \mathcal{N}(m^{0,hj}_{k|k}, P^{0,hj}_{k|k})\, \mathrm{d}x_k$$
$$+ \frac{r_{k|k}(1-p_s)}{1-r_{k+1|k}} \sum_{h\in\mathfrak{M}} \pi_{hi} \iint \mathcal{N}(x; A^h x_k + G^h a_k, Q^h)$$
$$\times \sum_{j=1}^{J^{1,h}_{k|k}} \omega^{1,hj}_{k|k}\, \mathcal{N}(m^{1,hj}_{k|k}, P^{1,hj}_{k|k}) \sum_{j=1}^{J^{1,h}_{k|k}} \omega^{1,hj}_{k|k}\, \mathcal{N}(m^{1,hj}_{k|k}, P^{1,hj}_{k|k})\, \mathrm{d}x_k$$

and finally to (7.84). ■

7.5 Numerical example

In this section, we demonstrate the effectiveness of the proposed Bayesian random-set approach for secure CPS state estimation in the presence of mode/signal switching attacks, extra packet injection attacks as well as uncertainty on measurement delivery. The proposed approach can be easily applied to the case of malicious source estimation described in Section 7.2.3, which is analogous to the source estimation problem presented in Chapter 5 with the additional challenge introduced by the presence of adversarial cyber attacks. However, it is important to note that the Bayesian random set filter proposed in Section 7.3 is very general and independent of the particular application under consideration. Hence, it is shown how the same approach can be adopted in power systems, specifically for secure estimation of electric power grids.

Let us consider the Western System Coordinating Council (WSCC) 9-bus test case shown in Fig. 1, consisting of 3 synchronous generators, 3 generator terminal buses, and 3 load buses. The transmission lines' parameters, the inertia and the damping coefficients of generators are taken from [114]. The dynamics of the system can be described by the linearized swing equation for the $n = 6$ active buses, derived through the Kron reduction by [115] of the linear small-signal power network model. The n-dimensional state of the system comprises both the rotor angles and the frequencies of each generator. After discretization (with sampling interval $T = 0.01s$), the power

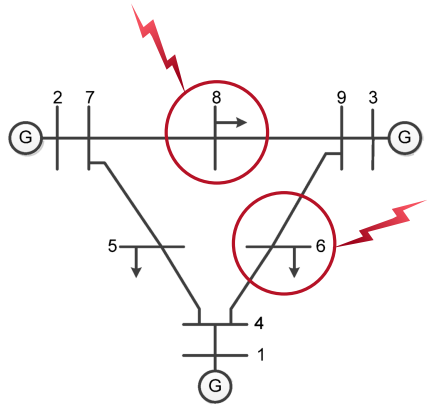


Figure 7.1: Single-line model of the WSCC 9-bus system. The *true* victim load buses 6 and 8 are circled in red.

system model takes the form (7.1)-(7.2), where each mode corresponds to one of the $\mathfrak{m} = 3$ different hypotheses on the set of vulnerable load buses $\mathcal{V}_1 = \{6, 8\}$, $\mathcal{V}_2 = \{5, 6\}$, and $\mathcal{V}_3 = \{5, 8\}$. At time $k = 50$ a signal attack vector $a_k = [0.05, 0.04]^T$ per-unit is injected into the system to abruptly increase the real power demand of the two victim load buses 6 and 8 with an additional loading of 5.56% and, respectively, 4%. This type of attack, referred to as *load altering attack* by [116], can provoke a loss of synchrony of the rotor angles and hence a deviation of the rotor speeds of all generators from the nominal value $\omega_s = 60$ Hz. In this numerical study, the probabilities of attack-birth and attack-survival are fixed, respectively, at $p_b = 0.2$ and $p_s = 0.8$. A network of 6 sensors is deployed to measure the state of the system. The system-generated measurement vector is supposed to be delivered at the monitor/control center with probability $p_d = 0.98$. The extra fake measurements injected into the sensor channel are modeled as a Poisson RFS with average number $\xi = 40$ and probability density uniformly distributed over the interval $[-10, 0]$, suitably chosen to emulate system-originated observations. Fig. 7.2 shows the resulting number of fake measurements maliciously injected at each time step and the cases of undelivered system-originated measurement.

For the joint task of signal attack detection and mode-state estimation, here we adopted the *Static* version (introduced in Section 2.2) of the GM-

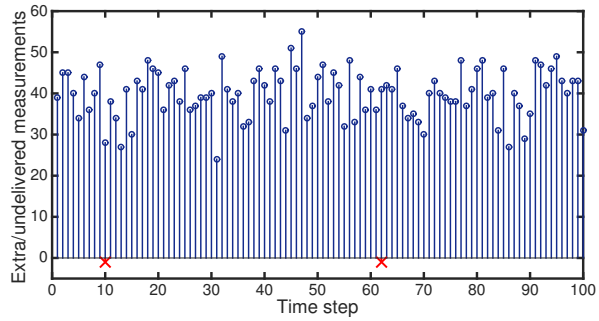


Figure 7.2: Number of extra fake measurements injected (blue circles), and cases of undelivered system-originated measurement (red cross in -1) vs time. The proposed approach turns out to be particularly robust to *extra packet injections*.

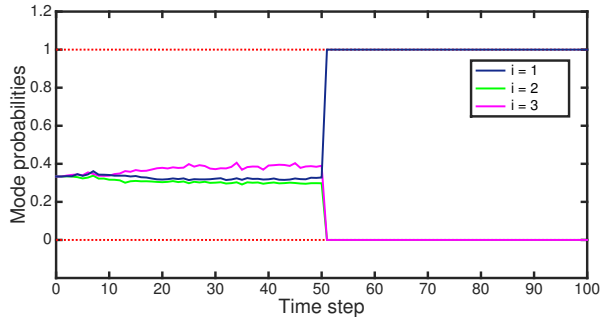


Figure 7.3: Mode probabilities $\bar{\mu}^i_{k|k}$, $i = 1, 2, 3$. The three possible attack modes of the system share similar probabilities within the time interval $[0, 49]$ when there is no signal attack. The different behaviour is revealed once a_k enters into action at time $k = 50$ and the unknown mode $i = 1$ is correctly estimated.

MM-HBF (described in Section 4). It can be noticed from Fig. 7.3 that the proposed secure state estimation algorithm succeeds in detecting the switching mode attack, and hence in estimating the true system's mode of operation (characterized by the highest mode probability) $i = 1$, corresponding to a

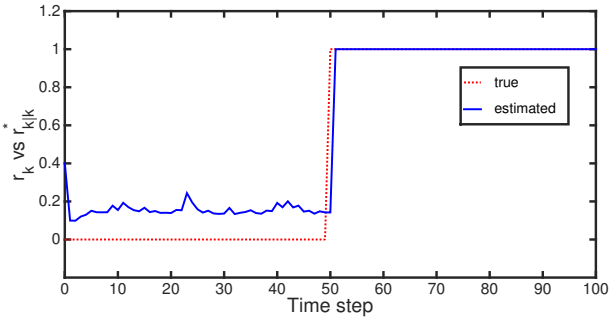


Figure 7.4: *True* (r_k) and estimated ($r^*_{k|k}$) probability of existence of the signal attack a_k.

load altering attack on $\mathcal{V}_1$. Note that the posterior mode probabilities shown in Fig. 7.3 are determined as follows:

$$\bar{\mu}^i_{k|k} = \frac{\omega^{0,i}_{k|k}(1 - r^i_{k|k}) + \omega^{1,i}_{k|k}\, r^i_{k|k}}{\sum_{i=1}^{\mathrm{m}} \omega^{0,i}_{k|k}(1 - r^i_{k|k}) + \omega^{1,i}_{k|k}\, r^i_{k|k}}, \quad i = 1, 2, 3.$$

Moreover, the proposed filter promptly detects the unknown signal attack, as it can be seen from the attack probability $r^*_{k|k}$ in Fig. 7.4 which takes the unitary value after time $k = 50$. At each time instant k the estimated attack probability $r^*_{k|k} = r^{i^*}_{k|k}$ can be computed from the estimated mode $i^* = \arg\max \bar{\mu}^i_{k|k}$.

7.6 Conclusions

It has been shown how to securely estimate the state of a cyber-physical system in presence of attacks of various types by which the cyber-attacker can simultaneously switch an attack signal and the attack mode, and can also inject fake measurements. All these ingredients have been incorporated in a random set stochastic Bayesian filtering problem where Bernoulli and Poisson random sets have been used to model the attack signal switching and, respectively, measurement injection while multiple models have been exploited to account for different attack modes. A recursive Bayesian filter solving the formulated problem has been derived and its Gaussian mixture

implementation has been developed and tested on a power network case study, exhibiting promising results in terms of prompt attack detection and resilient state estimation.

Chapter 8

Conclusion

8.1 Summary of contributions

The main contributions of this work can be summarized as follows.

Chapter 4: The contributions of this chapter are threefold. First, we develop *scalable* distributed filters for distributed-parameter systems by suitably adapting the so-called Schwarz decomposition methods [46–51], which allow to split the overall domain into smaller subdomains and assign each of them to different interconnected processing nodes. Second, we exploit the *finite element* (FE) method [35, 36, 52] in order to approximate the original infinite-dimensional filtering problem into a, possibly large-scale, finite-dimensional one. Combining these two ingredients, we propose a novel distributed *finite element Kalman filter* (FE-KF) which generalizes to the more challenging distributed case previous work on FE Kalman filtering [53, 54]. Moreover, we show that the parallel FE-based implementation of the Schwarz method on the overall system is equivalent to performing a particular time-discretization scheme on the interconnected subsystems, and we verify the well-posedness of the proposed discretization method in terms of numerical stability (i.e., in terms of boundedness and convergence of the time-discretization errors). Third, we provide results on the stability of the proposed distributed FE Kalman filter. Last but not least, a practical procedure, which requires the tuning of only one (or few) scalar parameters, is provided to check and guarantee the stability property.

Chapter 5: This chapter provides two major contributions to the source estimation problem. First, inspired by the classic notion of structural identifiability [66]- [67], this work defines the concept of *source identifiability*, i.e. the possibility of detecting the source and uniquely determining its position and intensity from available pointwise-in-time-and-space field measurements. Specifically, system-theoretic conditions for identifiability are derived in terms of rank tests on suitable polynomial matrices for both cases in which the source intensity is regarded as an unknown input or is modeled as the output of an appropriate exosystem. Second, we propose a robust field estimation strategy to reconstruct unknown sources in spatially distributed systems. In particular, a multiple-model Kalman filtering approach to source estimation is undertaken by considering all hypotheses (modes) corresponding to the source location in any possible element of the FE mesh plus a further hypothesis accounting for the possible source absence. Both cases of motionless source with unknown position and of moving source are addressed, leading to the design of two different algorithms for source estimation: the *Finite Element Static Multiple Model* (FE-SMM) and, respectively, the dynamic *Finite Element Interacting Multiple Model* (FE-IMM).

Chapter 6: The contributions of this chapter are threefold. First, relying on the so-called *noise-aided* paradigm, according to which in binary sensor networks the presence of measurement noise can be a helpful source of statistical information by randomly shifting the analog measurement, this chapter presents a novel approach to recursive state estimation given binary observations. The proposed approach is based on a *Moving-Horizon* (MH) approximation of the *Maximum A-posteriori Probability* (MAP) estimation and extends previous work [81]- [87] concerning parameter estimation to recursive state estimation. A further contribution is to show that for a linear system the optimization problem arising from the MH-MAP formulation turns out to be convex and, hence, practically feasible for real-time implementation. Moreover, the proposed optimization-based strategy, capable of dealing with physical constraints on state and noise variables, is applied to the challenging problem of dynamic field estimation over binary sensor networks, which convey the minimum amount of information. Finally, simulation results relative to a dynamic field estimation case-study have exhibited the conjectured noise-aided feature of the proposed estimator in that the estimation accuracy improves, starting from a null measurement noise, until the variance of the latter achieves an optimal value beyond which estimation

performance decays.

Chapter 7: The contributions of this chapter are threefold. First, the *joint attack detection and mode-state estimation* problem is formulated and solved following a stochastic Bayesian approach which exploits Bernoulli and Poisson random sets for modeling the attack presence/existence and, respectively, fake measurements, as well as multiple models for handling the different attack modes. The proposed approach can deal with nonlinear, noisy and perturbed systems. Additionally, it can encompass in a unique framework different types of attacks (switching signal and mode attacks, extra packet injection, packet substitution, etc.), and provides (discrete or continuous) probability distributions of the attack existence, attack mode, attack signal and system state which are very useful for taking decisions. Further, it is shown how the proposed general framework is well-suited to formulate the problem of dynamic field estimation in the presence of a malicious source injected into the spatially distributed system of interest. Finally, a Gaussian-mixture implementation of the joint attack detector and mode-state estimator has been developed based on the recursion derived for the *secure* Bayes-optimal filter.

8.2 Directions for future work

In this final section we present interesting directions for future research on dynamic field estimation in complex environments.

- The design of centralized and distributed field estimators exploiting the finite-element approximation discussed in Chapter 4 can be extended to the estimation of fields governed by nonlinear partial differential equations and/or different classes of PDEs, such as the wave equation arising in acoustics, optics, electromagnetics, fluid dynamics, and Navier-Stokes equations describing the dynamics of weather, ocean currents, water and air flows, etc.

- Future work on the problem of source estimation addressed in Chapter 5 may consider the multi-source case, which introduces the additional issues of an unknown (random) number of active sources altering the field of interest, and hence sensors measuring the field resulting from the combined effect of different sources. The consequent observation

model is said to be *superpositional* as measurements become functions of all the sources within the monitored region. The multi-source estimation problem amounts to estimating the number, intensity and location of all the diffusive sources present at every time step, as well as the overall induced field.

- Future research efforts will be also devoted to exploit the sparse and localized structure of the mass and stiffness matrices, originating from the application of the finite-element approximation presented in Chapter 4, in order to spatially decompose the overall system. In particular, the idea is to directly apply partition-based square-root filtering to the descriptor (implicit) system described in Section 4.3 (resulting from the spatial discretization of the original PDE system) so as to reduce the computational burden and improve the numerical properties of the field estimation scheme.

- Robust (with respect to the presence of unknown sources) centralized and distributed finite element Kalman filters can be employed for air quality monitoring in order to map the concentration of diffusive pollutants from measurements provided by a wireless network of environmental sensors deployed in known locations over an area of interest. Air quality monitoring of a complex urban environment involves multiple sources of pollution and imprecise awareness of the transport model parameters. Moreover, since a considerable contribution to air pollution issues is related to non-point sources, i.e. vehicles, it is also important to be able to estimate these traffic-induced emissions. Thus, this future research is motivated by the idea of identifying multiple sources and exploiting data fusion techniques in order to combine heterogeneous observations, i.e. pollution data, traffic flow estimates coming from real-time traffic monitoring systems, and meteorological measurements, such as wind speed and direction, temperature and humidity. The aggregated data allows for a better real-time reconstruction of the urban air quality, and ensures enhanced accuracy with respect to conventional monitors, which only measure the presence of contaminants in concentrated-in-space locations and rely on inaccurate steady-state models of the pollution propagation.

- Future directions on dynamic field estimation over binary sensor networks will concern stability properties of the MH-MAP state estimator

and the design of *fast* algorithms able to overcome the main limitation of the MH estimation approach, i.e. its need of on-line solutions of dynamic optimization problems. This issue becomes particularly relevant in large-scale applications where the solution of the MH-MAP problem might give rise to computational delays and thus efficient strategies are required to reduce the excessive computational burden.

- A possible direction of investigation relative to the *security* issues introduced in Chapter 7 may address the problem of detection-localization of malicious sources (e.g. biochemical attacks, fires) altering a field of interest characterized by PDE dynamics.

- Future steps will also include the design of distributed strategies for joint attack detection and secure field estimation. The task is to securely monitor the state of a cyber-physical system (governed by PDEs) over a cluster-based network wherein multiple fusion nodes collect data from sensors and cooperate in a neighborwise fashion by exchanging information and performing data fusion via non-secure communication links. If the attack detection-state estimation problem is formulated in the context of random set theory as in Chapter 7, the main issue becomes how to fuse local probability densities in a secure way, i.e. when it is not known how many and which densities received by neighboring nodes have been injected by attackers and hence are not reliable.

Appendix A

Publications

This research activity has led to several publications in international journals and conferences. These are summarized below.

International Journals

1. G. Battistelli, L. Chisci, **N. Forti**, G. Pelosi, and S. Selleri, "Distributed finite-element Kalman filter for field estimation", *IEEE Transactions on Automatic Control*, 2017. Accepted for publication.

2. E. Bou-Harb, W. Lucia, **N. Forti**, S. Weerakkody, N. Ghani, and B. Sinopoli, "Cyber meets control: A novel federated approach for resilient CPS leveraging real cyber threat intelligence", *IEEE Communications Magazine*, 2017. Accepted for publication.

Submitted

1. **N. Forti**, G. Battistelli, L. Chisci, S. Li, B. Wang, and B. Sinopoli, "Distributed joint attack detection and secure state estimation", *IEEE Transactions on Signal and Information Processing over Networks*, 2017.

International Conferences and Workshops

1. **N. Forti**, G. Battistelli, L. Chisci, and B. Sinopoli, "Secure state estimation of cyber-physical systems under switching attacks", *20th World Congress of the International Federation of Automatic Control (IFAC)*, Toulouse, France, 2017. Submitted.

2. **N. Forti**, G. Battistelli, L. Chisci, and B. Sinopoli, "A Bayesian approach for joint attack detection and resilient state estimation", in *Proc. of the 55th IEEE Conference on Decision and Control (CDC)*, Las Vegas, NV, 2016.
3. G. Battistelli, L. Chisci, **N. Forti**, and S. Gherardini, "MAP moving horizon state estimation with binary measurements", in *Proc. of the American Control Conference (ACC)*, pp. 5413–5418, Boston, MA, 2016.
4. G. Battistelli, L. Chisci, **N. Forti**, G. Pelosi, and S. Selleri, "Localized diffusive source estimation via an hybrid finite element/Kalman filtering approach", *International Workshop on Finite Elements for Microwave Engineering (FEM 2016)*, Florence, Italy, 2016.
5. G. Battistelli, L. Chisci, **N. Forti**, G. Pelosi, and S. Selleri, "Point source estimation via finite element multiple-model Kalman filtering", in *Proc. of the 54th IEEE Conference on Decision and Control (CDC)*, pp. 4984–4989, Osaka, Japan, 2015.
6. G. Battistelli, L. Chisci, **N. Forti**, G. Pelosi, and S. Selleri, "Distributed finite element Kalman filter", in *Proc. of the 14th European Control Conference (ECC)*, pp. 3695–3700, Linz, Austria, 2015.
7. G. Battistelli, L. Chisci, C. Fantacci, **N. Forti**, A. Farina, and A. Graziano, "Distributed peer-to-peer multitarget tracking with association-based track fusion", in *Proc. of the 17th International Conference on Information Fusion (FUSION)*, pp. 1–7, Salamanca, Spain, 2014.

National Conferences

1. G. Battistelli, L. Chisci, and **N. Forti**, "Dynamic field estimation in complex environments", *Convegno della Società Italiana Docenti e Ricercatori in Automatica (AUTOMATICA.IT 2016)*, Rome, Italy, 2016.
2. G. Battistelli, L. Chisci, **N. Forti**, G. Pelosi, and S. Selleri, "A finite-element based field estimation via a Kalman filtering approach", *Riunione Nazionale di Elettromagnetismo (RINEM 2016)*, Parma, Italy, 2016.

Technical Reports

1. G. Battistelli, L. Chisci, **N. Forti**, V. Salvo, A. Graziano, F. Ciaramaglia, G. Golino, A. Liburdi, and W. Mellano, "Fusion of GIS data and SAR images", Selex ES, Technical Report, 2015.

Bibliography

[1] A. Quarteroni, *Numerical models for differential problems*. Springer, 2014.

[2] M. Fisher, J. Nocedal, Y. Trémolet, and S. J. Wright, "Data assimilation in weather forecasting: A case study in PDE-constrained optimization," *Optimization and Engineering*, vol. 10, no. 3, pp. 409–426, 2009.

[3] M. Ortner, A. Nehorai, and A. Jeremic, "Biochemical transport modeling and Bayesian source estimation in realistic environments," *IEEE Transactions on Signal Processing*, vol. 55, no. 6, pp. 2520–2532, 2007.

[4] J. Mandel, L. S. Bennethum, J. D. Beezley, J. L. Coen, C. C. Douglas, M. Kim, and A. Vodacek, "A wildland fire model with data assimilation," *Mathematics and Computers in Simulation*, vol. 79, no. 3, pp. 584–606, 2008.

[5] H. Igel, M. Käser, and M. Stupazzini, *Simulation of seismic wave propagation in media with complex geometries*, pp. 7891–7914. Springer New York, 2009.

[6] G. Martelloni and F. Bagnoli, "Fractional and fractal dynamics approach to anomalous diffusion in porous media: Application to landslide behavior," in *EGU General Assembly Conference Abstracts*, vol. 18, p. 17206, 2016.

[7] L. Tartar, *An introduction to Navier-Stokes equation and oceanography*. Lecture Notes of the Unione Matematica Italiana, Springer Berlin Heidelberg, 2006.

[8] R. He and H. Gonzalez, "Zoned HVAC control via PDE-constrained optimization," in *2016 American Control Conference (ACC)*, pp. 587–592, 2016.

[9] S. Moura, J. Bendtsen, and V. Ruiz, "Observer design for boundary coupled PDEs: Application to thermostatically controlled loads in smart grids," in *52nd IEEE Conference on Decision and Control*, pp. 6286–6291, 2013.

[10] C. G. Claudel and A. M. Bayen, "Lax-Hopf based incorporation of internal boundary conditions into hamilton-jacobi equation. Part II: Computational methods," *IEEE Transactions on Automatic Control*, vol. 55, no. 5, pp. 1158–1174, 2010.

[11] J. de Halleux, C. Prieur, J.-M. Coron, B. d'Andrea Novel, and G. Bastin, "Boundary feedback control in networks of open channels," *Automatica*, vol. 39, no. 8, pp. 1365–1376, 2003.

[12] T. Taddei, J. D. Penn, M. Yano, and A. T. Patera, "Simulation-based classification: A model-order-reduction approach for structural health monitoring," *Archives of Computational Methods in Engineering*, pp. 1–23, 2016.

[13] L. Baudouin, C. Prieur, F. Guignard, and D. Arzelier, "Robust control of a bimorph mirror for adaptive optics systems," *Applied Optics*, vol. 47, no. 20, pp. 3637–3645, 2008.

[14] U. A. Khan and J. M. F. Moura, "Distributing the Kalman filter for large-scale systems," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 4919–4935, 2008.

[15] S. S. Stankovic, M. S. Stankovic, and D. M. Stipanovic, "Consensus based overlapping decentralized estimation with missing observations and communication faults," *Automatica*, vol. 45, no. 6, pp. 1397–1406, 2009.

[16] M. Farina, G. Ferrari-Trecate, and R. Scattolini, "Moving-horizon partition-based state estimation of large-scale systems," *Automatica*, vol. 46, no. 5, pp. 910–918, 2010.

[17] H. Zhang, J. M. F. Moura, and B. Krogh, "Dynamic field estimation using wireless sensor networks: Tradeoffs between estimation error and communication cost," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2383–2395, 2009.

[18] S. Das and J. M. F. Moura, "Distributed kalman filtering with dynamic observations consensus," *IEEE Transactions on Signal Processing*, vol. 63, no. 17, pp. 4458–4473, 2015.

[19] F. Dorfler, F. Pasqualetti, and F. Bullo, "Continuous-time distributed observers with discrete communication," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 296–304, 2013.

[20] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman, "Decentralized environmental modeling by mobile sensor networks," *IEEE Transactions on Robotics*, vol. 24, no. 3, pp. 710–724, 2008.

[21] J. Cortes, "Distributed kriged kalman filter for spatial estimation," *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2816–2827, 2009.

[22] D. B. Work, O. P. Tossavainen, S. Blandin, A. M. Bayen, T. Iwuchukwu, and K. Tracton, "An ensemble kalman filtering approach to highway traffic estimation using gps enabled mobile devices," in *2008 47th IEEE Conference on Decision and Control*, pp. 5062–5068, 2008.

[23] M. Rafiee, Q. Wu, and A. M. Bayen, "Kalman filter based estimation of flow states in open channels using lagrangian sensing," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 8266–8271, 2009.

[24] E. Kalnay, *Atmospheric modeling, data assimilation and predictability*. Cambridge University Press, 2003.

[25] M. Ghil and P. Malanotte-Rizzoli, "Data assimilation in meteorology and oceanography," vol. 33 of *Advances in Geophysics*, pp. 141–266, Elsevier, 1991.

[26] G. Evensen, *Data assimilation: The ensemble Kalman filter*. Earth and Environmental Science, Springer Berlin Heidelberg, 2009.

[27] M. A. Demetriou, "Design of consensus and adaptive consensus filters for distributed parameter systems," *Automatica*, vol. 46, no. 2, pp. 300–311, 2010.

[28] M. A. Demetriou, "Adaptive consensus filters of spatially distributed systems with limited connectivity," in *52nd IEEE Conference on Decision and Control*, pp. 442–447, 2013.

[29] J. Hoffman and S. Frankel, *Numerical methods for engineers and scientists*. Taylor & Francis, 2001.

[30] J. A. Trangenstein, *Numerical solutions of elliptic and parabolic partial differential equations*. Cambridge University Press, 2013.

[31] L. Evans, *Partial differential equations*. Graduate studies in mathematics, American Mathematical Society, 2010.

[32] A. Friedman, *Partial differential equations of parabolic type*. Dover Books on Mathematics, Dover Publications, 2013.

[33] T. Bergman, F. Incropera, D. DeWitt, and A. Lavine, *Fundamentals of heat and mass transfer*. Wiley, 2011.

[34] D. Hahn and M. Ozisik, *Heat conduction*. Wiley, 2012.

[35] S. Brenner and R. Scott, *The mathematical theory of finite element methods*. Texts in Applied Mathematics, Springer New York, 2007.

[36] G. Pelosi, R. Coccioli, and S. Selleri, *Quick finite elements for electromagnetic waves*. Artech House, 2009.

[37] P. Ciarlet, *The finite element method for elliptic problems*. Studies in Mathematics and its Applications, Elsevier Science, 1978.

[38] A. M. Quarteroni and A. Valli, *Numerical approximation of partial differential equations*. Springer Publishing Company, Incorporated, 1994.

[39] J. Akin, *Finite elements for analysis and design*. Computational mathematics and applications, Academic Press, 1994.

[40] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[41] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, 2005.

[42] G. C. Calafiore and F. Abrate, "Distributed linear estimation over sensor networks," *International Journal of Control*, vol. 82, no. 5, pp. 868–882, 2009.

[43] G. Battistelli and L. Chisci, "Kullback–Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability," *Automatica*, vol. 50, no. 3, pp. 707–718, 2014.

[44] G. Battistelli, L. Chisci, and C. Fantacci, "Parallel consensus on likelihoods and priors for networked nonlinear filtering," *IEEE Signal Processing Letters*, vol. 21, no. 7, pp. 787–791, 2014.

[45] G. Battistelli, L. Chisci, C. Fantacci, A. Farina, and A. Graziano, "Consensus CPHD filter for distributed multitarget tracking," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 508–520, 2013.

[46] A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*. Numerical mathematics and scientific computation, Clarendon Press, 1999.

[47] P. Lions, "On the Schwarz alternating method. I," in *First Int. Symp. on Domain Decomposition Methods for Partial Differential Equations*, pp. 1–142, 1988.

[48] M. J. Gander, "Schwarz methods over the course of time.," *ETNA. Electronic Transactions on Numerical Analysis*, vol. 31, pp. 228–255, 2008.

[49] T. F. Chan and T. P. Mathew, "Domain decomposition algorithms," 1994.

[50] A. Toselli and O. Widlund, *Domain decomposition methods - Algorithms and theory*. Springer Series in Computational Mathematics, Springer Berlin Heidelberg, 2004.

[51] B. Smith, P. Bjorstad, W. Gropp, and W. Gropp, *Domain decomposition: Parallel multilevel methods for elliptic partial differential equations*. Cambridge University Press, 2004.

[52] J.-F. Lee and Z. Sacks, "Whitney elements time domain (WETD) methods," *IEEE Transactions on Magnetics*, vol. 31, no. 3, pp. 1325–1329, 1995.

[53] R. Suga and M. Kawahara, "Estimation of tidal current using Kalman filter finite-element method," *Computers & Mathematics with Applications*, vol. 52, pp. 1289–1298, 2006.

[54] Y. Ojima and M. Kawahara, "Estimation of river current using reduced Kalman filter finite element method," *Computer methods in applied mechanics and engineering*, vol. 198, pp. 904–911, 2009.

[55] G. Battistelli, L. Chisci, N. Forti, G. Pelosi, and S. Selleri, "Distributed finite element Kalman filter," in *2015 European Control Conference (ECC)*, pp. 3695–3700, 2015.

[56] G. Battistelli, L. Chisci, N. Forti, G. Pelosi, and S. Selleri, "Distributed finite-element Kalman filter for field estimation," *IEEE Transactions on Automatic Control*, 2017. Accepted.

[57] E. Süli and D. Mayers, *An introduction to numerical analysis*. Cambridge University Press, 2003.

[58] S. E. Cohn and D. P. Dee, "Observability of discretized partial differential equations," *SIAM Journal on Numerical Analysis*, vol. 25, no. 3, pp. 586–617, 1988.

[59] W. Kang and L. Xu, "Partial observability and its consistency for linear PDEs," *IFAC Proceedings Volumes*, vol. 46, no. 23, pp. 445–450, 2013.

[60] M. Farina and R. Carli, "Partition-based distributed Kalman filter with plug and play features," *arXiv:1507.06820*, 2015.

[61] F. Kreith, R. Manglik, and M. Bohn, *Principles of heat transfer*. Cengage Learning, 2012.

[62] T. Zhao and A. Nehorai, "Distributed sequential Bayesian estimation of a diffusive source in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 4, pp. 1511–1524, 2007.

[63] T. van Waterschoot and G. Leus, "Distributed estimation of static fields in wireless sensor networks using the finite element method," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2853–2856, 2012.

[64] J. Weimer, B. Sinopoli, and B. Krogh, "Multiple source detection and localization in advection-diffusion processes using wireless sensor networks," in *30th IEEE Real-Time Systems Symposium*, pp. 333–342, 2009.

[65] L. A. Rossi, B. Krishnamachari, and C. C. J. Kuo, "Distributed parameter estimation for monitoring diffusion phenomena using physical models," in *IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, pp. 460–469, 2004.

[66] R. Bellman and K. J. Astrom, "On structural identifiability," 1970.

[67] J. A. Jacquez and P. Greif, "Numerical parameter identifiability and estimability: Integrating identifiability, estimability, and optimal sampling design," *Mathematical Biosciences*, vol. 77, no. 1, pp. 201–227, 1985.

[68] G. Battistelli, L. Chisci, N. Forti, G. Pelosi, and S. Selleri, "Point source estimation via finite element multiple-model Kalman filtering," in *Proc. of the IEEE 54th Conference on Decision and Control (CDC)*, pp. 4984–4989, 2015.

[69] R. Becker and R. Rannacher, "An optimal control approach to a posteriori error estimation in finite element methods," *Acta Numerica*, pp. 1–101, 2001.

[70] S. Gillijns and B. D. Moor, "Unbiased minimum-variance input and state estimation for linear discrete-time systems," *Automatica*, vol. 43, no. 1, pp. 111–116, 2007.

[71] B. Li, "State estimation with partially observed inputs: A unified Kalman filtering approach," *Automatica*, vol. 49, no. 3, pp. 816–820, 2013.

[72] G. Basile and G. Marro, *Controlled and conditioned invariants in linear system theory*. Prentice Hall, 1992.

[73] P. Antsaklis and A. Michel, *A linear systems primer*. Birkhäuser Boston, 2007.

[74] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2715–2729, 2013.

[75] F. Pasqualetti, F. Dorfler, and F. Bullo, "Control-theoretic methods for cyberphysical security: Geometric principles for optimal cross-layer resilient control systems," *IEEE Control Systems*, vol. 35, no. 1, pp. 110–127, 2015.

[76] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: Theory algorithms and software*. John Wiley and Sons, 2004.

[77] P. M. Djuric, M. Vemula, and M. F. Bugallo, "Signal processing by particle filtering for binary sensor networks," in *IEEE 11th Digital Signal Processing Workshop*, pp. 263–267, 2004.

[78] L. Y. Wang, J.-F. Zhang, and G. G. Yin, "System identification using binary sensors," *IEEE Transactions on Automatic Control*, vol. 48, no. 11, pp. 1892–1907, 2003.

[79] A. Capponi, I. Fatkullin, and L. Shi, "Stochastic filtering for diffusion processes with level crossings," *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2201–2206, 2011.

[80] L. Y. Wang, G. G. Yin, and J.-F. Zhang, "Joint identification of plant rational models and noise distribution functions using binary-valued observations," *Automatica*, vol. 42, no. 4, pp. 535–547, 2006.

[81] B. Ristic, A. Gunatilaka, and R. Gailis, "Achievable accuracy in parameter estimation of a Gaussian plume dispersion model," in *2014 IEEE Workshop on Statistical Signal Processing (SSP)*, pp. 209–212, 2014.

[82] A. Ribeiro and G. B. Giannakis, "Bandwidth-constrained distributed estimation for wireless sensor networks-part II: Unknown probability density function," *IEEE Transactions on Signal Processing*, vol. 54, no. 7, pp. 2784–2796, 2006.

[83] L. Y. Wang, C. Li, G. G. Yin, L. Guo, and C. Z. Xu, "State observability and observers of linear-time-invariant systems under irregular sampling and sensor limitations," *IEEE Transactions on Automatic Control*, vol. 56, no. 11, pp. 2639–2654, 2011.

[84] E.-W. Bai, H. E. Baidoo-Williams, R. Mudumbai, and S. Dasgupta, "Robust tracking of piecewise linear trajectories with binary sensor networks," *Automatica*, vol. 61, pp. 134–145, 2015.

[85] J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus, "Tracking a moving object with a binary sensor network," in *Proc. of the 1st International Conference on Embedded Networked Sensor Systems*, pp. 150–161, 2003.

[86] G. Battistelli, L. Chisci, and S. Gherardini, "Moving horizon state estimation for discrete-time linear systems with binary sensors," in *Proc. of the 54th IEEE Conference on Decision and Control (CDC)*, pp. 2414–2419, 2015.

[87] S. Vijayakumaran, Y. Levinbook, and T. F. Wong, "Maximum likelihood localization of a diffusive point source using binary observations," *IEEE Transactions on Signal Processing*, vol. 55, no. 2, pp. 665–676, 2007.

[88] G. Battistelli, L. Chisci, N. Forti, and S. Gherardini, "Map moving horizon state estimation with binary measurements," in *Proc. of the American Control Conference (ACC)*, pp. 5413–5418, 2016.

[89] G. Ferrari-Trecate, D. Mignone, and M. Morari, "Moving horizon estimation for hybrid systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 10, pp. 1663–1676, 2002.

[90] R. A. Delgado and G. C. Goodwin, "A combined MAP and Bayesian scheme for finite data and/or moving horizon estimation," *Automatica*, vol. 50, no. 4, pp. 1116–1121, 2014.

[91] C. V. Rao, J. B. Rawlings, and D. Q. Mayne, "Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approxima-

tions," *IEEE Transactions on Automatic Control*, vol. 48, no. 2, pp. 246–258, 2003.

[92] A. Alessandri, M. Baglietto, G. Battistelli, and V. Zavala, "Advances in moving horizon estimation for nonlinear systems," in *Proc. of the 49th IEEE Conference on Decision and Control (CDC)*, pp. 5681–5688, 2010.

[93] A. Alessandri, M. Baglietto, and G. Battistelli, "Moving-horizon state estimation for nonlinear discrete-time systems: New stability results and approximation schemes," *Automatica*, vol. 44, no. 7, pp. 1753–1765, 2008.

[94] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[95] Y. Mo, S. Weerakkody, and B. Sinopoli, "Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs," *IEEE Control Systems Magazine*, vol. 35, no. 1, pp. 93–109, 2015.

[96] S. Weerakkody and B. Sinopoli, "Detecting integrity attacks on control systems using a moving target approach," in *Proc. of the 54th IEEE Conference on Decision and Control (CDC)*, pp. 5820–5826, 2015.

[97] Y. Mo and B. Sinopoli, "Secure estimation in the presence of integrity attacks," *IEEE Transactions on Automatic Control*, vol. 60, no. 4, pp. 1145–1151, 2015.

[98] Y. Nakahira and Y. Mo, "Dynamic state estimation in the presence of compromised sensory data," in *Proc. of the 54th IEEE Conference on Decision and Control (CDC)*, pp. 5808–5813, 2015.

[99] H. Fawzi, P. Tabuada, and S. Diggavi, "Secure estimation and control for cyber-physical systems under adversarial attacks," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1454–1467, 2014.

[100] M. Pajic, P. Tabuada, I. Lee, and G. Pappas, "Attack-resilient state estimation in the presence of noise," in *Proc. of the 54th IEEE Conference on Decision and Control (CDC)*, pp. 5827–5832, 2015.

[101] Y. Shoukry, A. Puggelli, P. Nuzzo, A. Sangiovanni-Vincentelli, S. Seshia, and P. Tabuada, "Sound and complete state estimation for linear dynamical systems under sensor attacks using satisfiability modulo theory solving," in *Proc. of the American Control Conference (ACC)*, pp. 3818–3823, 2015.

[102] A. Teixeira, I. Shames, H. Sandberg, and K. Johansson, "A secure control framework for resource-limited adversaries," *Automatica*, vol. 51, no. 1, pp. 135–148, 2015.

[103] S. Yong, M. Zhu, and E. Frazzoli, "Resilient state estimation against switching attacks on stochastic cyber-physical systems," in *Proc. of the 54th IEEE Conference on Decision and Control (CDC)*, pp. 5162–5169, 2015.

[104] A. Farraj, E. Hammad, A. Daoud, and D. Kundur, "A game-theoretic analysis of cyber switching attacks and mitigation in smart grid systems," *IEEE Transactions on Smart Grid*, vol. 7, no. 4, pp. 1846–1855, 2016.

[105] S. Amin, X. Litrico, S. Sastry, and A. Bayen, "Stealthy deception attacks on water scada systems," in *Proc. of the 13th ACM Int. Conference on Hybrid Systems: Computation and Control (HSCC)*, pp. 161–170, 2010.

[106] N. Forti, G. Battistelli, L. Chisci, and B. Sinopoli, "A Bayesian approach to joint attack detection and resilient state estimation," in *Proc. of the 55th IEEE Conference on Decision and Control (CDC)*, 2016.

[107] Q. Gu, P. Liu, S. Zhu, and C.-H. Chu, "Defending against packet injection attacks in unreliable ad hoc networks," in *IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 3, pp. 1837–1841, 2005.

[108] X. Zhang, H. Chan, A. Jain, and A. Perrig, "Bounding packet dropping and injection attacks in sensor networks," 2007. Tech. Rep. Available online at: `https://www.cylab.cmu.edu/files/pdfs/tech_reports/cmucylab07019.pdf`.

[109] S. Gillijns and B. D. Moor, "Unbiased minimum-variance input and state estimation for linear discrete-time systems with direct feedthrough," *Automatica*, vol. 43, no. 5, pp. 934–937, 2007.

[110] H. Fang, R. A. De Callafon, and J. Cortés, "Simultaneous input and state estimation for nonlinear systems with applications to flow field estimation," *Automatica*, vol. 49, no. 9, pp. 2805–2812, 2013.

[111] R. Mahler, *Statistical multisource multitarget information fusion.* Artech House, 2007.

[112] B.-N. Vo and W. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, 2006.

[113] B.-N. Vo and W. K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, 2006.

[114] P. Sauer and M. Pai, *Power system dynamics and stability.* Prentice Hall, 1998.

[115] F. Pasqualetti, A. Bicchi, and F. Bullo, "A graph-theoretical characterization of power network vulnerabilities," in *Proc. of the American Control Conference (ACC)*, pp. 3918–3923, 2011.

[116] S. Amini, H. Mohsenian-Rad, and F. Pasqualetti, "Dynamic load altering attacks in smart grid," in *Innovative Smart Grid Technologies Conference (ISGT)*, pp. 1–5, 2015.